Learning to Filter Outlier Edges in Global SfM

Nicole Damblon[1] Marc Pollefeys[1,2] Dániel Baráth[1,3]
[1]ETH Zurich [2]Microsoft [3]HUN-REN SZTAKI

Abstract

We present a novel approach to enhance camera pose estimation in global Structure-from-Motion (SfM) frameworks by filtering inaccurate pose graph edges – representing relative translation estimates – before applying translation averaging. In SfM, pose graph vertices represent images, and edges represent relative poses (rotations and translations) between cameras. We reformulate the edge filtering problem as a vertex filtering in the dual graph, specifically, a line graph where vertices correspond to edges in the original graph and edges correspond to cameras. Utilizing this representation, we frame the problem as a binary classification over nodes in the dual graph. To identify outlier edges, we employ a Transformer-based architecture. To overcome the challenge of memory overflow caused by converting to a line graph, we introduce a clustering-based graph processing approach, enabling our method to be applied to arbitrarily large pose graphs. Our method outperforms existing relative translation filtering techniques in terms of camera position accuracy and can be seamlessly integrated with other filters. The code is available at https://github.com/DmblnNicole/LFOE-GlobalSfM.

1. Introduction

Generating 3D models from large sets of unordered images is a significant challenge in computer vision and robotics, with applications ranging from crowd-sourced mapping to autonomous navigation. The leading approach for 3D reconstruction is *Structure-from-Motion* (SfM), which simultaneously estimates camera parameters and reconstructs a 3D point cloud [62]. SfM methods can be broadly categorized into two types: *incremental* [29, 51, 56, 57, 66] and *global* methods [15, 49, 58, 70].

Incremental methods sequentially integrate images into the 3D reconstruction, achieving high accuracy through repeated numerical optimizations. Despite their precision, these approaches are computationally intensive due to the necessity of multiple bundle adjustment runs [60], making them less practical for large-scale datasets. In contrast, global methods optimize the pose graph in a single run, providing a faster and more scalable solution, although the pose graph may be contaminated by noise. While global methods are generally considered less accurate than incremental techniques [14], they offer promising ways for rapid 3D model reconstruction. Recent advancements, such as GLOMAP [49], have demonstrated the scalability and efficiency of such approaches, achieving state-of-the-art runtimes and accuracies comparable to those of COLMAP [51]. In this paper, we focus on global strategies and present an algorithm designed to enhance camera position estimation by using a Transformer-based technique to filter out outlier edges within the pose graph.

Global methods begin by identifying image pairs with overlapping fields of view, using image retrieval techniques like visual bag-of-words [21] or NetVLAD [1]. Subsequently, relative pose estimation is run using robust methods like RANSAC [22] or its state-of-the-art variants [6, 8]. Once camera rotations are established, translations are estimated while keeping rotations fixed, leading to the generation of a 3D point cloud. Finally, bundle adjustment is applied to refine the 3D model and camera parameters.

Translation averaging, also known as synchronization, is a well-studied problem [4, 25, 30, 44, 48, 61, 64, 72] and remains one of the most challenging aspects of global SfM pipelines. It is typically formulated as an optimization that seeks consistency of the estimated relative translations and the unknown global camera positions. A core difficulty arises from the absence of known translation scales, making it difficult to distinguish between short and long translations based solely on angular measurements. Noise significantly impacts the process, especially for short translation edges, which are more likely to become outliers. Moreover, estimating robust relative translations from image pairs is generally less stable than estimating rotations [8]. These challenges highlight the importance of filtering outlier translation edges to simplify the translation averaging problem.

In contrast, translation filtering has not been explored as extensively as averaging. 1DSfM [64] proposed a theoretical method for discarding outlier edges, but despite its solid theoretical foundation, it has been found to offer only marginal benefits in practice [41]. A recent effort [41] shifts focus from outlier filtering to eliminating camera configu-

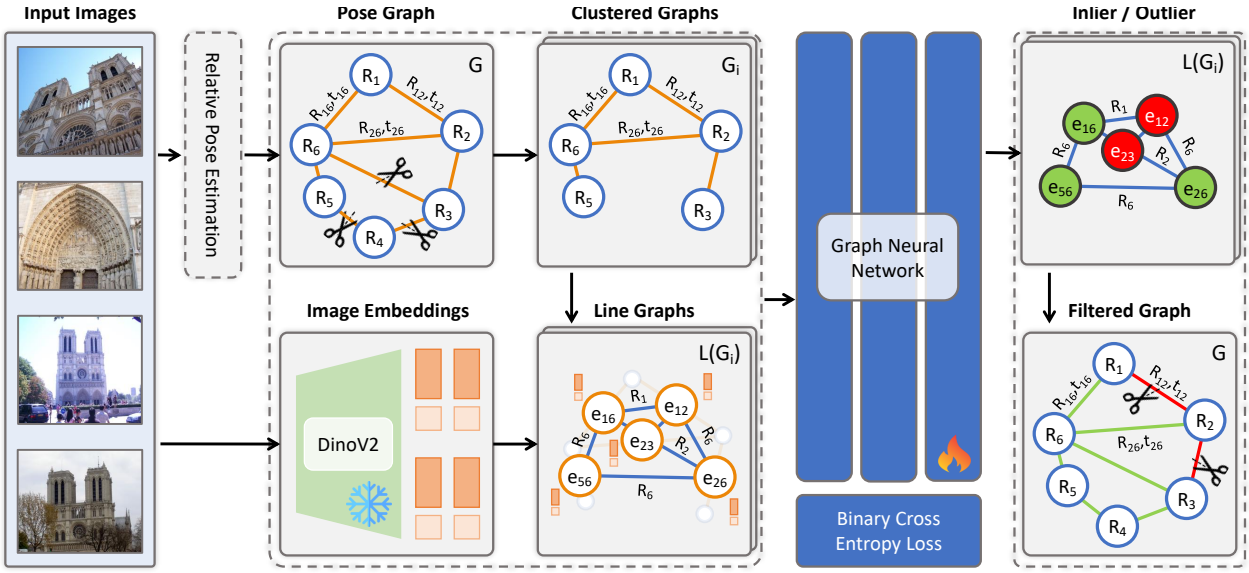

Figure 1. **Learned outlier filtering** takes pose graph G, where nodes represent images I_j with known orientations R_j and edges relative poses (R_{jk}, t_{jk}). To avoid memory issues, G is clustered into subgraphs G_i. Each G_i is converted into its line graph $L(G_i)$, where images become edges and edges images. Each edge in $L(G_i)$ is equipped with the camera orientation and image embedding [47], and each vertex with the relative pose. Finally, a Graph Neural Network predicts inlier/outlier labels for vertices in $L(G_i)$ that are used to filter edges in G.

rations that are detrimental to translation averaging, specifically those involving camera triplets in quasi-linear motion.

This paper presents a learning-based pipeline designed to identify outlier edges within a pose graph of relative translations. Our approach, illustrated in Fig. 1, transforms the pose graph into a line graph, where cameras become edges and relative translations become nodes. This setup frames outlier detection as a binary classification task, with each node classified as an inlier or outlier using a Transformer-based architecture. To address the increased memory requirements of transforming the pose graph into a line graph, we introduce a graph clustering approach. This enables our method to scale to graphs of any size. We evaluate the method on several large-scale real-world datasets, demonstrating substantial improvements in camera position estimation compared to the baseline outlier filter, 1DSfM [64]. Furthermore, when combined with the recent degeneracy filter [41], our method achieves the best results across all tested accuracy metrics and datasets.

2. Related Work

Global Structure-from-Motion reconstructs the scene in 3D and obtains the global camera poses by integrating relative pose estimates between image pairs. Predominantly, this integration is achieved through subsequent rotation [44, 46] and translation averaging [72], though some studies have opted to perform averaging within SE(3) [15]. Following the determination of rotations, the method proceeds to obtain translations and structural details that are, finally, jointly optimized by bundle adjustment. Several open-source frameworks support global Structure-from-Motion, including Theia [58], GLOMAP [49] and OpenMVG [45]. We have incorporated our relative translation filtering method into GLOMAP and Theia, keeping the rest of the pipelines unchanged. Nonetheless, our advancements are versatile and not restricted to a particular pipeline.

Rotation averaging has a long history. Govindu [25] linearizes the problem using a quaternion representation, while Martinec and Pajdla [43] simplify it by omitting certain non-linear constraints. Wilson *et al*. [65] examine the conditions that make the problem tractable. Eriksson *et al*. [20] solve it via strong duality. Semidefinite programming-based relaxation approaches [2, 23] provide optimality guarantees by minimizing the chordal distance [28]. Dellaert *et al*. [18] sequentially lift the problem into higher-dimensional rotations $\mathrm{SO}(n)$ to avoid local minima where standard numerical optimization techniques may fail [38, 42]. To handle outliers in relative rotations, various robust loss functions have been investigated [10, 11, 27, 55, 69].

The objective of **translation averaging** is to determine the absolute camera positions by leveraging estimated pairwise unscaled relative translations. Govindu [25] approaches this by minimizing the cross-product of input and derived directions. Jiang *et al*. [30] utilize geometric constraints of triangle formations among camera triplets. Moulon *et al*. [44] suggest a solution minimizing a relaxed problem via the L_∞ norm. Wilson *et al*. [64] reduces the discrepancy between input directions and those deduced from absolute translations. Tron *et al*. [61] minimize squared relative displacements in a distributed framework. Ozyesil *et*

al. [48] introduce the Least Unsquared Deviations (LUD) method to extend [61], incorporating L1 loss, framing the problem within a convex program. Arrigoni *et al*. [4] minimize the squared error from the orthogonal projection of estimated relative translations onto the input directions. Similarly, Goldstein *et al*. [24] minimize orthogonal projections through ADMM, adopting L1 loss for robustness. Zhuang *et al*. [72] relax the cost metrics from [64] by aligning estimated relative translations with observed directions. Other methods include leveraging two- and three-view camera geometry [2, 26, 44], determining edge scales via cycles in a network before solving for absolute translations [3, 5], employing point correspondence constraints [13, 16], refining directions iteratively [40], averaging matrices from two-view geometries [32, 33], and making use of the matrix structure resulting from pairwise displacements [19]. GLOMAP [49] proposes to replace translation averaging with global positioning, ignoring the estimated relative translations entirely. While these algorithms are crucial to getting accurate camera positions, they often fail due to degeneracies and outliers in the estimated pose graph.

Pose graph filtering serves as a strategy for removing inaccurate relative poses, thereby eliminating outliers to enhance the robustness of pose averaging. Since translation averaging tends to be more prone to noise and outliers than rotation estimation in practice, most filtering methods assume the rotations to be pre-estimated and focus on translations. Zach *et al*. [68] introduce a method to exclude outlier edges based on loop consistency. 1DSfM [64] involves projecting relative translations onto randomly selected 3D directions and then evaluating discrepancies within this one-dimensional subspace. This random projection is iterated multiple times to accumulate instances of inconsistency for each edge. Such accumulated inconsistency metrics are then employed to eliminate discordant edges across multiple random projections. Shen *et al*. [53] focus on identifying and retaining a subset of reliable edges to ensure strong connectivity among cameras. The more recent study by Manam *et al*. [41] highlights that the main issue often lies not with the outlier edges themselves but with edges that contribute to forming skewed triangles, which can lead to degenerate conditions in translation averaging. Our method focuses on filtering outlier edges and can be straightforwardly combined with any other filters, such as [41].

Recently, **learning-based approaches** have emerged, primarily to solve the rotation averaging problem. NeuRoRa [50] introduces a dual-network system: one for cleaning the view graph of relative rotations and another for fine-tuning, optimizing absolute orientations derived from the cleaned graph. MSP [67] proposes an end-to-end framework that optimizes orientations using multiple measurements and incorporates neighborhood information. PoGO-Net [39] presents a novel pose graph optimization (PGO) strategy implemented through a Graph Neural Network, aiming to estimate camera orientations accurately. DMF-synch [59] utilizes a matrix factorization technique for pose extraction. While these methods show promising results, they primarily focus on rotation estimation. Contrarily, we argue that rotation averaging, in practice, tends to be significantly more accurate than estimating camera positions. Consequently, our work concentrates on enhancing translation averaging by filtering outlier relative translation edges. We employ a line graph representation to frame outlier filtering as a binary node classification problem. Line graphs have been used in Graph Neural Networks for link prediction where edges in the original subgraph around two vertices are converted into nodes to better capture structural relationships [9]. In contrast, we leverage the line graph representation for classification in Structure-from-Motion to filter the pose graph from potential outliers.

3. Filtering Outlier Edges in a Pose Graph

Problem Statement. Suppose we are given a set of images $\mathcal{I}$ and a pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \subseteq \mathcal{I}$ denotes the images (or views) in the graph, and $\mathcal{E} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ represents the edges between views. We define a mapping $T : \mathcal{E} \rightarrow \mathrm{SE}(3)$ such that $T(v_i, v_j) = (\mathbf{R}, \mathbf{t})$ is the estimated relative transformation from view v_i to v_j, where $(v_i, v_j) \in \mathcal{E}$, $\mathbf{R} \in \mathrm{SO}(3)$ is the relative rotation, and $\mathbf{t} \in \mathbb{R}^3$ is the unit-length relative translation vector ($\mathbf{t}^\top \mathbf{t} = 1$). The construction of the pose graph $\mathcal{G}$, a preliminary step in both global and incremental SfM methods, involves image retrieval, feature detection and matching, and robust two-view geometry estimation [7, 52]. An overview of global SfM pipelines is depicted in Fig. 2.

After constructing the initial pose graph $\mathcal{G}$, methods typically proceed as follows. First, two-view geometries are filtered, *e.g*., by discarding edges with an insufficient number of inliers (*e.g*., fewer than 30 [58]). Next, rotation averaging is performed – using, for instance, the robust method of Chatterjee *et al*. [10] – to determine the absolute camera orientations independently of their positions. Rotation averaging generally achieves higher accuracy than position averaging, with errors typically within a few degrees [39]. Therefore, for the remainder of this paper, we assume known camera orientations $\mathbf{R}_v$ for all $v \in \mathcal{V}$ and focus on estimating the global camera translations $\mathbf{t}_v$.

In global SfM, following rotation averaging, translations are re-estimated based on the fixed orientations. Outlier edges are then filtered to ensure consistency or eliminate degeneracy, *e.g*., by projecting onto 1D subspaces [64] or identifying degenerate motions [41]. Subsequent steps include translation averaging to derive camera positions from the refined pose graph, triangulating points to obtain a point cloud, and performing bundle adjustment to jointly optimize all parameters by minimizing reprojection errors.

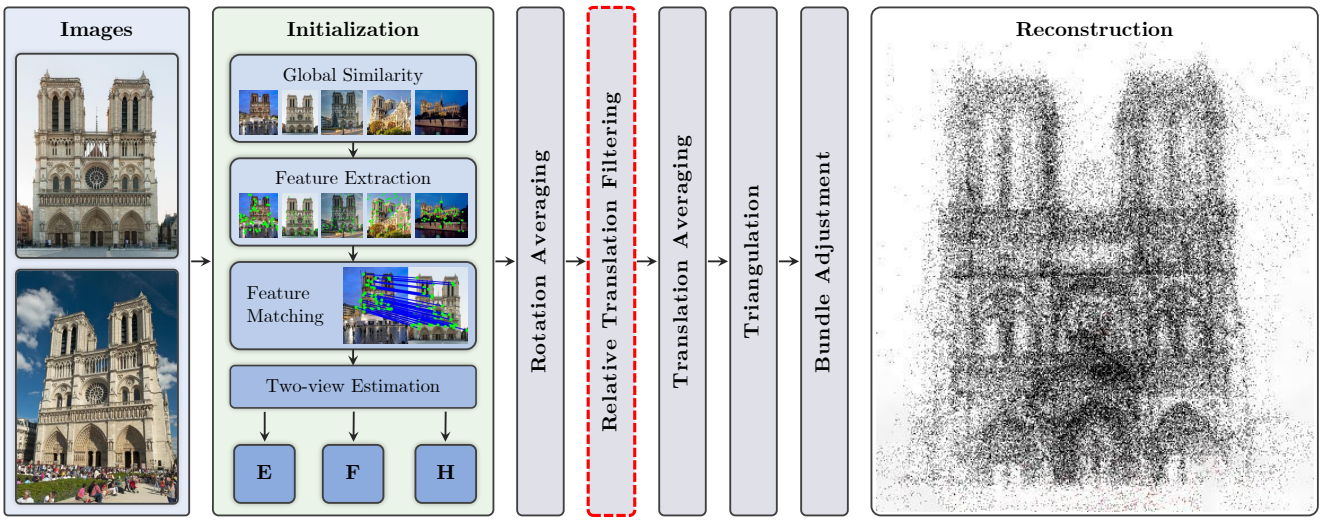

Figure 2. In **global Structure-from-Motion**, an image collection is fed into the initialization phase that selects overlapping image pairs, performs feature detection and matching, and, finally, obtains relative poses via homography (H), essential (E) and fundamental matrix (F) estimation. Next, rotation averaging runs to obtain the global camera orientations. Then the pose graph edges are filtered to remove outliers (*e.g.*, by 1DSfM [64] or the proposed method) or degenerate configurations [41]. Translation averaging obtains the camera positions from the filtered graph with fixed orientations. Finally, the 3D structure is triangulated, and bundle adjustment optimizes all parameters jointly.

Line Graph Representation. In this paper, we aim to learn a filter $\theta : \mathcal{E} \rightarrow \{0, 1\}$ that identifies each edge (representing a relative translation estimate) as an inlier (1) or outlier (0). To achieve this, we utilize a Graph Neural Network (GNN). While directly learning an edge classifier is feasible [35], we observed that it yields unstable results, often leading to excessive edge removal from the pose graph and, consequently, significant loss of cameras crucial for the 3D reconstruction. To mitigate this issue, we propose converting the pose graph $\mathcal{G}$ into its line graph $L(\mathcal{G})$.

Line graph $L(\mathcal{G})$ is constructed from the original graph $\mathcal{G}$ by representing each edge in $\mathcal{G}$ as a vertex in $L(\mathcal{G})$. Specifically, if $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, then $L(\mathcal{G}) = (\mathcal{V}_L, \mathcal{E}_L)$, where each vertex $v_{ij} \in \mathcal{V}_L$ corresponds to an edge $(v_i, v_j) \in \mathcal{E}$. Two vertices v_{ij} and v_{kl} in $L(\mathcal{G})$ are connected by an edge if and only if their corresponding edges in $\mathcal{G}$ share a common vertex, *i.e.*, $\{v_i, v_j\} \cap \{v_k, v_l\} \neq \emptyset$. Mathematically, the edge set $\mathcal{E}_L$ of $L(\mathcal{G})$ is defined as follows:

$$\mathcal{E}_L = \{(v_{ij}, v_{kl}) \mid (v_i = v_k \vee v_j = v_l) \wedge (v_i, v_j), (v_k, v_l) \in \mathcal{E}\}.$$

This allows us to reformulate edge classification in $\mathcal{G}$ as vertex classification in $L(\mathcal{G})$, facilitating the application of vertex-focused GNN architectures for outlier detection.

Outlier Filtering with a GNN. In the pose graph represented as line graph $L(\mathcal{G})$, we attribute to each vertex – corresponding to an edge in the original graph – the estimated relative pose, consisting of a 3D rotation and an unscaled translation. Two vertices (v_i, v_j) and (v_j, v_k) in the line graph $L(\mathcal{G})$ are connected by an edge if they share a common node in the original graph. We assign the edge between (v_i, v_j) and (v_j, v_k) in $L(\mathcal{G})$ two features concatenated into a single vector: The 3D global orientation $\mathbf{R}_{v_j}$ of the shared node and the corresponding image embedding $\mathbf{d}_{v_j}$.

We observed that leveraging context from the images helps to recognize outlier edges. We obtain the embeddings $\mathbf{d}_{v_i}$ and $\mathbf{d}_{v_j}$ using DINOv2 [47], which yields 384-dimensional image embeddings. Consequently, the concatenated feature embedding vector assigned to each vertex in $L(\mathcal{G})$ is of dimension 393. DINOv2 features capture the content of images, so neighboring images should have similar features, allowing the incorporation of high-level information directly into the view graph. If two images are connected in the graph, we expect them to have overlapping views and, thus, similar DINOv2 features [34, 63]. Often, such descriptors are already available from running image retrieval at the beginning of the SfM pipeline.

Next, we discuss the network structure. Our Graph Neural Network employs three `TransformerConv` layers with ReLU activations and dropout operations (with a probability of 0.3) and has 35,992 parameters in total. The `TransformerConv` is a Graph Convolutional Network (GCN) layer enhanced with a self-attention mechanism. It integrates edge attributes in the computation of the attention coefficients and in the node update process. The `TransformerConv` operator [54] is implemented in the PyTorch Geometric Library. The node update mechanism, detailed in Eq.1, combines the transformed representation of the current node x_i with aggregated information from its neighboring ones $j \in \mathcal{N}(i)$ and the connecting edges e_{ij}. The contribution of each neighbor x_j and edge e_{ij} to the updated state x'_i of the node is modulated by the attention coefficient α_{ij}, calculated using a scaled dot-product

attention mechanism. This attention framework, outlined in Eq. 2, leverages the query W_3x_i, the key W_4x_j, and the dimensionality of hidden channels d, with transformed edges W_6e_{ij} enriching the key:

$$x_i' = W_1x_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}(W_2x_j + W_5e_{i,j}), \tag{1}$$

where the attention weights α_{ij} are determined as:

$$\alpha_{i,j} = \text{softmax}\left(\frac{(W_3x_i)^{\mathrm{T}}(W_4x_j + W_5e_{i,j})}{\sqrt{d}}\right). \tag{2}$$

In the last layer, we employ a gated residual connection to prevent over-smoothing [54]. A parameter β_i is learned that controls how much of the previous and aggregated representations contribute to the final representation:

$$x_i' = \beta_i W_1x_i + (1-\beta_i)\underbrace{\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}(W_2x_j + W_5e_{i,j})\Big)}_{m_i}, \tag{3}$$

where $\beta_i = \text{sigmoid}\left(W_6^{\mathrm{T}}\left[W_1x_i, m_i, W_1x_i - m_i\right]\right)$.

Graph Clustering. Converting the graph $\mathcal{G}$ to $L(\mathcal{G})$ may require a large amount of memory, depending on the connectivity of $\mathcal{G}$. Specifically, the edge number in the line graph $|\mathcal{E}_L|$ is quadratic in the node degrees of $\mathcal{G}$ as:

$$|\mathcal{E}_L| = \frac{1}{2}\sum_{v\in\mathcal{V}} \deg(v)^2 - m, \tag{4}$$

where m is the node number in $L(\mathcal{G})$ and v is the original graph's vertices. This property often leads to a memory explosion in large and well-connected graphs, preventing the execution of the proposed classification network.

To overcome this issue, we employ a graph clustering technique [12] to partition the graph $\mathcal{G}$ before its conversion to the line graph $L(\mathcal{G})$. This clustering allows us to control the maximum number of edges in the line graphs and, thus, the maximum storage requirements of the proposed method. The approach involves partitioning $\mathcal{G}$ into k clusters, $\{C_1, C_2, \ldots, C_k\}$, where each cluster C_i contains a subset of vertices from $\mathcal{V}$. This partitioning is designed to minimize the inter-cluster edge cut while balancing the cluster sizes. Formally, the objective is to solve:

$$\min \sum_{i=1}^{k} \text{cut}(C_i, \mathcal{V}\setminus C_i), \tag{5}$$

where $\text{cut}(C_i, \mathcal{V}\setminus C_i)$ denotes the total weight of edges that connect cluster C_i to the rest of the graph. We set the edge weights equal to the number of inliers found by RANSAC for the corresponding relative pose, which proved to be a good indicator of edge quality. During inference, an edge may belong to multiple subgraphs, thus receiving more than one inlier/outlier vote. To make a final decision, majority voting is applied. The algorithm is further detailed in the appendix supplementary material.

The resulting clusters are then processed separately. Each cluster $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ is converted to its line graph $L(\mathcal{G}_i)$, significantly reducing the memory requirements compared to converting the entire graph $\mathcal{G}$. This clustering-based approach mitigates the memory explosion problem while preserving the essential topological and connectivity information from the original graph, enabling the effective application of the proposed classification network on large and densely connected graphs.

4. Experiments

Implementation. Our filter is incorporated both in the GLOMAP [49] and Theia libraries [58], using their default settings. In both pipelines, relative pose estimation is performed using LO$^+$-RANSAC [37]. The method of Chatterjee *et al*. [10] performs rotation averaging. GLOMAP uses the BATA loss [72] to estimate the camera positions, while Theia employs the LUD algorithm [48]. Even though GLOMAP performs global positioning instead of translation averaging, the proposed filtering still applies before the positioning step, removing edges from the pose graph. For calculating the error metrics, the reconstructions are robustly aligned to reference data using RANSAC.

Baselines. Our method is compared with various translation filters within the same framework, including:

1. *No filter*: Translation averaging without any filtering.
2. *MSAC score*: To emulate the process proposed in [40], we assign a weight to each point correspondence by calculating the MSAC score from its Sampson error, given the estimated relative pose. The weight of an edge is the sum of these weights, similar to the approach in [40]. We filter edges by removing those with a score lower than 30. Note that we cannot use the weights directly from [40] as they require absolute camera positions.
3. *1DSfM*: Utilizes the filter from [64] to eliminate incorrect translations through projections into 1D subspaces.
4. *Triangle*: Employs a filter from [41] targeting the removal of degenerate triplet configurations rather than outliers. This method can be straightforwardly combined with the proposed one, as we will demonstrate.
5. *CleanNet*: An outlier detection network that filters outliers in the initial stage of the NeuRoRA framework [50].
6. *Oracle*: Shows the potential maximum accuracy of an ideal filter by excluding translations with errors exceeding 20° (Theia) and 10° (GLOMAP) w.r.t. ground truth. Note that while the accuracy of Oracle may be the upper bound without outliers, it can be surpassed by removing degenerate configurations.

Datasets. Evaluations were conducted on the 1DSfM [64], PhotoTourism [56], and ScanNet [17] datasets. The 1DSfM dataset, encompassing 15 landmark scenes with internet-

Table 1. Mean position errors in meter and recall thresholded at 10m for reconstructions by Theia [58] on the **1DSfM dataset** [64], using only default filters (w/o Filter), MSAC score [40], CleanNet [50], Triangle filter [41], 1DSfM [64], 1DSfM combined with Triangle filter, and our proposed method, both standalone and in conjunction with Triangle filter. Additionally, the results of the Oracle filter, removing all outlier translations, are presented. On M. N. Dame, [40] and our method could not be aligned with the COLMAP reconstruction for evaluation due to reconstructing a different set of cameras. Best results are in bold, and the second-best ones are underlined.

Scene	Alamo	E. Isl.	Gendar.	M. Metro	M. N. Dame	NYC Lib.	N. Dame	P. del Pop.	Piccad.	R. Forum	T. of London	Trafal.	U. Square	V. Cath.	Yorkm.	AVG
Mean position error (↓)																
Oracle	7.69	20.91	41.38	23.24	6.37	5.25	4.83	17.77	12.49	18.54	30.43	2082.17	17.08	13.06	10.83	164.69
w/o Filter	10.99	96.48	46.44	30.61	8.52	17.43	5.44	20.38	17.45	25.39	53.18	1749.00	29.68	18.14	39.47	154.29
MSAC score	11.07	96.56	48.95	30.10	–	21.54	5.46	19.18	17.01	25.57	99.22	1762.31	30.30	18.24	38.18	158.83
CleanNet	10.86	96.84	54.48	29.94	8.61	16.29	5.45	19.10	17.16	25.50	48.51	1763.33	30.41	17.74	42.43	155.57
Triangle	6.66	**19.82**	46.38	21.75	**6.08**	6.36	5.45	**18.56**	11.06	22.15	36.57	78.22	14.95	13.86	14.78	22.61
1DSfM	11.02	96.46	49.87	30.28	11.10	17.36	5.46	19.11	17.73	27.53	55.97	1742.24	29.71	18.86	37.28	154.21
1DSfM+Tri.	6.77	19.87	51.18	23.02	9.77	**6.01**	5.32	19.26	**9.69**	23.95	41.78	**75.03**	14.75	**13.74**	16.34	23.34
Ours	10.80	23.69	**42.89**	27.80	–	10.81	5.44	19.00	16.90	23.57	39.80	73.07	27.43	16.29	40.03	26.97
Ours+Tri.	**6.62**	21.56	43.85	**19.95**	–	6.06	**5.29**	19.16	10.90	**21.18**	**29.08**	73.29	**14.18**	14.02	**10.95**	**21.15**
Position Recall@10m (↑)																
Oracle	76.74	45.34	28.45	38.03	79.55	88.94	89.39	50.12	80.51	48.87	34.83	8.93	61.17	59.69	77.98	56.36
w/o Filter	65.18	37.28	13.53	25.12	**77.73**	49.63	**87.43**	37.38	64.51	**35.29**	19.90	13.04	40.22	41.65	23.83	39.57
MSAC score	65.05	37.28	**21.24**	23.24	–	52.58	**87.43**	37.74	64.66	34.46	4.48	14.28	37.29	39.59	23.47	38.77
CleanNet	66.36	38.70	15.93	25.12	75.30	52.33	87.13	37.62	65.29	34.23	19.07	6.96	37.60	41.75	25.27	39.53
Triangle	**69.78**	**44.39**	7.96	28.64	71.05	**75.18**	87.06	38.09	64.88	37.76	23.05	14.76	41.91	**42.27**	49.10	44.18
1DSfM	65.44	37.28	17.32	23.71	64.57	49.63	87.13	37.38	63.73	28.45	17.25	5.41	41.76	41.03	23.10	38.47
1DSfM+Tri.	68.46	42.81	18.84	28.17	69.64	72.24	87.06	37.50	61.94	30.93	14.93	15.15	**42.99**	41.24	43.68	43.28
Ours	64.91	41.55	14.16	25.59	–	60.44	87.06	**38.92**	**65.85**	32.06	20.9	15.27	40.06	35.26	28.88	40.78
Ours+Tri.	68.46	36.65	18.33	**29.11**	–	70.76	87.21	35.14	63.58	33.33	**25.04**	**15.49**	42.37	**42.27**	**54.15**	**44.42**

Table 2. Median position errors in meters and recalls of GLOMAP [49], on the 1DSfM dataset [64], using only its default filters and when combined with the proposed method.

Filtering	Median position error (↓)	Position Recall@10m (↑)
Default	9.56	59.39
Default + Ours	**8.59**	**61.68**
Oracle	7.41	68.69

sourced photos, includes two-view matches, epipolar geometries, and a reference incremental SfM reconstruction (via Bundler [56, 57]) for error analysis. Given that Bundler is now considered outdated, we generated new reference reconstructions using COLMAP [52]. To ensure that the reconstruction is approximately metric, we robustly align it with the provided Bundler reconstruction. The scenes used for the PhotoTourism dataset are based on the CVPR Image Matching Challenge 2020 [31]. These scenes are non-metric; thus, the reported position errors are *not* in meters. The ScanNet dataset [17] consists of 1,613 monocular sequences with ground truth poses and metric depth. To evaluate relative translation filters, we utilize the same six sequences as those used by Zhu *et al*. [71].

Training. We trained the proposed method on scene Piazza San Marco from the PhotoTourism dataset, comprising 249 images and 10,295 view graph edges from Theia in total. We use the COLMAP reconstruction to provide target inlier/outlier labels. We label a relative translation an outlier if its error is higher than 20° in Theia, leading to 4,418 outlier edges and good performance in our experiments. We train the same model on the pose graph generated in GLOMAP. Here we use a threshold of 10° for outlier translations which leads to 6,395 outlier edges in Piazza San Marco pose graph of 12,755 edges. As the validation set, we use scene Taj Mahal. We exclude these scenes from the main experiments. The model is trained for 300 epochs, with a learning rate of 0.009, using Binary Cross Entropy Loss, weight decay of 5×10^{-4}, and the Adam optimizer [36], on three subgraphs shuffled during training. We use this model in *all* experiments and on all datasets, demonstrating the generalization capabilities of the proposed method. Specifically, we train it on a *single* outdoor scene and test it on three large-scale datasets with significant domain gaps (indoor/outdoor), showing that it performs accurately across different noise and outlier distributions. We performed hyper-parameter tuning for the baselines on the same training data for a fair comparison.

The **metrics** include mean position errors and recall rates at thresholds of 1, 5, and 10. We report the mean error as it clearly shows large failures in the reconstruction. We include recall as a robust metric to show the accuracy of the well-reconstructed cameras. Other metrics are reported in the supp. mat. To make the recall rates fair across methods, we consider the reconstructed cameras after applying the Oracle filter and calculate the accuracy on these ones. For each camera missing from the reconstruction with a particular filter, we consider the error to be infinity for recall calculation. On 1DSfM and ScanNet, the errors are in meters, as the reference is a metric reconstruction. There, we report the recall thresholded at 10 and 1 meters. On PhotoTourism, the errors have no units; thus, we report the recall at a threshold of 5, chosen to make the results meaningful.

1DSfM. Results on the 1DSfM dataset are shown in Table 1. The proposed filtering significantly surpasses the

Table 3. Mean position errors and recall thresholded at 5 for reconstructions by Theia [58] on the **PhotoTourism dataset** [56], using unfiltered relative translations (w/o Filter), MSAC score [40], CleanNet [50], Triangle filter [41], 1DSfM [64], 1DSfM combined with the Triangle filter, and our proposed method, both standalone and in conjunction with the Triangle filter. Additionally, the results of the Oracle filter, removing all outlier translations, are also presented. The best results are shown in bold, and the second-best ones are underlined.

Scene	B. Museum	F. Cathedral	L. Memorial	M. Cathedral	M. Rushmore	Reichstag	Sacre Coeur	S. Familia	St. P. Cathedral	St. P. Square	AVG
Mean position error (↓)											
Oracle	2.15	1.19	1.22	1.87	2.47	2.17	1.35	1.35	1.99	2.38	1.81
w/o Filter	2.83	2.26	**2.49**	**1.97**	**2.79**	2.66	1.95	1.77	2.29	2.79	2.38
MSAC score	2.75	2.22	2.56	1.99	**2.79**	2.60	1.97	1.77	2.32	**2.78**	2.37
CleanNet	2.53	2.25	2.50	1.99	**2.79**	2.64	1.95	1.76	**2.28**	2.79	2.35
Triangle	2.83	2.26	**2.49**	2.01	**2.79**	2.66	1.94	1.78	2.29	2.79	2.38
1DSfM	**2.52**	2.27	2.58	1.99	**2.79**	2.60	1.95	1.85	2.35	2.79	2.37
1DSfM+Tri.	2.88	2.25	2.58	2.01	**2.79**	2.63	1.96	1.77	2.30	2.82	2.40
Ours	**2.52**	2.14	**2.49**	1.98	**2.79**	2.69	**1.86**	**1.74**	2.29	2.79	2.33
Ours+Tri	**2.52**	**1.89**	**2.49**	1.98	**2.79**	**2.42**	**1.86**	**1.74**	2.29	2.79	**2.28**
Position Recall@5 (↑)											
Oracle	94.23	98.04	98.94	93.39	84.96	90.67	91.76	95.71	92.45	93.08	93.32
w/o Filter	81.64	88.24	85.87	90.91	**85.71**	86.67	81.65	93.69	**91.46**	89.52	87.54
MSAC score	86.04	87.25	**92.82**	90.91	**85.71**	89.33	81.05	93.94	90.48	88.32	88.59
CleanNet	91.20	87.25	86.10	89.26	**85.71**	89.33	81.73	93.94	**91.46**	90.04	88.60
Triangle	83.61	**89.22**	92.70	90.08	**85.71**	**90.67**	81.82	93.94	90.97	89.64	88.84
1DSfM	**92.11**	**89.22**	87.28	**91.74**	**85.71**	88.00	81.73	92.42	89.98	90.04	88.82
1DSfM+Tri.	81.18	88.24	88.10	89.26	**85.71**	88.00	81.65	93.94	90.97	88.44	87.55
Ours	91.35	**89.22**	92.34	90.91	**85.71**	89.33	82.58	**94.70**	90.97	**90.12**	**89.72**
Ours+Tri	91.35	82.35	90.08	85.71	85.33	84.00	**94.95**	93.69	90.97	**90.12**	88.86

conventional outlier filtering technique, 1DSfM, in performance. Notably, it reduces the average position error by approximately a factor of 5 and enhances the recall rate by approximately 6%. The MSAC score baseline has significantly higher error implying that it fails to filter incorrect pose graph edges effectively. CleanNet only has a minor impact on accuracy. The combined Ours+Triangle method attains the highest recall and lowest mean errors.

The results when the proposed filter is integrated into the GLOMAP [49] pipeline are reported in Table 2. The proposed filtering leads to a 1 meter decrease in the median position error while also improving the recall. Other results with GLOMAP are in the supplementary material.

PhotoTourism. The results on the PhotoTourism dataset are shown in Table 3. MSAC score, CleanNet, the standalone Triangle, the 1DSfM filters, and their combination have a negligible impact. On average, upon integration with the Triangle filter, the proposed method outperforms the baselines in all accuracy metrics. This shows that the two filters complement each other; the proposed one removes outliers, and the Triangle filter deals with degenerate configurations. Employed independently of the Triangle filter, our approach secures best or second-best results. As expected, the Oracle filter achieves superior performance on nearly all scenes. However, it is crucial to highlight that its excellence is in outlier removal and does not extend to identifying degenerate configurations, which could adversely impact reconstruction quality. Moreover, the threshold of 20° employed for removing incorrect translations may not be optimal in all scenes. These are the reasons why the Oracle filter does not always lead to the best reconstructions.

ScanNet. The performance on six scenes from ScanNet [17], as selected by [71], is reported in Table 4. The trends are similar to those observed on other datasets. MSAC score is not improving over the baselines. Notably, CleanNet exhibits negligible enhancement over the scenario without any filter applied. Interestingly, the conventional outlier filter, 1DSfM, decreases the accuracy on average. However, combined with the Triangle filter, it enhances accuracy across all metrics. The proposed filter, on the other hand, improves performance across all accuracy metrics compared to the unfiltered approach, even without the Triangle filter. The proposed method achieves a recall rate that is approximately 10% higher than 1DSfM. When integrated with the Triangle filter, it attains the lowest mean errors and the highest recalls.

Runtime. The average run-time of the proposed method is 8.6 mins. on PhotoTourism, 2.55 mins. on 1DSfM, and 4.42 mins. on ScanNet. The times on all scenes are in the supp. mat. Although filters like 1DSfM [64] are faster (1.8 secs. PhotoTourism, 0.6 secs. 1DSfM, 1.2 secs. ScanNet), it is crucial to highlight that none of the filtering methods, including 1DSfM filtering, nor SfM itself are designed for real-time performance. Despite this, Theia combined with our filtering method remains orders-of-magnitude faster than an incremental approach like COLMAP. For instance, on 11 scenes of the 1DSfM dataset, Theia takes an average of 23 minutes for reconstruction and GLOMAP 138 min-

Table 4. Mean position errors in meter and recall thresholded at 1 meter for reconstructions by Theia [58] on the **ScanNet dataset** [17], using unfiltered relative translations (w/o Filter), MSAC score [40], CleanNet [50], the Triangle filter [41], 1DSfM [64], 1DSfM combined with the Triangle filter, and our proposed method, both standalone and in conjunction with the Triangle filter. Additionally, the results of the Oracle filter, removing all outlier translations, are also presented. The best results are shown in bold, and the second-best ones are underlined. On Scene 0207, Oracle reconstructs a different set of cameras than the proposed filter; thus, we exclude this scene from the average recall and report the unnormalized recall on it.

Scene	0000	0059	0106	0169	0181	0207	AVG
Mean position error (↓)							
Oracle	0.49	0.61	0.39	0.45	0.64	0.37	0.49
w/o Filter	0.65	**0.58**	0.87	0.58	1.21	0.81	0.78
MSAC score	**0.62**	1.20	1.03	1.27	1.26	1.13	1.09
CleanNet	0.65	**0.58**	0.87	0.58	1.21	0.81	0.78
Triangle	0.65	0.60	0.87	**0.48**	1.17	0.59	0.73
1DSfM	0.69	**0.58**	1.08	0.55	1.26	0.80	0.83
1DSfM+Tri.	0.69	0.59	0.90	**0.48**	**0.70**	0.61	0.66
Ours	0.66	**0.58**	0.85	0.63	1.12	0.55	0.73
Ours+Tri.	0.66	0.60	**0.57**	0.50	1.02	**0.36**	**0.62**
Position Recall@1m (↑)							
Oracle	90.13	90.94	99.38	94.41	86.07	100.00	92.19
w/o Filter	90.30	89.06	91.05	92.81	24.65	58.88	77.57
MSAC score	**94.12**	64.33	54.98	53.14	36.27	47.38	60.57
CleanNet	90.20	89.06	91.05	92.97	24.65	58.71	77.59
Triangle	90.24	89.06	90.94	93.71	28.03	97.97	78.40
1DSfM	82.66	**90.35**	72.87	**93.93**	20.45	58.38	72.05
1DSfM+Tri.	82.18	89.47	85.85	92.91	**75.52**	96.79	85.19
Ours	87.43	88.95	84.70	91.10	42.87	95.64	79.01
Ours+Tri.	86.86	88.77	**94.12**	92.43	66.36	**99.49**	**85.71**

utes, compared to COLMAP's 886 minutes. Thus, spending a few additional minutes on filtering is inconsequential in an SfM pipeline.

Camera Numbers. The average number of cameras retained across the tested datasets for each method is as follows: w/o Filter (1618), MSAC score (1612), CleanNet (1590), Triangle (1356), 1DSfM (1617), 1DSfM + Triangle (1318), Ours (1463), Ours + Triangle (1263), and Oracle (1341). When combined with the Triangle filter, the proposed method retains fewer cameras than the Oracle method. This can happen because the Oracle filter removes only outlier edges, while the Triangle filter also discards degenerate edges, which may not be outliers yet degrade translation averaging. Thus, combining the proposed method and the Triangle filter can remove more edges than the Oracle method. However, it is important to note that the reported recalls were calculated on the same set of cameras returned by the Oracle filter. The combination of our method with the Triangle filter achieves the best recalls while retaining the fewest cameras, indicating its effectiveness in removing inaccurate poses from the graph – specifically, cameras that could not be recovered accurately.

Table 5. Ablation study averaged over scenes T. of London, M. Rushmore, and E. Island. We report the mean position errors, the recall at 10, and the number of cameras.

	Mean Position Error (↓)	Recall@10 (↑)	# of cameras (↑)
1DSfM	51.74	48.75	**740**
Ours w/o image features	37.78	46.14	464
Ours w/ 2 layers	24.39	47.33	565
Ours trained with 8 clusters	22.49	48.14	441
Ours inference w/o clustering	46.60	44.07	686
Proposed	**22.09**	**51.39**	548

Ablation studies. To gain a more nuanced understanding of the proposed filtering, we run the following configurations on scenes T. of London, M. Rushmore, and E. Island, as presented in Table 5. *Ours w/o image features* removes the image embeddings from the network. Only two layers in the graph neural network instead of three are used in *Ours w/ 2 layers*. The next ablation explores how changing the number of clusters during training affects performance. In *Ours trained with 8 clusters*, we increase the number of clusters from the usual three to eight. In *Ours inference w/o clusters*, we do not employ our clustering method, so the entire graphs are used for inference. Additionally, we show the results of the entire pipeline with 1DSfM filtering as baseline. In the supplementary material, we provide additional ablations showing the importance of the network inputs.

Even without image features, the proposed method improves the mean position error upon the 1DSfM baseline. Reducing the number of layers has a minor effect on the accuracy. Artificially increasing the number of clusters during training leads to high camera loss due to splitting the graph into too small subgraphs. *Ours inference w/o clustering* runs the method on entire graphs to evaluate whether clustering introduces any approximation or loss of information. While it reconstructs the most cameras after the baseline, it has lower recall and a higher mean error compared to our proposed method. This shows that clustering followed by majority voting does not degrade performance, but improves accuracy across all metrics by focusing on local subgraphs to detect outlier edges.

5. Conclusion

This paper presents a novel filtering method that enhances camera position estimation in global SfM, demonstrating improved accuracy across a diverse set of datasets. By jointly addressing outliers with the proposed method and degenerate configurations by [41], our approach ensures superior reconstruction quality. Although it is slower than other alternatives, the additional time required remains negligible compared to the overall Structure-from-Motion pipeline. The proposed method without [41] is superior, in terms of accuracy, to the standard outlier filtering techniques, e.g., 1DSfM [64]. These results highlight the critical role of advanced filtering in Structure-from-Motion.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5297–5307, 2016. 1

[2] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *Int. Conf. on 3D Imag., Model., Proc., Vis. & Trans.*, pages 81–88. IEEE, 2012. 2, 3

[3] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. On computing the translations norm in the epipolar graph. In *Int. Conf. on 3D Vis.*, pages 300–308. IEEE, 2015. 3

[4] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Robust and efficient camera motion synchronization via matrix decomposition. In *Image Anal. and Proc.*, pages 444–455. Springer, 2015. 1, 3

[5] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. In *Int. Conf. on 3D Vis.*, pages 546–555. IEEE, 2016. 3

[6] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1304–1312, 2020. 1

[7] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14546–14555, 2021. 3

[8] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in RANSAC. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15744–15753, 2022. 1

[9] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5103–5113, 2021. 3

[10] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Int. Conf. Comput. Vis.*, pages 521–528, 2013. 2, 3, 5

[11] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):958–972, 2017. 2

[12] Yu Chen, Shuhan Shen, Yisong Chen, and Guoping Wang. Graph-based parallel large scale Structure-from-Motion. *Pattern Recognition*, 107:107537, 2020. 5

[13] Hainan Cui, Shuhan Shen, and Zhanyi Hu. Robust global translation averaging with feature tracks. In *Int. Conf. Pattern Recog.*, pages 3727–3732. IEEE, 2016. 3

[14] Hainan Cui, Xiang Gao, Shuhan Shen, and Zhanyi Hu. HSfM: Hybrid Structure-from-Motion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1212–1221, 2017. 1

[15] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *Int. Conf. Comput. Vis.*, 2015. 1, 2

[16] Zhaopeng Cui, Nianjuan Jiang, Chengzhou Tang, and Ping Tan. Linear global translation estimation with feature tracks. *arXiv preprint arXiv:1503.01832*, 2015. 3

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017. 5, 6, 7, 8

[18] Frank Dellaert, David M Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: global optimality by surfing $SO(p)^n$. In *Eur. Conf. Comput. Vis.*, pages 292–308. Springer, 2020. 2

[19] Qiulei Dong, Xiang Gao, Hainan Cui, and Zhanyi Hu. Robust camera translation estimation via rank enforcement. *IEEE Transactions on Cybernetics*, 52(2):862–872, 2020. 3

[20] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 127–135, 2018. 2

[21] David Filliat. A visual bag of words method for interactive qualitative localization and mapping. pages 3921–3926. IEEE, 2007. 1

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981. 1

[23] Johan Fredriksson and Carl Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In *Asian Conf. Comput. Vis.*, pages 245–258. Springer, 2012. 2

[24] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable Structure-from-Motion. In *Eur. Conf. Comput. Vis.*, pages 289–304. Springer, 2016. 3

[25] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages II–II. IEEE, 2001. 1, 2

[26] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3

[27] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3041–3048. IEEE, 2011. 2

[28] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *Int. J. Comput. Vis.*, 103:267–305, 2013. 2

[29] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3287–3295, 2015. 1

[30] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *Int. Conf. Comput. Vis.*, pages 481–488, 2013. 1, 2

[31] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *Int. J. Comput. Vis.*, 129(2):517–547, 2021. 6

[32] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. Algebraic characterization of essential matrices and their averaging in multiview settings. In *Int. Conf. Comput. Vis.*, pages 5895–5903, 2019. 3

[33] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. Gpsfm: Global projective sfm using algebraic constraints on multi-view fundamental matrices. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3264–3272, 2019. 3

[34] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *Robo. and Auto. Let.*, 2023. 4

[35] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11–20, 2019. 4

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[37] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac–full experimental evaluation. In *Brit. Mach. Vis. Conf.* Citeseer, 2012. 5

[38] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 2

[39] Xinyi Li and Haibin Ling. Pogo-net: pose graph optimization with graph neural networks. In *Int. Conf. Comput. Vis.*, pages 5895–5905, 2021. 3

[40] Lalit Manam and Venu Madhav Govindu. Correspondence reweighted translation averaging. In *Eur. Conf. Comput. Vis.*, pages 56–72. Springer, 2022. 3, 5, 6, 7, 8

[41] Lalit Manam and Venu Madhav Govindu. Sensitivity in translation averaging. *NeurIPS*, 36, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[42] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Soc. for Ind. and App. Math.*, 11(2):431–441, 1963. 2

[43] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8. IEEE, 2007. 2

[44] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable Structure-from-Motion. In *Int. Conf. Comput. Vis.*, pages 3248–3255, 2013. 1, 2, 3

[45] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *Int. Work. on Repro. Res. in Pat. Recog.*, 2016. 2

[46] Carl Olsson and Olof Enqvist. Stable Structure-from-Motion for unordered image collections. In *Scand. Conf. on Image Anal.*, pages 524–535. Springer, 2011. 2

[47] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 4

[48] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2674–2683, 2015. 1, 3, 5

[49] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *Eur. Conf. Comput. Vis.*, 2024. 1, 2, 3, 5, 6, 7

[50] Pulak Purkait, Tat-Jun Chin, and Ian Reid. Neurora: Neural robust rotation averaging. In *Eur. Conf. Comput. Vis.*, pages 137–154. Springer, 2020. 3, 5, 6, 7, 8

[51] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1

[52] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4104–4113, 2016. 3, 6

[53] Tianwei Shen, Siyu Zhu, Tian Fang, Runze Zhang, and Long Quan. Graph-based consistent matching for Structure-from-Motion. In *Eur. Conf. Comput. Vis.*, pages 139–155. Springer, 2016. 3

[54] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020. 4, 5

[55] Chitturi Sidhartha and Venu Madhav Govindu. It is all in the weights: Robust rotation averaging revisited. In *Int. Conf. on 3D Vis.*, pages 1134–1143. IEEE, 2021. 2

[56] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH*, pages 835–846. 2006. 1, 5, 6, 7

[57] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vis.*, 80:189–210, 2008. 1, 6

[58] Chris Sweeney. Theia multiview geometry library: Tutorial & reference. `http://theia-sfm.org`. 1, 2, 3, 5, 6, 7, 8

[59] Gk Tejus, Giacomo Zara, Paolo Rota, Andrea Fusiello, Elisa Ricci, and Federica Arrigoni. Rotation synchronization via deep matrix factorization. *arXiv preprint arXiv:2305.05268*, 2023. 3

[60] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Int. Work. on Vis. Alg.*, pages 298–372. Springer, 2000. 1

[61] Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *IEEE Conf. on Dec. and Cont. held jointly with 2009 28th Chin. Cont. Conf.*, pages 901–908. IEEE, 2009. 1, 2, 3

[62] Shimon Ullman. The interpretation of Structure-from-Motion. *Proc. of the Royal Soc. of London*, 203(1153):405–426, 1979. 1

[63] Tong Wei, Philipp Lindenberger, Jiri Matas, and Daniel Barath. Breaking the frame: Visual place recognition by overlap prediction. 2025. 4

[64] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Eur. Conf. Comput. Vis.*, pages 61–75. Springer, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[65] Kyle Wilson, David Bindel, and Noah Snavely. When is rotations averaging hard? In *Eur. Conf. Comput. Vis.*, 2016. 2

[66] Changchang Wu. Towards linear-time incremental Structure-from-Motion. In *Int. Conf. on 3D Vis.*, pages 127–134. IEEE, 2013. 1

[67] Luwei Yang, Heng Li, Jamal Ahmed Rahim, Zhaopeng Cui, and Ping Tan. End-to-end rotation averaging with multi-source propagation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11774–11783, 2021. 3

[68] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1426–1433. IEEE, 2010. 3

[69] Ganlin Zhang, Viktor Larsson, and Daniel Barath. Revisiting rotation averaging: Uncertainties and robust losses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17215–17224, 2023. 2

[70] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4568–4577, 2018. 1

[71] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-SLAM: Neural implicit scalable encoding for SLAM. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12786–12796, 2022. 6, 7

[72] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4539–4547, 2018. 1, 2, 3, 5