Sensitivity Analysis in Translation Averaging

Lalit Manam and Venu Madhav Govindu

Abstract—Translation averaging solves for absolute translations given a set of pairwise relative translation directions. Most of the existing literature focuses on robustness to outliers and studies the uniqueness of the solution. In this paper, we deal with a distinctly different problem of sensitivity in translation averaging under uncertainty in observed directions. We first analyze sensitivity in estimating the translation of a camera and extend it to all the cameras involved in the problem. Then, we define the conditioning of the translation averaging problem, which assesses the reliability of estimated translations based solely on the input directions. We provide a sufficient criterion to ensure that the problem is well-conditioned. Based on the criterion, we present an efficient algorithm to identify and remove combinations of directions which make the problem ill-conditioned. Applying our algorithm leads to improved conditioning of translation averaging, resulting in the reduction of absolute translation errors. Project page/source code: https://ee.iisc.ac.in/cvlab/research/tasensitivity.

Index Terms—Structure-from-Motion, Global Methods, Translation Averaging, Sensitivity Analysis, Conditioning.

1 Introduction

The aim of translation averaging is to estimate absolute translations given a set of pairwise relative directions as input observations. This problem arises in global methods for solving structure-from-motion (SfM) [25], [26], [58], which, unlike incremental methods [50], [56], [64], jointly solves for absolute rotations and translations of all the cameras. Using 2D point correspondences between a pair of cameras, we can recover the relative rotation and relative translation direction between the two cameras via epipolar geometry. These relationships result in a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ is the set of nodes and edges, respectively. Each node $i \in \mathcal{V}$ represents a 3D rotation $\mathbf{R}_i \in \mathbb{SO}(3)$ and a 3D translation $\mathbf{T}_i \in \mathbb{R}^3$ with respect to a global frame of reference. Each edge $(i, j) \in \mathcal{E}$ denotes the relative rotation and translation direction $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ between the nodes i and j, where $\mathbf{R}_{ij} \in \mathbb{SO}(3)$ and $\mathbf{t}_{ij} \in \mathbb{S}^2$. These result in the following relationships:

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}, \tag{1}$$

$$\mathbf{t}_{ij} = \frac{\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)}{\|\mathbf{R}_j(\mathbf{T}_i - \mathbf{T}_j)\|}. \tag{2}$$

Absolute rotations, $\mathbf{R}_i,\ i \in \mathcal{V}$, are estimated from relative rotations, $\mathbf{R}_{ij},\ (i, j) \in \mathcal{E}$, using rotation averaging methods [12], [13], [23], [28], [53], [54]. Then, the relative directions are aligned to the global reference frame using absolute rotations as follows:

$$\mathbf{v}_{ij} = -\mathbf{R}_j^{-1}\mathbf{t}_{ij} = \frac{\mathbf{T}_j - \mathbf{T}_i}{\|\mathbf{T}_j - \mathbf{T}_i\|}, \tag{3}$$

where $\mathbf{v}_{ij} \in \mathbb{S}^2$ is the relative direction represented in a global frame of reference. Now, the problem of translation averaging can be described using the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the set of edges $\mathcal{E}$ representing the input to the problem i.e. pairwise relative directions $\mathbf{v}_{ij},\ (i, j) \in \mathcal{E}$, and the set of nodes $\mathcal{V}$ representing the output i.e. absolute translations $\mathbf{T}_i,\ i \in \mathcal{V}$. Since every edge $(i, j) \in \mathcal{E}$ represents a direction, the network $\mathcal{G}$ is called a **bearing-based network**.

The dissimilarity in the input space (directions i.e. unit norm vectors in 3D space) and output space (translations i.e. vectors in 3D space) makes translation averaging a challenging problem since it requires estimating translation scales [25]. The presence of noise and outliers in input directions makes it further difficult to obtain a reasonable solution [24], [38], [40], [47], [63], [68]. Moreover, the existence of a unique solution for a bearing-based network is related to the non-trivial issue of parallel rigidity [3], [48].

In this paper, we deal with a completely distinct aspect of translation averaging. We analyze sensitivity in translation averaging under input uncertainty. It can be seen as a perturbation analysis of the translation averaging solution with respect to small changes in input directions. This is useful for determining the configuration of input directions for which the output translations are reliable. This analysis is independent of parallel rigidity [3] since parallel rigidity deals with the uniqueness of the solution, while we deal with change in the estimate for small input perturbations. Parallel rigidity can be equivalently described with the algebraic rank of a specific matrix [3], which is similar to the analysis of the uniqueness of a solution given a matrix obtained from a linear system of equations. In contrast, our sensitivity analysis is similar in spirit to the conditioning of a matrix while solving a linear system of equations ($\mathbf{Ax} = \mathbf{b}$ problem), where the reliability of a solution is studied under uncertainty in observations. Sensitivity is also different from outlier detection in the inputs [52], [63] because it only deals with perturbations of the given input directions and not with the noise/outlier levels. The issue of sensitivity remains relevant even without outliers.

In the literature [30], [43], [55], empirical studies have been presented that show the effects of perturbing input directions in the bearing-based networks with very few nodes (3 to 20), but their procedures are computationally

- *Lalit Manam was with, and Venu Madhav Govindu is with the Department of Electrical Engineering, Indian Institute of Science, Bengaluru, Karnataka - 560012, INDIA.*
E-mail: lalitmanam@gmail.com, venug@iisc.ac.in

expensive for large-scale networks. Analyzing sensitivity requires a theoretical analysis for generalization to large-scale networks. Our previous work [39] was the first step to theoretically analyze sensitivity in the problem based solely on the input directions without employing numerical simulations with input perturbations. It takes an edge-based approach where it analyzes sensitivity in estimating edge scales using the smallest solvable network, i.e. 3 nodes with 3 edges connecting all pairs of nodes, which is also termed a *triplet of nodes*. In translation averaging, a node triplet forms a triangle whose properties are used in [39] to quantify sensitivity of the triplet. Then, it extends the principle of sensitivity to the entire bearing-based network with the restriction that the network consists of triplets of nodes. This restriction on the network topology prevents the sensitivity analysis of general networks with the formulation presented in [39].

Our Contributions: In this paper, our aim is to analyze the sensitivity in bearing-based networks without any restriction on the network topology, generalizing the idea introduced in our previous work [39]. Different from [39], where an edge-based analysis has been done, we present a completely new approach to sensitivity through a node-based analysis to achieve our goal. We first analyze the sensitivity in estimating translation at a single node with the smallest solvable problem, i.e. a node with two incident edges (directions). Then, we characterize the sensitivity in the smallest solvable problem, extend it to a node with multiple incident directions, and provide a sufficient criterion for a node to be well-conditioned. We extend this analysis over the entire network and define node-based conditioning of translation averaging. We propose an efficient algorithm to identify the configuration of directions that lead to ill-conditioned nodes and remove such directions from the network to improve the conditioning of the problem. We demonstrate the utility of this sensitivity analysis in structure-from-motion with improved accuracy of translation estimates on benchmark datasets.

2 Literature Review

In this section, we briefly review the literature on motion averaging, translation averaging and parallel rigidity.

Motion Averaging: Motion averaging [12], [13], [25], [26], [63], [68], synchronization [4], [6], [8], [29], [57], or graph optimization [18], [31], [37], [59] is a class of problems on networks where absolute values on the nodes are estimated from relative values on the edges. Many SfM [44], [58], and simultaneous localization and mapping (SLAM) [27], [45] pipelines employ these methods in its different stages. In this paper, we deal with translation averaging, a crucial component of global SfM approaches that falls into this category of problems. Each node value represents a vector in $\mathbb{R}^3$ that needs to be estimated from relative directions in $\mathbb{S}^2$ represented on edges. [67] provides a survey on several motion averaging problems.

Rotation Averaging: Rotation averaging, also belonging to the class of motion averaging problems, solves for absolute rotations (on the nodes) from relative rotations (on the edges) obtained from epipolar geometry. There are two main paradigms for solving rotation averaging. Intrisic methods [12], [13], [26], [28] utilize the Lie group structure $\mathbb{SO}(3)$ of rotations [26], [62], while extrinsic methods [14], [23], [41] solve a relaxed problem. Recent developments can be found in [19], [46], [53], [54] and their references.

Translation Averaging: In [25], translation averaging was solved by minimizing the cross-product between the observed input directions to that of the equivalent obtained from estimated translations. A linear system of cross-product constraints using epipolar geometry was formulated in [2] to solve the problem. In [43], the problem of aligning edges was converted to aligning triplets of nodes by utilizing trifocal tensors with known rotations. In a similar spirit, node triplets were used in [30] and the problem was formulated based on the constraints on a triangle formed by triplets. 1DSfM [63] compared the observed and estimated heading directions and added constraints between 3D points and cameras to stabilize the problem. A pre-processing step was incorporated in [63] to remove outliers. In [61], squared relative displacements between nodes were compared, and the problem was solved in a distributed manner based on the network structure. Least Unsquared Deviations (LUD) [47] extended [61] by using L_1 loss for robustness and posed the problem as a convex program. In [7], the squared error of the orthogonal projection of estimated relative translations onto observed directions was minimized. ShapeFit/ShapeKick [24] also minimized the orthogonal projection using ADMM but with an L_1 loss for robustness. Methods like [15], [16] used 2D point tracks to estimate the edge scales and then solved a linear system. To avoid the influence of outliers, multiple estimates of edge scales from different tracks were handled carefully. Bilinear Angle-based Translation Averaging (BATA) [68] compared the estimated heading directions from camera translations to that of the observed directions, relaxing the cost in [63]. [40] proposed to fuse the intermediate solutions obtained from the two costs: comparing displacements [47], [61] or directions [68]. Some methods removed severely corrupted directions [52] to obtain reliable translation estimates based on triplets of nodes. Other approaches which solved absolute translations include similarity averaging [17], averaging essential and fundamental matrices [33], [34], and utilization of the properties of a matrix of pairwise camera displacements [20].

Parallel Rigidity: Theoretical work on uniqueness of a solution obtained from bearing-based networks has been contributed by different communities: computer vision [3], [5], [48], robotics [36], computer-aided design [51] and decision control [21], [22], [60], [66]. There are two approaches to defining parallel rigidity. The first approach considers nodes, which deals with the configuration of absolute translations (also called a point formation) [21], [22], [51] to study parallel rigidity. The second approach deals with edges and reasons for parallel rigidity based on edge scales in terms of the cycles in a network [5], [35], [60]. Readers are referred to [3] for an excellent survey on parallel rigidity.

3 Sensitivity in Translation Estimation from Directions

In this section, we study node-based sensitivity in bearing-based networks by examining the sensitivity in estimating translation at a node given a set of directions incident from other nodes. We consider the smallest solvable case, i.e. estimating translation at a node from two directions. Then, the constraints on the current node can be written as $\mathbf{T}_i + s_i\mathbf{v}_i$, for $i \in \{1, 2\}$, where $\mathbf{v}_i \in \mathbb{S}^2$ denotes i^{th} direction and $\mathbf{T}_i \in \mathbb{R}^3$ denotes the value of i^{th} node connected to the current node via i^{th} direction. s_i denotes the free variable for i^{th} direction, which gives us a point when fixed to a value. The least squares problem to obtain the intersection point from the constraints can be written as

$$\min_{\mathbf{s}} \|\mathbf{V}\mathbf{s} - \mathbf{c}\|^2, \tag{4}$$

where $\mathbf{V} = [\ \mathbf{v}_1 \mid \text{-}\mathbf{v}_2\ ]$, $\mathbf{s} = [s_1, s_2]^T$ and $\mathbf{c} = \mathbf{T}_2 - \mathbf{T}_1$. The solution to the problem in Eqn. 4 can be obtained by solving the normal equation $\mathbf{V}^T\mathbf{V}\mathbf{s} = \mathbf{V}^T\mathbf{c}$. When $\mathbf{V}^T\mathbf{V}$ is invertible, the solution is given as

$$\mathbf{s} = \left(\mathbf{V}^T\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{c}. \tag{5}$$

For the problem in hand, $\mathbf{V}^T\mathbf{V}$ is invertible when $\mathbf{v}_1^T\mathbf{v}_2 \neq \pm 1$ i.e. the angle between the two directions is neither $0°$ nor $180°$.

The quality of the solution to the normal equation, i.e. $\mathbf{s}$, is dependent on the conditioning of the matrix $\mathbf{V}^T\mathbf{V}$ [42]. To analyze the sensitivity of the estimated translation, which is dependent on the free variables s_i, we check the conditioning of $\mathbf{V}^T\mathbf{V}$. The following theorem relates matrix-2 norm conditioning of $\mathbf{V}^T\mathbf{V}$, i.e. $\kappa(\mathbf{V}^T\mathbf{V})$, with the angle between the two directions, ϕ_{12}:

Theorem 1. *Given two directions $\mathbf{v}_1$ and $\mathbf{v}_2$, and the matrix $\mathbf{V} = [\ \mathbf{v}_1 \mid \text{-}\mathbf{v}_2\ ]$, the condition number of the matrix $\mathbf{V}^T\mathbf{V}$ with matrix-2 norm is given as*

$$\kappa(\mathbf{V}^T\mathbf{V}) = \begin{cases} \cot^2(\phi_{12}/2) & \text{if } 0° < \phi_{12} \leq 90°; \\ \tan^2(\phi_{12}/2) & \text{if } 90° < \phi_{12} < 180°; \\ \infty & \text{if } \phi_{12} \in \{0°, 180°\}, \end{cases} \tag{6}$$

where ϕ_{12} is the angle between the two directions $\mathbf{v}_1$ and $\mathbf{v}_2$.

Please refer to the supplementary material for the proof of Thm. 1. Fig. 1 provides the plot of $\kappa(\mathbf{V}^T\mathbf{V})$ with respect to ϕ_{12}. It can be seen that conditioning of $\mathbf{V}^T\mathbf{V}$ worsens when the angle between the two directions is near $0°$ or $180°$, and conditioning is best at $90°$. From Thm. 1, it can be seen that at any node in the bearing-based network $\mathcal{G}$, if the angle between all pairs of directions at that node are not close to $0°$ or $180°$, then the translation estimated at that node is reliable. Our aim is to quantify the conditioning of any node and the entire network $\mathcal{G}$ in translation averaging based solely on the input directions, which is discussed next.

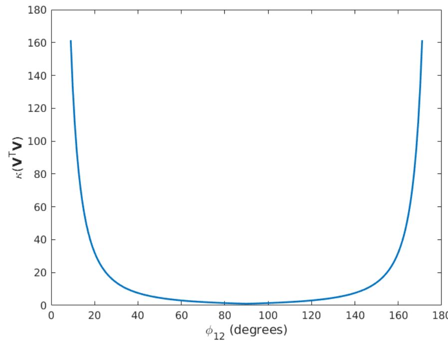
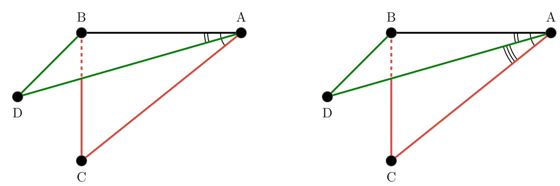


Fig. 1: Conditioning of $\mathbf{V}^T\mathbf{V}$ with angle between two directions. $\kappa(\mathbf{V}^T\mathbf{V})$: Condition number of the matrix $\mathbf{V}^T\mathbf{V}$ with matrix-2 norm. ϕ_{12}: Angle between two directions (in degrees). $\mathbf{V}^T\mathbf{V}$ is ill-conditioned near $0°$ and $180°$, and best conditioned at $90°$.



(a) Angles in [39]. (b) Angles in our method.

Fig. 2: Comparison of our proposed node-based formulation to that of the edge-based formulation [39] for sensitivity analysis. The network shown consists of two node triplets for easy comparison. For node A in the network, the edge-based formulation considers $\angle BAC$ and $\angle BAD$ (Fig. (a)), while our node-based formulation considers $\angle BAC$, $\angle BAD$ and $\angle CAD$ (Fig. (b)). This suggests that our node-based formulation uses a stronger criterion for reasoning about sensitivity in bearing-based networks compared to the edge-based formulation.

4 Node-Based Conditioning of Translation Averaging

In this section, we analyze node-based conditioning of a bearing-based network. At first, we compare our node-based formulation for quantifying conditioning of translation averaging, based on Thm. 1 to that of the edge-based formulation based on Thm. 1 presented in [39]. Our node-based formulation considers angles between all pairs of directions at a given node while the edge-based formulation [39] considers angles between the directions within the triangles formed by triplets of nodes to determine the conditioning of translation averaging.

In Fig. 2, we show an example network consisting of two triplets of nodes with a common edge between them. This allows us to make a fair comparison since the edge-based formulation is based on triplets of nodes. Consider node A in the network. The edge-based formulation [39] checks for angles only in the triplets and thus considers $\angle BAC$ and $\angle BAD$ to reason about sensitivity, as shown in Fig. 2a. Our node-based formulation checks for all possible angles between pairs of directions incident on the node, as shown in Fig. 2b, and thus considers $\angle BAC$, $\angle BAD$ and $\angle CAD$ to reason about sensitivity. Thus, our node-based formulation is built on a stronger criterion to reason about sensitivity in bearing-based networks compared to the edge-based formulation [39].

In the node-based formulation, our aim is to define the conditioning of a node in a bearing-based network $\mathcal{G}$, which captures the idea of conditioning of a translation estimate based on Thm. 1. We define a *restricted dot product matrix* given two directions $\mathbf{v}_1$ and $\mathbf{v}_2$, as

$$\mathbf{RDP}_{12} = \begin{bmatrix} |\mathbf{v}_1^T\mathbf{v}_1| & |\mathbf{v}_1^T\mathbf{v}_2| \\ |\mathbf{v}_2^T\mathbf{v}_1| & |\mathbf{v}_2^T\mathbf{v}_2| \end{bmatrix} = \begin{bmatrix} 1 & |\mathbf{v}_1^T\mathbf{v}_2| \\ |\mathbf{v}_1^T\mathbf{v}_2| & 1 \end{bmatrix}, \quad (7)$$

where $|\cdot|$ denotes the absolute value of the argument. The rows and columns of $\mathbf{RDP}_{12}$ signify the two directions $\mathbf{v}_1$ and $\mathbf{v}_2$, and the entries denote the absolute value of dot product between the directions. It can be seen that $\mathbf{RDP}_{12}$ is symmetric with diagonal entries as 1 since $\mathbf{v}_i^T\mathbf{v}_i = \|\mathbf{v}_i^2\| = 1$. The following theorem relates the matrix-2 norm condition number of $\mathbf{V}^T\mathbf{V}$ (as defined for the problem in Eqn. 4) and $\mathbf{RDP}_{12}$:

Theorem 2. *Given two directions $\mathbf{v}_1$ and $\mathbf{v}_2$, the matrix $\mathbf{V} = [\ \mathbf{v}_1 \mid \text{-}\mathbf{v}_2\ ]$, and the restricted dot product matrix for two directions $\mathbf{RDP}_{12}$, defined in Eqn. 7, the condition number of both $\mathbf{RDP}_{12}$ and $\mathbf{V}^T\mathbf{V}$ with matrix-2 norm are the same.*

Please refer to the supplementary material for the proof of Thm. 2. It can be seen that $\mathbf{RDP}_{12}$ captures the conditioning of a node with two directions in a manner similar to $\mathbf{V}^T\mathbf{V}$. Now, we extend the restricted dot product matrix for the case when a node has p edges incident on it, i.e. having p directions. It is defined as

$$rdp^{ij} = |\mathbf{v}_i^T\mathbf{v}_j|, \quad (8)$$

where rdp^{ij} denotes the entry in i^{th} row and j^{th} column of the restricted dot product matrix $\mathbf{RDP}$ for p directions, and $\mathbf{v}_k$ denotes the k^{th} direction incident on the node. It can be seen from Eqn. 8 that $\mathbf{RDP}$ is also a symmetric matrix with diagonal entries as 1, similar to $\mathbf{RDP}_{12}$. $\mathbf{RDP}$ has the desired property that when the angle between any two directions is close to $0°$ or $180°$, its condition number goes high. Let us consider any two columns of $\mathbf{RDP}$, i and j, which are given as

$$\mathbf{RDP}^{*i} = \begin{bmatrix} |\mathbf{v}_1^T\mathbf{v}_i| \\ |\mathbf{v}_2^T\mathbf{v}_i| \\ |\mathbf{v}_3^T\mathbf{v}_i| \\ \vdots \\ |\mathbf{v}_p^T\mathbf{v}_i| \end{bmatrix}, \mathbf{RDP}^{*j} = \begin{bmatrix} |\mathbf{v}_1^T\mathbf{v}_j| \\ |\mathbf{v}_2^T\mathbf{v}_j| \\ |\mathbf{v}_3^T\mathbf{v}_j| \\ \vdots \\ |\mathbf{v}_p^T\mathbf{v}_j| \end{bmatrix}. \quad (9)$$

When $\mathbf{v}_i = \pm\mathbf{v}_j$, then both $\mathbf{RDP}^{*i}$ and $\mathbf{RDP}^{*j}$ are the same, increasing the condition number of $\mathbf{RDP}$. Thus, $\mathbf{RDP}$ captures the idea of conditioning between any two directions as suggested by Thm. 1 when p directions are considered. The following theorem provides a sufficient condition for a well-conditioned restricted dot product matrix $\mathbf{RDP}$.

Theorem 3. *Consider p directions, $\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_p$, and the restricted dot product matrix $\mathbf{RDP}$ defined in Eqn. 8 using the p directions. The matrix $\mathbf{RDP}$ is well-conditioned if the angle between all pairs of directions is not close to $0°$ or $180°$.*

Please refer to the supplementary material for the proof of Thm. 3. In practice, we use a threshold to determine the closeness of a pair of directions to $0°$ or $180°$, which will be discussed in Sec. 5. The restricted dot product matrix $\mathbf{RDP}$ captures the conditioning of a single node in the bearing-based network $\mathcal{G}$. Let $\mathbf{RDP}[i]$ denote the restricted dot product matrix for node i in the network $\mathcal{G}$. We note that the size of $\mathbf{RDP}[i]$ can vary with node i since the number of edges incident at each node can be different. When the matrix $\mathbf{RDP}[i]$ is ill-conditioned for a node i in $\mathcal{G}$, we term it an **ill-conditioned node**. However, our aim is to quantify the conditioning of the entire bearing-based network. We define a block diagonal matrix, $\mathbf{RDP}_{\mathcal{G}}$, whose i^{th} diagonal block, $[\mathbf{RDP}_{\mathcal{G}}]_i$ is given as

$$[\mathbf{RDP}_{\mathcal{G}}]_i = \mathbf{RDP}[i]. \quad (10)$$

The following theorem relates the conditioning of $\mathbf{RDP}_{\mathcal{G}}$ to that of $\mathbf{RDP}[i]$ for any node i.

Theorem 4. *Consider a bearing-based network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the restricted dot product matrix $\mathbf{RDP}[i]$ for every node $i \in \mathcal{V}$ as defined in Eqn. 8, and the block diagonal matrix $\mathbf{RDP}_{\mathcal{G}}$, as defined in Eqn. 10. Then, the condition number, with matrix-2 norm, of the matrix $\mathbf{RDP}_{\mathcal{G}}$ is lower bounded by the condition number of $\mathbf{RDP}[i]$, for any $i \in \mathcal{V}$, i.e.*

$$\kappa(\mathbf{RDP}_{\mathcal{G}}) \geq \kappa(\mathbf{RDP}[i]),\ \forall i \in \mathcal{V}. \quad (11)$$

Please refer to the supplementary material for the proof of Thm. 4. Thm. 4 reveals that if any node in a bearing-based network $\mathcal{G}$ is ill-conditioned, i.e. $\kappa(\mathbf{RDP}[i])$ is high, then the condition number, $\kappa(\mathbf{RDP}_{\mathcal{G}})$, is at least as high as $\kappa(\mathbf{RDP}[i])$. Thus, $\mathbf{RDP}_{\mathcal{G}}$ captures the conditioning of all the nodes in $\mathcal{G}$. We define the **node-based conditioning of the translation averaging problem** as the condition number of the matrix $\mathbf{RDP}_{\mathcal{G}}$. It can be seen that if the sufficiency criterion in Thm. 3 is satisfied for all nodes $i \in \mathcal{V}$, then $\mathbf{RDP}_{\mathcal{G}}$ is well-conditioned, and hence, the translation averaging problem is well-conditioned.

We note that $\mathbf{RDP}_{\mathcal{G}}$ is never constructed explicitly to compute its conditioning. This is because $\lambda_{max}\left(\mathbf{RDP}_{\mathcal{G}}^T\mathbf{RDP}_{\mathcal{G}}\right) \geq \lambda_{max}\left(\mathbf{RDP}[i]^T\mathbf{RDP}[i]\right)$ and $\lambda_{min}\left(\mathbf{RDP}_{\mathcal{G}}^T\mathbf{RDP}_{\mathcal{G}}\right) \leq \lambda_{min}\left(\mathbf{RDP}[i]^T\mathbf{RDP}[i]\right)$ (please refer to the proof of Thm. 4), where $\lambda(\cdot)$ denotes an eigen value of the argument and its subscript, *max* or *min*, denotes maximum or minimum among all their eigen values, respectively. Hence, we use $\lambda_{max}\left(\mathbf{RDP}[i]^T\mathbf{RDP}[i]\right)$ and $\lambda_{min}\left(\mathbf{RDP}[i]^T\mathbf{RDP}[i]\right)$ of all nodes in $\mathcal{G}$ to compute the conditioning of $\mathbf{RDP}_{\mathcal{G}}$, which is computationally cheaper than explicitly computing conditioning of $\mathbf{RDP}_{\mathcal{G}}$. Based on Thms. 3 and 4, we develop an algorithm to remove edges from a bearing-based network to ensure all nodes are well-conditioned, which is discussed in the next section.

5 Proposed Node-Based Method

In this section, we present a method to efficiently remove edges in a bearing-based network $\mathcal{G}$ to improve the conditioning of translation averaging based on the sufficiency result of Thm. 3. We define a *restricted angle matrix*, $\mathbf{RAM}$, given p directions, $\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_p$, which can be written as

$$ram^{ij} = \cos^{-1}\left(|\mathbf{v}_i^T\mathbf{v}_j|\right) = \min(\phi_{ij}, 180^\circ - \phi_{ij}), \qquad (12)$$

where ram^{ij} denotes the entry in i^{th} row and j^{th} column of the restricted angle matrix $\mathbf{RAM}$ and ϕ_{ij} is the angle between i^{th} and j^{th} directions. We call $\min(\phi_{ij}, 180^\circ - \phi_{ij})$ as the **restricted angle** between i^{th} and j^{th} directions. From Eqn. 12, it can be seen that $\mathbf{RAM}$ is a symmetric matrix with diagonal entries as 0. We note that elements in $\mathbf{RAM}$ can be obtained from the cosine inverse of elements in $\mathbf{RDP}$. $\mathbf{RAM}$ is not similar to the distance matrix in Multi-Dimensional Scaling (MDS) [9], [10] because when $\mathbf{v}_i = -\mathbf{v}_j$, then $ram^{ij} = 0$, which violates the property of a distance function that its value is zero if and only if both of its arguments are the same. Our proposed method, which utilizes $\mathbf{RAM}$, is described in the following stages.

Stage I - Identify Edges to be Removed: Our aim is to identify a maximum set of directions at every node in $\mathcal{G}$, which do not have angles close to 0° or 180° (or the restricted angles $\min(\phi_{ij}, 180^\circ - \phi_{ij})$ not close to 0°). We want a maximum set of directions since having more observations (directions) is generally useful for any estimation problem. In other words, we want to remove as few directions as possible or identify a minimum set of directions to be removed at every node in $\mathcal{G}$. To achieve this, we construct the restricted angle matrix $\mathbf{RAM}$ at every node in $\mathcal{G}$. We note that $\mathbf{RAM}$ represents the adjacency matrix of a complete graph[1] which is undirected[2], where a node represents a direction and an edge connects the nodes i and j with a weight $\cos^{-1}\left(|\mathbf{v}_i^T\mathbf{v}_j|\right)$.

From Eqn. 12, it can be seen that the entries in $\mathbf{RAM}$ consists of restricted angles $\min(\phi_{ij}, 180^\circ - \phi_{ij})$, where ϕ_{ij} is the angle between i^{th} and j^{th} direction incident on the node in consideration. We threshold the entries of $\mathbf{RAM}$ with an angular threshold τ, which will consider both cases when pairs of directions have angles near 0° and 180°, simultaneously. We denote the thresholded matrix as $\mathbf{RAM}_{\geq\tau}$, whose entry at i^{th} row and j^{th} column is

$$ram_{\geq\tau}^{ij} = \begin{cases} 1 & \text{if } \cos^{-1}\left(|\mathbf{v}_i^T\mathbf{v}_j|\right) \geq \tau, \\ 0 & \text{otherwise.} \end{cases} \qquad (13)$$

$\mathbf{RAM}_{\geq\tau}$ also represents the adjacency matrix of a graph, in which a node represents a direction and an edge represents an **acceptable angle** between a pair of directions with respect to the threshold τ i.e the angle is not close to 0° or 180° by a margin of τ, or equivalently, the restricted angle is at least τ. Let us denote this new graph as $\mathcal{G}_{RAM_{\geq\tau}} = (\mathcal{V}_{RAM_{\geq\tau}}, \mathcal{E}_{RAM_{\geq\tau}})$. Our aim is motivated by Thm. 3, which is to find a maximum set of directions at the node in $\mathcal{G}$ that is being considered, that has acceptable angles between all pairs of directions. In effect, given a node in $\mathcal{G}$ and its corresponding $\mathcal{G}_{RAM_{\geq\tau}}$, we need to find a maximum subgraph (subgraph with maximum number of nodes) of $\mathcal{G}_{RAM_{\geq\tau}}$, where all pairs of nodes are connected to each other. Such a subgraph is called a maximum clique[3] in graph theory [1].

An equivalent problem for finding maximum cliques in a graph is to solve for maximum independent sets[4] in the complement graph[5] [32]. Instead of solving for a maximum clique in $\mathcal{G}_{RAM_{\geq\tau}}$, we aim to get a maximum independent set in the complement graph of $\mathcal{G}_{RAM_{\geq\tau}}$, denoted as $\mathcal{G}^C_{RAM_{\geq\tau}}$. An edge in $\mathcal{G}^C_{RAM_{\geq\tau}}$ denotes that restricted angle between the two directions (denoted as nodes in $\mathcal{G}^C_{RAM_{\geq\tau}}$) are close to 0°. Hence, obtaining a maximum independent set in $\mathcal{G}^C_{RAM_{\geq\tau}}$ ensures that restricted angles between pairs of directions are not close to 0°. Finding maximum independent sets is an NP-hard problem [49], and many algorithms with exponential time complexity have been proposed [11], [32], [49], [65].

In an ideal case, we would like to exactly solve for a maximum independent set in $\mathcal{G}^C_{RAM_{\geq\tau}}$. However, given our problem of translation averaging, two issues arise. First, we need to solve the maximum independent set problem at every node in the bearing-based network $\mathcal{G}$. Since the algorithms take exponential time to find independent sets, the time taken is prohibitive for large graphs. In an unordered collection of images, a node in the bearing-based network $\mathcal{G}$ can have as high as 1000 directions incident on them, which results in 1000 nodes in $\mathcal{G}^C_{RAM_{\geq\tau}}$. Repeating the process for many such nodes in $\mathcal{G}$ is expensive in terms of computation time. The second issue arises from the fact that the decision for an edge to be retained in the bearing-based network $\mathcal{G}$ is inferred from two independent decisions taken at the two incident nodes of the edge. In cases where the two decisions are conflicting, we need to either select or remove the edge that cannot favour the two decisions made by the two nodes. Thus, solving exactly for the maximum independent set problem can still lead to a compromise when conflicting decisions are handled.

Hence, we chose to solve for an independent set in $\mathcal{G}^C_{RAM_{\geq\tau}}$, which may not be maximum but is time efficient with its output reasonable for the problem in hand, i.e. translation averaging. We will use the complement of thresholded matrix $\mathbf{RAM}_{\geq\tau}$, which is denoted as $\mathbf{RAM}_{<\tau}$.

1. A *complete graph* consists of all possible edges between the nodes.
2. In an *undirected graph*, if an edge connects node i to node j, then node j is also connected to node i.
3. A *clique* is a subgraph where all possible edges exist between the nodes. A *maximal clique* is a subgraph which is not contained in any other clique. A *maximum clique* is a maximal clique with a maximum number of nodes.
4. An *independent set* in a graph is a set of nodes, none of which are connected to each other. A *maximal independent set* in a graph is a set of nodes which is independent and is not contained in any other independent set in the graph. A *maximum independent set* in a graph is a maximal independent set with a maximum number of nodes.
5. A *complement graph* of a given graph has the same nodes as the given graph in which two nodes are connected by an edge if they are not connected in the given graph, and vice versa.

Its entry for the i^{th} row and j^{th} column is given as

$$ram^{ij}_{<\tau} = \begin{cases} 1 & \text{if } \cos^{-1}\left(|\mathbf{v}_i^T \mathbf{v}_j|\right) < \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We note that $\mathbf{RAM}_{<\tau}$ is similar to the adjacency matrix of $\mathcal{G}^C_{RAM_{\geq\tau}}$ but with a difference that the diagonal elements have a value of 1. To obtain an independent set, we present a greedy procedure. We recursively remove k^{th} row and column in $\mathbf{RAM}_{<\tau}$, which has the maximum row sum until all row sums are equal to one. This is equivalent to recursively removing nodes from $\mathcal{G}^C_{RAM_{\geq\tau}}$ (and their corresponding edges), which has the highest degree until all the remaining nodes are disconnected, thus obtaining an independent set in $\mathcal{G}^C_{RAM_{\geq\tau}}$. Given the incident directions at a node in the bearing-based network $\mathcal{G}$, this process is also the same as recursively removing directions that have non-acceptable angles to other directions. When there is more than one row in $\mathbf{RAM}_{<\tau}$ with maximum row sum, we choose to remove the row and column with a lower index. This procedure ensures that nodes in $\mathcal{G}$ have directions that have acceptable angles between all their pairs. This procedure has polynomial time complexity, thus keeping compute time within reasonable limits.

In practice, we perform the above greedy procedure for all nodes independently and mark the directions which are suggested to be removed from each node. This allows us to reconsider edges once decisions about directions (edges) are obtained from all nodes in $\mathcal{G}$, which is discussed in the next stage. We note that this stage can be parallelized since analysis at each node is done independently. This reduces computation time by $> 50\%$ for $\sim 10^3$ nodes and $\sim 10^4$ edges when using 20 threads, which further reduces for larger networks. We provide a time comparison for both serial and parallel versions in Sec. 6.

Stage II - Prune Edges: In this stage, we remove edges from a bearing-based network $\mathcal{G}$ given the decisions provided by the nodes in $\mathcal{G}$. Every edge has two decisions from the two nodes it is incident on. We can prune edges **aggressively** or **non-aggressively**. In aggressive pruning, an edge is removed if one of the nodes suggests its removal. In non-aggressive pruning, we remove an edge if both edges suggest its removal. When edges are removed non-aggressively, not all directions (edges) with non-acceptable angles are removed. Thm. 3 is based on the sufficiency criterion of improving $\kappa(\mathbf{RDP}[i])$ at every node, and thus, non-aggressive pruning can still improve the conditioning of the problem. On the other hand, if we remove directions aggressively, those edges in which only one of its adjacent nodes suggests its removal are also removed. This can degrade the conditioning of such nodes that did not suggest the removal of those edges. Thus, there is no procedure that can improve the conditioning of all nodes under all possible circumstances. In practice, removing edges aggressively results in a significant fraction of edges being filtered out, severely affecting connectivity and parallel rigidity. Hence, in our experiments, we choose to remove edges non-aggressively. Algo. 1 summarizes our method.

Algorithm 1: Removal of Edges for a Well-Conditioned Bearing-Based Network

Input: Bearing-based network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is parallel rigid, and a threshold for the minimum restricted angle between two directions τ.

Output: Bearing-based network $\mathcal{G}_F = (\mathcal{V}_F, \mathcal{E}_F)$, with improved conditioning.

for *every node* $i \in \mathcal{V}$ *in* $\mathcal{G}$ **do**

- Compute the restricted angle matrix, $\mathbf{RAM}$, using Eqn. 12.
- Threshold every element of $\mathbf{RAM}$ with τ, using Eqn. 14, to obtain $\mathbf{RAM}_{<\tau}$.
- Recursively remove k^{th} row and column in $\mathbf{RAM}_{<\tau}$ with k^{th} row having the maximum sum, until all row sums are 1.
- Mark the row indexes removed from $\mathbf{RAM}_{<\tau}$ in the above step to the corresponding directions in $\mathcal{G}$.

end

Remove edges from $\mathcal{G}$ either aggressively or non-aggressively based on suggestions marked from both nodes for an edge.

Get the largest connected component of $\mathcal{G}$.

Extract the parallel rigid component of $\mathcal{G}$ to obtain $\mathcal{G}_F$, which is well-conditioned.

6 EXPERIMENTS

We consider SfM datasets provided in 1DSfM [63] for the experiments. It provides relative motions between camera pairs and a reference reconstruction using Bundler [56]. Since Bundler was published a long time ago, we use COLMAP [50] to generate pairwise relative rotations and translations, and use COLMAP's solution as the ground truth for a better reference. We take COLMAP's solution and align it to 1DSfM's solution to get absolute translations in meters. We compute absolute rotations using the rotation averaging method presented in [13]. We remove edges that are inconsistent with the estimated absolute rotations ($> 10°$), as also done in [47]. Then, we use RANSAC on the epipolar geometric relation with known absolute rotations to obtain relative translation directions. The output of this process gives us a bearing-based network. We extract the maximal parallel rigid component of the network using [36]. Since the filtering process is independent of the cost function to solve the problem, we consider two representative cost functions. Revised LUD (an improvement on [47], provided by [68]), compares relative displacements, and BATA [68], compares relative directions. Please refer to the supplementary material for their problem formulations. Our code is implemented in MATLAB. All experiments are performed on a PC with the Intel Xeon Silver 4210 processor with 128 GB RAM. Finally, in each table, **w/o** and **w/** denotes the network **before** and **after** applying Algo. 1, **RNE** denotes errors of nodes in the unfiltered network which were removed in the filtered network, $\#N_{rem}$ and $\#M_{rem}$ denote the number of nodes and edges removed only due to ill-conditioned configuration of directions and not due to connectivity and parallel rigidity, and **bold** entries denote better performance

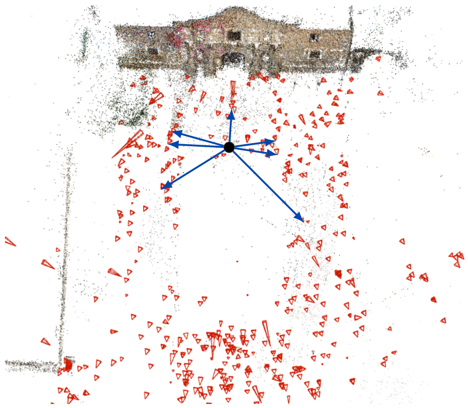


Fig. 3: Reference reconstruction of the Alamo (ALM) dataset [63] to show the spread of relative directions. Reconstruction was obtained using COLMAP [50]. Blue arrows indicate relative translation directions from the black node. Only some of the possible directions are shown for clarity.

between the solutions of the two networks.

6.1 Analysis of Real Data

In this subsection, we first provide an illustrative example and then analyze different aspects of real data.

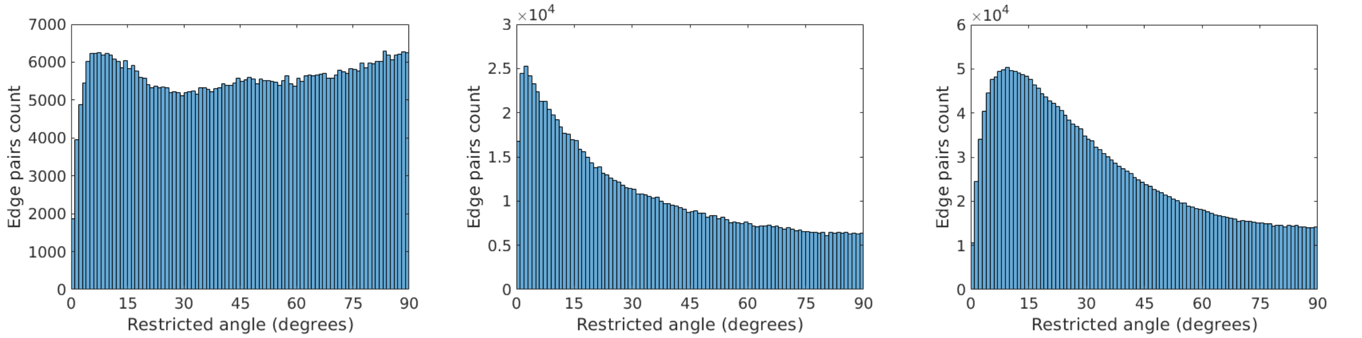
Illustration of Configurations of Directions: Fig. 3 shows an illustrative example of relative directions at a node forming different angles among them. In ALM [63], most cameras capture the front part of the museum, and thus, the cameras are densely connected in the network. Such cameras will have most of the possible connections among them. In Fig. 3, the blue arrows show some of the relative directions from the black node. It can be seen that some pairs of directions have angles close to $0°$ and $180°$ between them, while some other pairs have near $90°$. This shows that a node can be well-conditioned by removing some directions (i.e. edges in the bearing-based network $\mathcal{G}$).

Conditioning of Nodes: Here, we analyze real data to understand the conditioning of nodes in bearing-based networks. In Fig. 4, we show histograms of restricted angles ($\min(\phi_{ij}, 180° - \phi_{ij})$) between pairs of relative directions computed at all the nodes for various networks from 1DSfM datasets [63] having different edge densities. We observe that there are a considerable number of directions, pairs of which have restricted angles close to $0°$ (or equivalently, the angles are close to $0°$ or $180°$). This implies that many nodes are ill-conditioned. Moreover, it can be seen that many pairs of directions have restricted angles farther from $0°$, suggesting that the removal of directions with small restricted angles can lead to well-conditioned nodes.

Analyzing Directions: In Fig. 5, we check the minimum of restricted angles formed by a direction with other adjacent directions and plot the histograms across all directions. It can be seen that many directions have the minimum restricted angles close to $0°$, implying that removal of all such directions can lead to large parts of the bearing-based network $\mathcal{G}$ disconnected. Hence, we choose to remove edges in a non-aggressive fashion, as mentioned in Sec. 5.

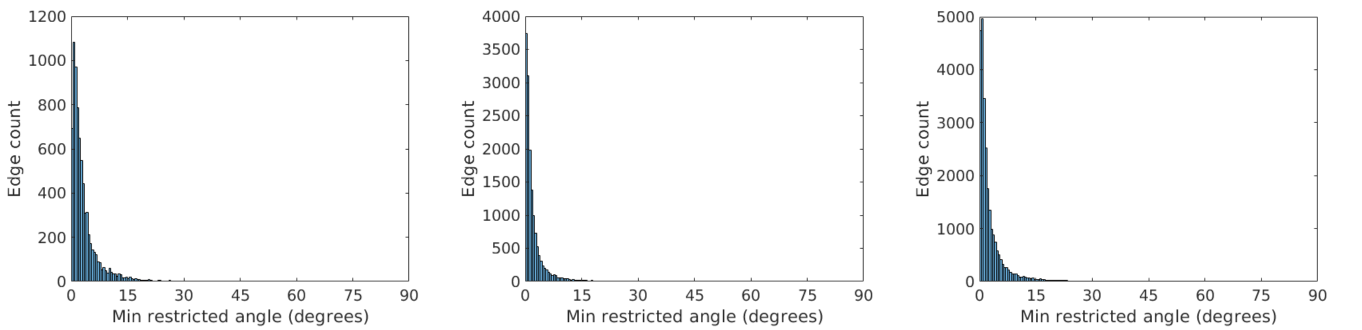
Sensitivity Independent of Outliers: In Fig. 6, we present scatter plots between errors of relative translation directions (with respect to ground truth) vs minimum of restricted angles formed by directions (edges) adjacent to other directions to study the relation between edges contributing to ill-conditioned nodes with that of the outliers. We observe that errors in directions are independent of the restricted angles formed among them, implying that the presence of outliers and ill-conditioning of nodes are not correlated.

Comparing Edge and Node-Based Formulations: We also provide a comparison of our node-based formulation with that of the edge-based formulation [39] to reason about sensitivity in bearing-based networks. For this experiment, we only consider networks consisting of triplets for a fair comparison since the edge-based formulation is applicable to such networks. We apply the method presented in [39] and Algo. 1 on the networks and check the number of nodes and edges removed in Table 1. We only report the nodes and edges removed, which were lost due to ill-conditioned configurations of directions and not because of connectivity and parallel rigidity. This enables us to empirically study the difference between two formulations. For a fair comparison, we use the same threshold ($0.5°$) for both formulations to remove ill-conditioned (or skewed) triangles using the method in [39] and edges using Algo. 1. We note that both formulations utilize angles between directions for an indication of ill-conditioned directions. Although our node-based sensitivity checks for angles between directions close to $0°$ and $180°$ while the edge-based sensitivity checks for angles in triangles close to $0°$, in effect, the edge-based sensitivity also checks for angles in triangles close to $180°$. This is because if any angle in a triangle is close to $180°$, its other two angles will be close to $0°$, and, thus, is checked by the edge-based sensitivity. Hence, using the same threshold value for both formulations in this experiment provides a similar ground for comparison. Once ill-conditioned directions are detected, we perform non-aggressive pruning of edges in both algorithms. For non-aggressive pruning in the edge-based formulation, please refer to Step 3, Sec. 5 in [39]. It can be seen that the number of nodes removed using our node-based formulation is much higher than that of the edge-based formulation. Moreover, the number of edges lost is significantly higher with our node-based formulation compared to the edge-based formulation. This is an expected behaviour since all possible angles between directions are considered at nodes in $\mathcal{G}$ for reasoning about sensitivity in our node-based formulation, while only angles in triplets are considered in the edge-based formulation (as discussed in Sec. 4). This shows that our node-based formulation uses a stronger criterion to reason about sensitivity in bearing-based networks compared to the edge-based formulation. In the following, we first provide results on outlier-free data and then on real data using our node-based formulation.



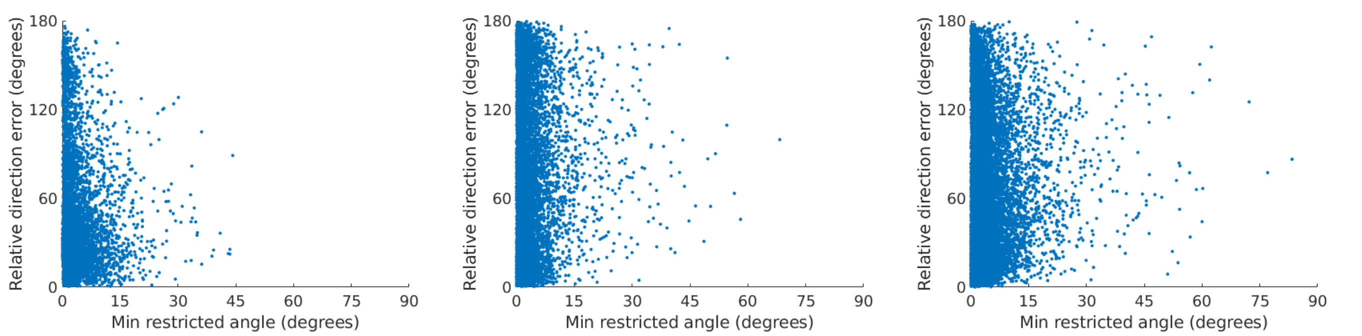
Ellis Island ($ed = 4.95\%$) Piazza del Popolo ($ed = 3.01\%$) Vienna Cathedral ($ed = 2.76\%$)

Fig. 4: Histograms of restricted angles ($\min(\phi_{ij}, 180° - \phi_{ij})$) between pairs of relative directions (pairs of edges) across all the nodes on 1DSfM datasets [63]. ed: edge density.



Ellis Island Piazza del Popolo Vienna Cathedral

Fig. 5: Histograms of minimum of restricted angles ($\min(\phi_{ij}, 180° - \phi_{ij})$) formed by directions (edges) adjacent to other directions on 1DSfM datasets [63].



Ellis Island Piazza del Popolo Vienna Cathedral

Fig. 6: Scatter plots of error of relative directions (with respect to ground truth) vs minimum of restricted angles ($\min(\phi_{ij}, 180° - \phi_{ij})$) formed by directions (edges) adjacent to other directions on 1DSfM datasets [63].

6.2 Outlier-Free Data

Here, we examine the impact of ill-conditioned nodes on the translation estimates independent of outliers. We consider 1DSfM [63] datasets with COLMAP solution as the ground truth and extract the component of the network for which ground truth is available. We use ground truth rotations (instead of estimated rotations) to align relative directions to a global coordinate system. This avoids the effects of errors in absolute rotation estimates. To make the data outlier-free, we remove any direction that differs from its ground truth by more than $10°$. Then, we extract its maximal parallel rigid component using the method in [36]. We call this network the *unfiltered network*. This gives us a network which does not have outliers but has the noise and network topology same as that of the real data. Now, we use Algo. 1 to remove edges, and refer it as the *filtered network*. To remove edges, we choose a low threshold ($\tau = 0.5°$) and employ non-aggressive pruning to avoid disconnecting $\mathcal{G}$ because the minimum restricted angles for most edges are close to $0°$ as seen in Fig. 5.

Dataset Details: In Table 2, we provide the details

TABLE 1: Comparison of nodes and edges removed using our node-based formulation to that of the edge-based formulation [39] on 1DSfM [63] datasets containing only node triplets.

Dataset	#Nodes	#Edges	Edge-based formulation [39]		Node-based formulation [Ours]	
			$\#N_{rem}$	$\#M_{rem}$	$\#N_{rem}$	$\#M_{rem}$
Alamo (ALM)	694	15619	0	4	7	9514
Ellis Island (ELS)	324	7411	0	3	0	2609
Gendarmenmarkt (GMM)	950	13913	1	28	7	8013
Madrid Metropolis (MDR)	401	4367	1	5	8	2280
Montreal Notre Dame (MND)	564	18297	0	0	17	14808
NYC Library (NYC)	450	6228	1	4	6	2870
Notre Dame (ND)	1421	70759	0	10	214	65447
Piazza del Popolo (PDP)	909	14770	2	9	36	10427
Piccadilly (PIC)	2706	45306	5	23	17	24537
Roman Forum (ROF)	1361	18855	3	42	1	6732
Tower of London (TOL)	569	8710	1	14	45	6716
Trafalgar (TFG)	6327	110876	17	131	59	66529
Union Square (USQ)	866	12638	1	5	2	5376
Vienna Cathedral (VNC)	1015	24793	1	14	31	18766
Yorkminster (YKM)	964	12033	1	6	13	5227

TABLE 2: Details of networks from 1DSfM [63] datasets without outliers. $\kappa(\mathbf{RDP}_{\mathcal{G}})$: Condition number of the block diagonal matrix $\mathbf{RDP}_{\mathcal{G}}$ with matrix-2 norm. t_{filter}, t^{par}_{filter}: Time taken without and with Steps 1-6 in Algo. 1 parallelized.

Dataset	#Nodes		#Edges		$\#N_{rem}$	$\#M_{rem}$	$\kappa(\mathbf{RDP}_{\mathcal{G}}) \downarrow$		t_{filter} (sec)	t^{par}_{filter}
	w/o filter	w/ filter	w/o filter	w/ filter			w/o filter	w/ filter	(sec)	(sec)
ALM	598	582	7671	3803	4	3856	2.4e+10	1.2e+10	0.25	0.14
ELS	133	133	721	714	0	7	4.5e+09	4.5e+09	0.02	0.05
GMM	712	685	8212	3980	5	4195	1.4e+10	9.2e+09	0.17	0.09
MDR	283	265	2663	1290	6	1352	6.9e+10	4.0e+10	0.05	0.06
MND	467	413	10417	2112	18	8256	3.0e+10	6.8e+10	0.21	0.11
NYC	346	341	3291	2055	1	1230	6.3e+09	4.4e+09	0.06	0.07
ND	1257	967	26973	4382	121	22356	4.4e+10	5.5e+10	0.89	0.25
PDP	722	485	8770	2183	34	5925	4.4e+10	1.0e+10	0.21	0.10
PIC	2051	1973	22257	11606	17	10559	6.7e+10	2.6e+10	0.86	0.23
ROF	1088	1084	9329	6892	0	2433	1.1e+11	7.4e+10	0.25	0.10
TOL	501	312	5260	1291	26	3512	6.2e+10	2.6e+10	0.10	0.07
TFG	4586	4401	55477	25879	36	29310	2.4e+11	2.6e+11	3.61	0.87
USQ	446	436	4189	2543	0	1624	1.7e+10	1.0e+10	0.07	0.06
VNC	708	554	11957	3016	10	8242	4.1e+10	1.5e+10	0.25	0.10
YKM	442	415	4334	2583	3	1701	6.0e+10	3.4e+10	0.07	0.06

of 1DSfM [63] datasets without outliers. It can be seen that only a small number of nodes but many edges are removed for most datasets by Algo. 1. This is because more the number of edges (directions), more likely it is to have restricted angles between directions close to $0°$, thus making the nodes ill-conditioned. Moreover, the conditioning improves for most datasets except MND, ND, and TFG. This is because we prune edges non-aggressively to retain connectivity in $\mathcal{G}$, due to which restricted angles between some directions can still be close to $0°$. We also compare the time taken by Algo. 1 in serial and parallel versions. It can be seen that, for large datasets ($>$ 500 cameras), the parallelized version is faster than the serial version by 50%, while for small datasets, the time taken is similar. This suggests that the parallelized version of Algo. 1 is more suitable for large datasets, while the serial version should be preferred for small datasets.

TABLE 3: Absolute translation errors (in m) on 1DSfM [63] datasets without outliers using BATA [68]. '-': No nodes were removed by Algo. 1. t_{BATA}: Time taken by BATA.

Dataset	Mean		RMS		RNE		t_{BATA}
	w/o	w/	w/o	w/	Mean	RMS	(sec)
ALM	2.8	**2.7**	5.4	**5.3**	5.6	7.6	7
ELS	0.9	**0.8**	1.2	**1.1**	-	-	2
GMM	8.4	**8.0**	13.3	**13.2**	8.7	9.8	5
MDR	6.8	**7.9**	**13.0**	15.2	13.7	17.0	3
MND	2.5	**2.2**	4.2	**3.3**	2.9	4.9	3
NYC	2.3	**2.2**	**4.4**	**4.4**	1.3	1.3	4
ND	**2.6**	2.8	**5.1**	5.5	2.7	4.7	8
PDP	5.0	**4.5**	6.4	**5.9**	5.5	7.4	3
PIC	2.5	**2.4**	5.2	**5.0**	4.0	6.1	34
ROF	7.2	**5.3**	14.9	**11.0**	-	-	12
TOL	9.2	**7.4**	18.5	**14.2**	5.4	7.2	2
TFG	8.1	**7.2**	15.6	**12.8**	13.6	19.2	45
USQ	5.1	**5.0**	7.6	**6.7**	-	-	4
VNC	5.9	**5.2**	8.0	**7.9**	8.9	9.7	5
YKM	5.8	**3.9**	12.6	**7.7**	24.1	34.4	4

Camera Translation Estimates: In Table 3, we report absolute translation errors obtained using BATA [68]. It can be seen that the removal of edges (as seen in Table 2) improves the conditioning of nodes, leading to improved translation accuracy. We also check the errors of the removed nodes (RNE) due to them being ill-conditioned in Table 3. It can be seen that such nodes have higher errors compared to errors in the unfiltered network, suggesting that the translation estimates of the removed nodes are not reliable. Absolute translation errors obtained using RLUD are provided in the supplementary material, which also shows that the removal of edges using Algo. 1 leads to improvement in translation estimates. We also observe that the time taken to remove edges is $< 5\%$ of the time taken for translation averaging using BATA and RLUD for most datasets, which further

TABLE 4: Details of networks from 1DSfM [63] datasets. $\kappa(\mathbf{RDP}_{\mathcal{G}})$: Condition number of the block diagonal matrix $\mathbf{RDP}_{\mathcal{G}}$ with matrix-2 norm. t_{filter}, t^{par}_{filter}: Time taken without and with Steps 1-6 in Algo. 1 parallelized.

Dataset	#Nodes		#Edges		$\#N_{rem}$	$\#M_{rem}$	$\kappa(\mathbf{RDP}_{\mathcal{G}}) \downarrow$		t_{filter}	t^{par}_{filter}
	w/o filter	w/ filter	w/o filter	w/ filter			w/o filter	w/ filter	(sec)	(sec)
ALM	822	801	16192	6602	5	9571	9.5e+10	3.9e+10	0.42	0.19
ELS	325	325	7430	4819	0	2611	2.0e+10	1.4e+10	0.12	0.07
GMM	962	928	13930	5942	4	7937	5.2e+10	2.2e+10	0.34	0.13
MDR	435	381	4521	2131	8	2288	6.0e+10	3.4e+10	0.09	0.08
MND	581	537	18356	3513	16	14809	3.5e+10	6.6e+10	0.37	0.12
NYC	704	664	7782	4493	8	3183	1.2e+10	9.9e+09	0.16	0.09
ND	1421	1003	70771	5091	213	65449	5.4e+10	2.6e+10	2.42	0.65
PDP	980	850	15007	4350	34	10469	6.8e+10	1.7e+10	0.36	0.11
PIC	2826	2769	45944	21273	15	24603	1.7e+11	1.2e+11	2.13	0.59
ROF	1404	1391	19049	12306	1	6724	6.2e+10	4.5e+10	0.53	0.15
TOL	633	451	8960	1993	40	6715	2.1e+10	1.1e+10	0.22	0.10
TFG	6957	6559	113832	46142	42	66895	4.7e+11	1.4e+11	10.47	2.49
USQ	1002	987	13133	7685	2	5396	3.5e+10	2.2e+10	0.30	0.11
VNC	1284	1084	26023	6566	24	18776	2.2e+11	6.3e+10	0.75	0.19
YKM	990	937	12115	6824	10	5221	5.7e+10	2.3e+10	0.34	0.14

decreases for large datasets. This shows the efficiency of our method, making it practical for use.

6.3 Real Data

In this subsection, we present results on real data.

Dataset Details: In Table. 4, we provide the details of 1DSfM [63] datasets before and after edge removal using Algo. 1. We use the same threshold for experiments ($\tau = 0.5°$) and prune edges non-aggressively, similar to the outlier-free data. We observe that a small number of nodes and many edges are removed after applying our filter, as seen for outlier-free data. This is because many directions have restricted angles close to $0°$, which is evident from Figs. 4 and 5. The conditioning also improves for most datasets after removing edges non-aggressively in our method, as seen in Table 4. Additionally, the parallel version of Algo. 1 is much faster for almost all datasets, with compute time similar to the serial version for smaller datasets. A comparison of aggressive and non-aggressive edge pruning, and the impact of varying τ in our method is provided in Sec. C of the supplementary material.

Camera Translation Estimates: In Table 5, we compare absolute translations obtained using BATA [68]. It can be seen that mean and RMS errors decrease for most datasets. Moreover, nodes that were removed due to ill-conditioned directions have higher errors than the overall errors in the unfiltered networks, suggesting that the removed nodes are not reliable for translation estimation. In Table 6, we also find similar observations when using RLUD [47], [68]. This shows that sensitivity is an independent aspect and is not related to the cost function in translation averaging. Similar to outlier-free data, the time taken to remove edges is $< 3\%$ of the time required for translation averaging.

TABLE 5: Absolute translations errors (in m) on 1DSfM [63] datasets using BATA [68]. t_{BATA}: Time taken by BATA. '-': No nodes were removed by Algo. 1.

Dataset	Mean		RMS		RNE		t_{BATA}
	w/o	w/	w/o	w/	Mean	RMS	(sec)
ALM	**5.2**	5.3	**9.6**	9.8	8.4	11.2	10
ELS	22.8	**22.4**	51.9	**48.5**	-	-	6
GMM	40.7	**40.6**	61.0	**60.4**	87.9	104.3	7
MDR	13.7	**12.5**	30.4	**27.5**	6.7	11.5	3
MND	4.4	**3.4**	10.2	**6.2**	14.8	34.5	4
NYC	6.5	**4.7**	13.6	**8.5**	73.7	73.7	6
ND	**3.3**	**3.3**	6.4	**5.8**	4.0	7.3	7
PDP	7.7	**7.1**	12.9	**10.6**	7.1	11.3	5
PIC	5.5	**5.2**	11.9	**10.8**	12.8	20.5	45
ROF	12.5	**11.2**	25.3	**23.7**	36.0	36.0	20
TOL	15.7	**17.3**	**29.0**	31.4	14.0	28.8	3
TFG	19.7	**14.4**	69.5	**27.7**	60.7	96.3	124
USQ	15.9	**13.9**	25.1	**20.7**	43.2	43.3	4
VNC	13.9	**8.9**	27.7	**13.2**	32.7	55.0	9
YKM	27.9	**14.2**	39.6	**22.5**	45.1	54.2	9

TABLE 6: Absolute translations errors (in m) on 1DSfM [63] datasets using RLUD [47], [68]. t_{RLUD}: Time taken by RLUD. '-': No nodes were removed by Algo. 1.

Dataset	Mean Errors		RMS Errors		RNE		t_{RLUD}
	w/o	w/	w/o	w/	Mean	RMS	(sec)
ALM	**5.7**	5.9	**10.1**	10.3	6.9	8.1	6
ELS	19.6	**19.5**	35.7	**34.7**	-	-	4
GMM	41.6	**40.9**	60.3	**59.1**	60.8	74.9	6
MDR	15.3	**14.2**	29.4	**27.4**	9.0	12.9	2
MND	3.8	**3.4**	5.7	**5.2**	5.6	8.9	3
NYC	5.3	**4.7**	8.8	**8.1**	11.0	11.0	4
ND	**3.9**	4.4	6.6	**6.7**	4.5	7.3	6
PDP	7.6	**7.5**	11.2	**10.0**	7.3	11.1	5
PIC	5.1	**4.9**	9.6	**9.0**	6.5	9.8	40
ROF	17.5	**16.8**	33.2	**32.8**	23.6	23.6	15
TOL	**19.0**	23.8	**33.4**	39.2	14.4	24.8	2
TFG	17.7	**13.3**	57.7	**23.4**	38.6	54.7	130
USQ	**13.6**	13.7	**20.2**	**20.2**	33.7	34.5	7
VNC	13.1	**10.0**	24.1	**13.2**	19.2	27.1	7
YKM	19.9	**17.0**	29.4	**23.8**	41.8	51.3	6

Impact on 3D Reconstruction: We obtain 3D reconstructions using Theia [58] with the absolute translations from Table 5. In Table 7, we report the number of 3D points triangulated for unfiltered and filtered networks, where we observe a reduction after applying our filter. This is because our method removes many edges, leading to the loss of many 2D point matches that belong to these edges. Although there is a reduction in 3D points, the translation estimates are more reliable after improving the conditioning of translation averaging. This can be useful in SfM pipelines in the following ways. First, we can add back the filtered edges that are consistent with the absolute translation solution ob-

TABLE 7: Number of 3D points reconstructed with BATA [68] solutions on 1DSfM [63] datasets.

Dataset	# 3D Points (in 10^3)		Dataset	# 3D Points (in 10^3)	
	w/o	w/		w/o	w/
ALM	148	141	PDP	92	74
ELS	57	51	PIC	228	216
GMM	103	90	ROF	194	184
MDR	50	35	TOL	109	45
MND	117	90	USQ	57	48
NYC	96	92	VNC	247	170
ND	375	196	YKM	177	159

tained from a well-conditioned bearing-based network. This can reduce the loss of 3D points as more 2D point matches are available from the edges that are added back. Second, we can fix the cameras whose translation estimates are available and add the lost cameras based on the constraints from fixed cameras. This will not only avoid affecting the currently estimated cameras due to ill-conditioned cameras that were lost, but also potentially allow reliable recovery of the lost cameras during bundle adjustment. Our focus in this paper is on sensitivity analysis, and we defer the suggested pipeline improvements to future work.

7 Conclusion

In this paper, we introduce the idea of sensitivity in translation averaging under input uncertainty. We study the sensitivity in estimating the translation of a single camera, which suggests that the angle between two directions, when being close to 0° or 180°, leads to an unstable solution. We extend this principle to multiple directions for a single camera and then to the entire bearing-based network. We define node-based sensitivity in translation averaging and provide a sufficient criterion to ensure that the problem is well-conditioned. Then, we propose an efficient algorithm to identify and remove those edges which made the problem ill-conditioned. We demonstrate the utility of sensitivity analysis with improved translation estimates on benchmark datasets. Moreover, we also compare our node-based formulation to the edge-based formulation [39], which reveals that our node-based formulation uses a stronger criterion of sensitivity than the edge-based formulation.

Acknowledgments

Lalit Manam was supported by a Prime Minister's Research Fellowship, Government of India. This research was supported in part by a Core Research Grant from the Science and Engineering Research Board, Department of Science and Technology, Government of India.

References

[1] James Abello, Panos M Pardalos, Mauricio G C Resende, and Others. On maximum clique problems in very large graphs. *External Memory Algorithms*, 50:119–130, 1998.
[2] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 81–88. IEEE, 2012.
[3] Federica Arrigoni and Andrea Fusiello. Bearing-based network localizability: A unifying view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2049–2069, 2018.
[4] Federica Arrigoni and Andrea Fusiello. Synchronization problems in computer vision with closed-form solutions. *International Journal of Computer Vision*, 128(1):26–52, 2020.
[5] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. On computing the translations norm in the epipolar graph. In *International Conference on 3D Vision*, pages 300–308. IEEE, 2015.
[6] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. In *International Conference on 3D Vision*, pages 546–555. IEEE, 2016.
[7] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Robust and efficient camera motion synchronization via matrix decomposition. In *International Conference on Image Analysis and Processing*, pages 444–455. Springer, 2015.
[8] 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.
[9] Richard Beals, David H Krantz, and Amos Tversky. Foundations of multidimensional scaling. *Psychological Review*, 75(2):127, 1968.
[10] Ingwer Borg and Patrick J F Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005.
[11] Nicolas Bourgeois, Bruno Escoffier, Vangelis T Paschos, and Johan M M van Rooij. Fast algorithms for max independent set. *Algorithmica*, 62:382–415, 2012.
[12] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *International Conference on Computer Vision*, pages 521–528. IEEE, 2013.
[13] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):958–972, 2017.
[14] David Crandall, Andrew Owens, Noah Snavely, and Daniel P Huttenlocher. Discrete-continuous optimization for large-scale structure-from-motion. In *Conference on Computer Vision and Pattern Recognition*, pages 3001–3008. IEEE, 2011.
[15] Hainan Cui, Shuhan Shen, and Zhanyi Hu. Robust global translation averaging with feature tracks. In *International Conference on Pattern Recognition*, pages 3727–3732. IEEE, 2016.
[16] Zhaopeng Cui, Nianjuan Jiang, Chengzhou Tang, and Ping Tan. Linear global translation estimation with feature tracks. In *European Conference on Computer Vision*, volume 3, pages 61–75, 2014.
[17] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *International Conference on Computer Vision*, pages 864–872. IEEE, 2015.
[18] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, 2012.
[19] Frank Dellaert, David M Rosen, Jing Wu, Robert E Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing $\mathrm{SO}(p)^n$. In *European Conference on Computer Vision*, volume 12351, pages 292–308. Springer, 2020.
[20] Qiulei Dong, Xiang Gao, Hainan Cui, and Zhanyi Hu. Robust camera translation estimation via rank enforcement. *IEEE Transactions on Cybernetics*, 2020.
[21] Tolga Eren, Walter Whiteley, and Peter N Belhumeur. Using angle of arrival (bearing) information in network localization. In *IEEE Conference on Decision and Control*, pages 4676–4681. IEEE, 2006.
[22] Tolga Eren, Walter Whiteley, A Stephen Morse, Peter N Belhumeur, and Brian D O Anderson. Sensor and network topologies of formations with direction, bearing, and angle information between agents. In *IEEE International Conference on Decision and Control*, volume 3, pages 3064–3069. IEEE, 2003.
[23] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *Conference on Computer Vision and Pattern Recognition*, pages 127–135. IEEE, 2018.
[24] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure-from-motion. In *European Conference on Computer Vision*, pages 289–304. Springer, 2016.
[25] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 218–225. IEEE, 2001.
[26] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Conference on Computer Vision and Pattern Recognition*, volume 1. IEEE, 2004.
[27] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2011.
[28] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L-1 rotation averaging using the weiszfeld algorithm. In *Conference on Computer Vision and Pattern Recognition*, pages 3041–3048. IEEE, 2011.

[29] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. *Advances in neural information processing systems*, 30:1459–1468, 2017.

[30] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *International Conference on Computer Vision*, pages 481–488. IEEE, 2013.

[31] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[32] Richard M Karp. *Reducibility Among Combinatorial Problems*. Springer, 2010.

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

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

[35] Bastian Katz, Marco Gaertler, and Dorothea Wagner. Maximum rigid components as means for direction-based localization in sensor networks. *Lecture Notes in Computer Science*, 4362:330–341, 2007.

[36] Ryan Kennedy, Kostas Daniilidis, Oleg Naroditsky, and Camillo J Taylor. Identifying maximal rigid components in bearing-based localization. In *International Conference on Intelligent Robots and Systems*, pages 194–201. IEEE, 2012.

[37] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g^2o: A general framework for graph optimization. In *International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[38] Lalit Manam and Venu Madhav Govindu. Correspondence reweighted translation averaging. In *European Conference on Computer Vision*, pages 56–72. Springer, 2022.

[39] Lalit Manam and Venu Madhav Govindu. Sensitivity in translation averaging. In *Advances in Neural Information Processing Systems*, pages 62740–62763, 2023.

[40] Lalit Manam and Venu Madhav Govindu. Fusing directions and displacements in translation averaging. In *International Conference on 3D Vision*, pages 75–84. IEEE, 2024.

[41] Daniel Martinec and Tomáš Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[42] Carl D Meyer. *Matrix Analysis and Applied Linear Algebra*, volume 71. SIAM, 2000.

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

[44] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[45] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[46] Carl Olsson, Yaroslava Lochman, Johan Malmport, and Christopher Zach. Certifiably optimal anisotropic rotation averaging. In *International Conference on Computer Vision*, pages 14856–14865. IEEE, 2025.

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

[48] Onur Ozyesil, Amit Singer, and Ronen Basri. Stable camera motion estimation using convex programming. *SIAM Journal on Imaging Sciences*, 8(2):1220–1262, 2015.

[49] John Michael Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.

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

[51] Brigitte Servatius and Walter Whiteley. Constraining plane configurations in computer-aided design: Combinatorics of directions and lengths. *SIAM Journal on Discrete Mathematics*, 12(1):136–153, 1999.

[52] Yunpeng Shi and Gilad Lerman. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. In *Conference on Computer Vision and Pattern Recognition*, pages 2868–2876. IEEE, 2018.

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

[54] Chitturi Sidhartha and Venu Madhav Govindu. It is all in the weights: Robust rotation averaging revisited. In *International Conference on 3D Vision*, pages 1134–1143. IEEE, 2021.

[55] Kristy Sim and Richard Hartley. Recovering camera motion using L-∞ minimization. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1230–1237. IEEE, 2006.

[56] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH 2006 Papers*, pages 835–846. ACM, 2006.

[57] Yifan Sun, Jiacheng Zhuo, Arnav Mohan, and Qixing Huang. K-best transformation synchronization. In *International Conference on Computer Vision*, pages 10252–10261. IEEE, 2019.

[58] Christopher Sweeney, Tobias Hollerer, and Matthew Turk. Theia: A fast and scalable structure-from-motion library. In *International Conference on Multimedia*, pages 693–696. ACM, 2015.

[59] Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.

[60] Roberto Tron, Luca Carlone, Frank Dellaert, and Kostas Daniilidis. Rigid components identification and rigidity control in bearing-only localization using the graph cycle basis. In *American Control Conference*, pages 3911–3918. IEEE, 2015.

[61] Roberto Tron and René Vidal. Distributed image-based 3D localization of camera sensor networks. In *International Conference on Decision and Control*, pages 901–908. IEEE, 2009.

[62] Kyle Wilson and David Bindel. On the distribution of minima in intrinsic-metric rotation averaging. In *Conference on Computer Vision and Pattern Recognition*, pages 6031–6039. IEEE, 2020.

[63] Kyle Wilson and Noah Snavely. Robust global translations with 1DSfM. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014.

[64] Changchang Wu. VisualSfM: A visual structure-from-motion system. http://www.cs.washington.edu/homes/ccwu/vsfm, 2011.

[65] Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255:126–146, 2017.

[66] Shiyu Zhao and Daniel Zelazo. Localizability and distributed protocols for bearing-based network localization in arbitrary dimensions. *Automatica*, 69:334–341, 2016.

[67] Zhenjun Zhao, Heng Yang, Bangyan Liao, Yingping Zeng, Shaocheng Yan, Yingdong Gu, Peidong Liu, Yi Zhou, Haoang Li, and Javier Civera. Advances in global solvers for 3D vision. *arXiv preprint arXiv:2602.14662*, 2026.

[68] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *Conference on Computer Vision and Pattern Recognition*, pages 4539–4547. IEEE, 2018.

Lalit Manam received his PhD in Electrical Engineering from the Indian Institute of Science, Bengaluru, INDIA. His research interests broadly concern 3D reconstruction problems in computer vision. Prior to his PhD, he worked on impulse noise removal from colour images.

Venu Madhav Govindu is Professor, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, INDIA. His research interests are in geometric problems in 3D computer vision.