Revisiting Global Translation Estimation with Feature Tracks

Peilin Tao[1,2,3] Hainan Cui[1,2,3†] Mengqi Rong[1,2,3] Shuhan Shen[1,2,3†]
[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] CASIA-SenseTime Research Group
taopeilin2023@ia.ac.cn, {hncui,mengqi.rong,shshen}@nlpr.ia.ac.cn

Abstract

Global translation estimation is a highly challenging step in the global structure from motion (SfM) algorithm. Many existing methods rely solely on relative translations, leading to inaccuracies in low parallax scenes and degradation under collinear camera motion. While recent approaches aim to address these issues by incorporating feature tracks into objective functions, they are often sensitive to outliers. In this paper, we first revisit global translation estimation methods with feature tracks and categorize them into explicit and implicit methods. Then, we highlight the superiority of the objective function based on the cross-product distance metric and propose a novel explicit global translation estimation framework that integrates both relative translations and feature tracks as input. To enhance the accuracy of input observations, we re-estimate relative translations with the coplanarity constraint of the epipolar plane and propose a simple yet effective strategy to select reliable feature tracks. Finally, we demonstrate the effectiveness of our approach through experiments on urban image sequences and unordered Internet images, showcasing its superior accuracy and robustness compared to many state-of-the-art techniques.

1. Introduction

The accurate estimation of camera poses and the generation of scene point clouds from image collections are fundamental tasks in the field of 3D vision, with broad applications in areas such as autonomous driving [7, 42, 43], augmented reality [31, 39, 40], and Neural Radiance Fields [33, 51, 55]. Generally, SfM stands out as a common and effective approach for achieving these objectives. It begins by constructing a view graph [4, 5, 50], where nodes represent cameras, and edges connect cameras that share a sufficient number of feature matches. Subsequently, the camera poses are estimated, and the scene structure is triangulated.

The primary manner for camera pose estimation is incremental, such as COLMAP [44]. This method commences by carefully selecting an image pair to create an initial model. Then, images containing a sufficient number of 2D-3D correspondences are registered using the Perspective-n-Point (PnP) algorithm [17, 18]. Finally, the scene structure and camera poses are jointly estimated through an iterative process that includes triangulation, bundle adjustment (BA) [41, 52], and PnP steps. While incremental methods [44, 48, 49] exhibit exceptional precision and robustness against outliers, they are susceptible to variations in the sequence of image registration, potentially leading to error accumulation and drift [25]. Additionally, the repetitive, nonlinear bundle adjustments significantly impede efficiency, making them unsuitable for large-scale scenes.

To address these issues in incremental methods, global approaches [8, 30, 32, 58] are proposed, where all cameras are registered simultaneously by estimating global rotations and translations from relative poses. Subsequently, scene structure is triangulated and optimized through a single BA refinement, leading to substantially improved efficiency and uniform error distribution across all images. Specifically, the global poses $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ and relative poses $(\boldsymbol{R}_{ij}, \boldsymbol{t}_{ij})$ of the camera satisfy the following equations:

$$\boldsymbol{R}_j \boldsymbol{R}_i^T = \boldsymbol{R}_{ij}, \quad \frac{\boldsymbol{t}_i - \boldsymbol{t}_j}{||\boldsymbol{t}_i - \boldsymbol{t}_j||_2} = \boldsymbol{R}_j^T \boldsymbol{t}_{ij} = \boldsymbol{v}_{ij}. \quad (1)$$

The notation $\boldsymbol{v}_{ij}$ denotes relative translation in the global coordinate system. For global rotation estimation, existing methods [9, 10, 22, 29] based on the Lie algebra structure have been well-studied. By contrast, relative translation estimation is sensitive to outliers [37, 47], low-parallax feature matches [12, 30] and has scale uncertainty, which makes global translation estimation harder. Approaches that rely exclusively on relative translations are limited to registering cameras within a parallel rigid graph [3, 38] and encounter issues of degeneracy when cameras undergo collinear motion. Even if the camera motion trajectory is nearly collinear, slight perturbations in relative translations

†Corresponding author.

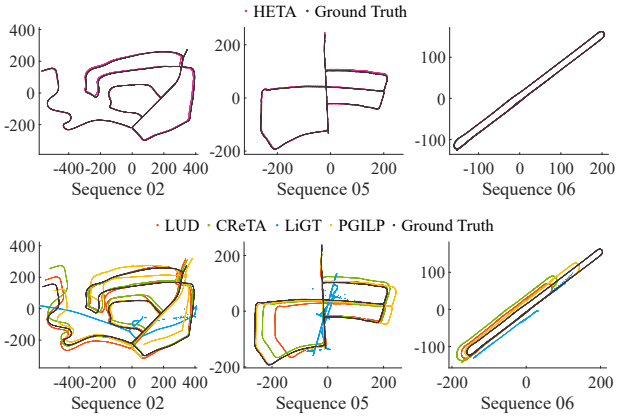

Figure 1. Camera motion trajectories of part of the KITTI [19] dataset, estimated by our HETA and several state-of-the-art methods, including LUD [38], CReTA [32], PGILP [30] and LiGT [8].

can lead to substantial changes in estimated camera positions, which makes it impossible to achieve accurate estimations with solely relative translations. To address these challenges, some methods incorporate constraints from feature tracks into their objective functions. Depending on whether the corresponding 3D points of feature tracks are estimated during optimization, these methods can be categorized as either implicit or explicit. Most implicit methods estimate camera baseline scales [14, 15, 26] or leverage camera-to-point constraints from implicit 3D points [16, 30], which are all sensitive to relative translation outliers. To address this issue, LiGT [8] aims to construct constraints with solely feature tracks. However, it lacks robustness as feature tracks generally exhibit higher outlier ratios than relative translations. A typical explicit method, 1DSfM [54], which incorporates both camera-to-camera and camera-to-point constraints to estimate 3D points and camera positions, also yields noisy solutions when dealing with feature track outliers. The failure of 1DSfM is primarily attributed to the use of an inappropriate objective function and inaccurate observations. In this study, we revisit these issues and introduce a novel hybrid explicit translation averaging framework named HETA. The term "hybrid" reflects the use of both relative translations and feature tracks as inputs.

Our contributions span three key aspects: (1) We categorize global translation estimation methods with feature tracks into explicit and implicit methods and revisit their strengths and weaknesses. (2) We perform a comparative analysis of two forms of linear objective functions and introduce a novel hybrid explicit method to concurrently estimate cameras and points in a two-step process, involving robust L_1 norm optimization followed by unbiased L_2 norm optimization. (3) To improve the accuracy of relative translations, we re-estimate them with the coplanarity constraint in epipolar geometry. To enhance the robustness of this re-estimation, we analyze the impact of parallax angles and filter out unstable feature matches. Finally, we propose a simple yet effective method for selecting reliable feature tracks.

We validate our method through experiments on both the sequential KITTI odometry benchmark [19] and the Internet dataset 1DSfM [54]. As shown in Fig. 1, our approach outperforms many state-of-the-art global SfM techniques.

2. Related Work

2.1. Global Rotation Estimation

Global translation estimation presumes the availability of global rotation estimation, which is a well-studied problem. Govindu and Chatterjee [9, 10, 22] propose to estimate global rotations in Lie algebraic structure. For a better initialization, Lee *et al*. [29] propose a hierarchical strategy and Yang *et al*. [56] present an end-to-end scheme. In both Zhang *et al*. [57] and Sidhartha *et al*. [45], the weights for relative rotations are emphasized for more accurate results.

2.2. Global Translation Estimation

Without Feature Tracks. Govindu [21] proposes a least-squared solution of global translation estimation and refines the solution with iterative weights. Several works [27, 28, 35, 46] utilize L_∞ norm-based quasi-convex optimization to estimate camera translations. However, these methods necessitate meticulous handling of outliers [38, 54]. Ozyesil *et al*. [38] use pairwise feature matches to estimate relative translations and propose a least unsquared deviations (LUD) formulation to enhance the robustness. Goldstein *et al*. [20] propose to minimize the projection of $\boldsymbol{t}_i - \boldsymbol{t}_j$ on the orthogonal complement of $\boldsymbol{v}_{ij}$ under L_1 norm by the alternating direction method of multipliers (ADMM) method. Zhuang *et al*. [59] propose an angle-based formulation with an iterative reweighted least square (IRLS) scheme to mitigate the impact of different camera baselines. Zhu *et al*. [58] introduce a distributed framework to enhance the efficiency in large-scale scenes. Manam *et al*. [32] propose an iterative averaging scheme to filter outliers and refine relative translations with re-weighted feature correspondences.
With Feature Tracks. For explicit methods, camera translations and 3D points are estimated simultaneously. For example, Crandall *et al*. [13] employ a discrete Markov Random Field formulation to estimate cameras and 3D points. Wilson *et al*. [54] propose to initially eliminate relative translation outliers through multiple one-dimensional projections and subsequently integrate both relative translations and feature tracks into a non-convex objective function. For implicit methods, 3D points are not estimated but are represented with feature rays. Then, the depths of feature points are used to impose constraints on the camera translations. Cui *et al*. [15] estimate the camera baseline scales based on a satellite graph, and then compute the camera motions by similarity averaging. Similarly, based on the adjacent triangles in feature tracks, Cui *et al*. [14] utilize the

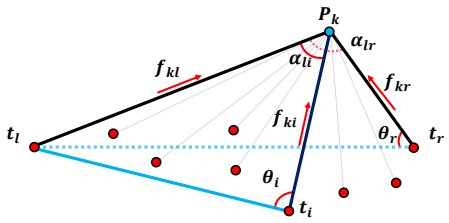

Figure 2. This figure shows a toy example of camera-to-camera and camera-to-point constraints, where red points denote cameras and blue point denotes a sample 3D point P_k. Red arrows represent the direction of feature rays. Cameras $\boldsymbol{t}_l$ and $\boldsymbol{t}_r$ have the largest parallax angle in the corresponding feature track of P_k.

law of sines to estimate the scale of camera baselines. Cui *et al*. [16] propose to represent 3D points with relative translations and feature rays based on the rotation trick [26]. Then a linear constraint is derived for cameras seeing a common scene point. However, in low parallax scenes, the represented 3D points are unstable and the relative translations are error-prone. To enhance the stability of representation, Liu *et al*. [30] propose to represent each 3D point with two cameras featuring a sufficient parallax angle. Then, a linear constraint is constructed between the represented 3D point and the remaining cameras in each feature track. Cai et al. [8] propose a linear global translation (LiGT) constraint, where 3D points are represented solely with feature rays, aiming to avoid the impact of errors in relative translations.

3. Explicit vs. Implicit

We revisit the explicit and implicit methods and conduct a thorough analysis of their strengths and weaknesses.

Assuming light rays emitted from a 3D point $\boldsymbol{P}_k$ generate n projection feature points on n camera planes. For a feature point, we denote its coordinate in the local normalized camera coordinate system as $\boldsymbol{X}_{ki}^T = (x_{ki}, y_{ki}, 1)^T$, where k is the index of feature track or 3D point, i is the index of camera. The relationship between 3D points and cameras in the global camera coordinate system satisfies:

$$\frac{\boldsymbol{P}_k - \boldsymbol{t}_i}{||\boldsymbol{P}_k - \boldsymbol{t}_i||_2} = \boldsymbol{R}_i^T \frac{\boldsymbol{X}_{ki}}{||\boldsymbol{X}_{ki}||_2} = \boldsymbol{f}_{ki}. \tag{2}$$

Here $\boldsymbol{t}_i$ denotes camera position and $\boldsymbol{f}_{ki}$ denotes the normalized feature ray from camera $\boldsymbol{t}_i$ to 3D point $\boldsymbol{P}_k$. From Eq. (1) and Eq. (2), the mathematical expressions for the camera-to-camera constraint and camera-to-point constraint are equivalent. Therefore, the core idea of explicit methods is to estimate 3D points using the same objective function employed for estimating camera translations. Compared to the error-prone relative translations in low parallax scenes, feature rays, as raw information derived from images, naturally exhibit higher precision. Hence, using feature rays in explicit methods can theoretically deliver superior performance compared to methods that rely solely on relative translations. Implicit methods primarily constrain cameras in two ways. One category of approaches [14, 15, 26] leverages the depth consistency of feature points to compute the camera baseline scales. An alternative category of methods [8, 16, 30] represents 3D points with feature tracks and constrains cameras based on these 3D points and their corresponding feature rays. For the first type, we take [14] as an example. As shown in Fig. 2, two adjacent triangles $\{P_k - t_l - t_r\}$ and $\{P_k - t_l - t_i\}$ are constructed by connecting the 3D point to its visible cameras. According to the sine theorem, the ratio of two camera baselines is:

$$\frac{||\boldsymbol{t}_l - \boldsymbol{t}_r||_2}{||\boldsymbol{t}_l - \boldsymbol{t}_i||_2} = \frac{\sin\theta_i \cdot \sin\alpha_{lr}}{\sin\theta_r \cdot \sin\alpha_{li}}. \tag{3}$$

However, this method is highly sensitive in low parallax scenes. On one hand, the relative translation estimation is inaccurate in low parallax scenes, leading to incorrect angles such as θ_i and θ_r. On the other hand, the low parallax angles, such as α_{li}, in the denominator result in numerical instability. This means that slight changes in parallax angles lead to substantial changes in the computation of the ratios. For the second type of implicit method, a linear constraint is derived for cameras seeing a common scene point. However, the representation of 3D points in [16] and [30] still relies on relative translations, whose errors are accumulated into the represented 3D points. To handle these concerns, Cai *et al*. [8] propose a LiGT constraint to linearly represent the 3D point for each feature track only by feature rays in two base cameras with the largest parallax angle. As shown in Fig. 2, for a feature track with two base cameras t_l, t_r, the depth of feature point in camera l is computed by:

$$\begin{aligned} ||\boldsymbol{P}_k - \boldsymbol{t}_l||_2 &= \frac{||\boldsymbol{t}_l - \boldsymbol{t}_r||_2 \cdot \sin\theta_r}{\sin\alpha_{lr}} = \frac{||\boldsymbol{f}_{kr} \times (\boldsymbol{t}_l - \boldsymbol{t}_r)||_2}{||\boldsymbol{f}_{kl} \times \boldsymbol{f}_{kr}||_2} \\ &= \frac{((\boldsymbol{f}_{kl} \times \boldsymbol{f}_{kr}) \times \boldsymbol{f}_{kr}) \cdot (\boldsymbol{t}_l - \boldsymbol{t}_r)}{||\boldsymbol{f}_{kl} \times \boldsymbol{f}_{kr}||_2^2}. \end{aligned} \tag{4}$$

From Eq. (4), $\boldsymbol{P}_k$ is represented by $\boldsymbol{t}_l, \boldsymbol{t}_r$ and feature rays:

$$\boldsymbol{P}_k = \boldsymbol{t}_l + \frac{\boldsymbol{f}_{kl}((\boldsymbol{f}_{kl} \times \boldsymbol{f}_{kr}) \times \boldsymbol{f}_{kr})^T}{||\boldsymbol{f}_{kl} \times \boldsymbol{f}_{kr}||_2^2}(\boldsymbol{t}_l - \boldsymbol{t}_r). \tag{5}$$

Based on Eq. (2), camera-to-point constraints are established between the implicit 3D point and the remaining visible cameras in the corresponding feature track, such as $\boldsymbol{t}_i$.

We compare these two kinds of methods in two key aspects. In terms of robustness, explicit methods typically perform better than implicit methods, since the 3D point for each feature track is optimized with all feature rays in explicit methods, while the 3D point is represented only with two feature rays from base cameras in implicit methods. In terms of efficiency, although implicit methods avoid the optimization of additional variables, the newly introduced camera-to-point constraints increase the connectivity of cameras, thereby disrupting the sparsity characteristic

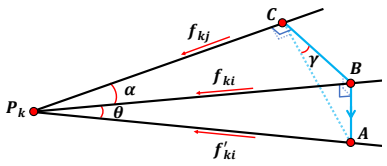

Figure 3. A toy example showing how angular errors in feature rays affect the normal vector of epipolar plane.

of the optimization matrix compared to conventional global translation averaging methods [32, 59]. As shown in our experiments, the efficiency of the explicit methods is comparable to that of the implicit methods.

4. Our Hybrid Explicit Method

Incorporating constraints from feature tracks brings a lot of benefits, but also yields a noisy solution [54] when there are many outliers in the feature tracks. In contrast, relative translations can offer more direct and stringent constraints between cameras than feature rays. To enhance robustness and efficiency, instead of just using the entire feature tracks like [8, 30], we utilize the relative translations in the view graph to directly constrain the relationship between cameras and select more reliable feature tracks to constrain the relationship between cameras and points.

A view track graph $G = \{V \bigcup P, E_v \bigcup E_p\}$ is first constructed, where each node in V represents a camera, each node in P represents a 3D point, each edge in E_v connects pair of cameras in V and each edge in E_p represents a feature ray from the camera to the 3D point. Let $\mathbb{C}$ be the camera-to-camera constraints and $\mathbb{P}$ be the camera-to-point constraints. The objective function is formulated as:

$$\min_{V;P} \sum_{E_v} \rho(||\mathbb{C}||_p) + \sum_{E_p} \rho(||\mathbb{P}||_p), \tag{6}$$

where p denotes the optimization norm and $\rho(\cdot)$ denotes the robust estimator function. Considering robustness, there are three primary tasks: (1) Enhancing the precision of relative translations in low parallax scenes; (2) Selecting reliable feature tracks; (3) Defining robust objective functions for two types of constraints. We address each of these tasks in the following subsections and present our complete optimization framework at the end.

4.1. Relative Translation Re-estimation

In two-view epipolar geometry, the coplanarity constraint is defined as: $\boldsymbol{X}_{kj} \cdot (\boldsymbol{t}_{ij} \times \boldsymbol{R}_{ij}\boldsymbol{X}_{ki}) = 0$. Given global camera rotations, this constraint is rewritten as:

$$\begin{aligned}(\boldsymbol{R}_i^T\boldsymbol{X}_{ki} \times \boldsymbol{R}_j^T\boldsymbol{X}_{kj}) \cdot \boldsymbol{R}_j^T\boldsymbol{t}_{ij} = 0 \\ \Leftrightarrow (\boldsymbol{f}_{ki} \times \boldsymbol{f}_{kj}) \cdot \boldsymbol{v}_{ij} = 0,\end{aligned} \tag{7}$$

where $\boldsymbol{v}_{ij}$ is the same as defined in Eq. (1). The detailed derivation of Eq. (7) is provided in our supplemental material. From Eq. (7), each relative translation can be re-estimated using the normal vectors of epipolar planes, which are calculated by $\boldsymbol{f}_{ki} \times \boldsymbol{f}_{kj}$. Due to inaccuracies in both camera intrinsic parameters and global rotations, the normal vectors estimated from the feature rays inevitably exhibit some angular errors. The method presented in [38] re-estimates relative translations with all normalized normal vectors by minimizing the cosine angles between the relative translations and normal vectors, which are equivalent to minimizing the sine values of the angular error in the normal vectors. However, as the accuracy of the normal vectors is also influenced by parallax angles, it is unreasonable to employ the same weight for each normal vector during the estimations. To investigate how angular errors in feature rays affect the normal vectors across varying parallax angles, we decompose them into components along both the normal direction and the epipolar plane direction. Since errors along the epipolar plane direction do not impact the direction of the normal vector, for simplicity, we exclusively consider errors along the normal direction. As shown in Fig. 3, two feature rays $\boldsymbol{f}_{ki}, \boldsymbol{f}_{kj}$ triangulate a 3D point $\boldsymbol{P}_k$ with a parallax angle of α. A minor angular error θ occurring in $\boldsymbol{f}_{ki}$ along the normal direction results in a deviation from $\boldsymbol{f}_{ki}$ to $\boldsymbol{f}'_{ki}$. We mark a point A on $\boldsymbol{f}'_{ki}$ and extend a perpendicular line from point A to $\boldsymbol{f}_{ki}$, intersecting it at point B. Subsequently, we extend another perpendicular line from point B to $\boldsymbol{f}_{kj}$, intersecting it at point C. As the angular error θ is along the normal direction, line AB is perpendicular to the plane $\{P_k - B - C\}$. Hence, line P_kC is perpendicular to plane $\{A - B - C\}$, which also means that the angle γ between AC and BC, is equal to the angular error of the normal vector. As the angle γ is expected to be small, $\sin\gamma$ approximately equals $\tan\gamma$. We have

$$\sin\gamma \approx \tan\gamma = \frac{||AB||}{||BC||} = \frac{||P_kB|| \cdot \tan\theta}{||P_kB|| \cdot \sin\alpha} = \frac{\tan\theta}{\sin\alpha}. \tag{8}$$

From Eq. (8), when θ is fixed, the angular error of the normal vector $\sin\gamma$ and $\sin\alpha$ exhibit an inverse relationship. This implies that the larger the parallax angles, the more accurate the normal vectors become. Therefore, in contrast to the method in [38] where the normal vectors $\boldsymbol{f}_{ki} \times \boldsymbol{f}_{kj}$ are normalized, we maintain the reasonable weight $||\boldsymbol{f}_{ki} \times \boldsymbol{f}_{kj}||_2 = \sin\alpha$ for each feature match during the estimations. Then, an IRLS scheme [24] is leveraged to estimate relative translations with the objective function:

$$\min_{\boldsymbol{v}_{ij}} \sum_k \rho(||(\boldsymbol{f}_{ki} \times \boldsymbol{f}_{kj}) \cdot \boldsymbol{v}_{ij}||_2) \quad s.t. \quad ||\boldsymbol{v}_{ij}||_2 = 1. \tag{9}$$

The robust estimator function is defined as the Cauchy loss function $\rho(\varepsilon) = \log(\beta^2 + \varepsilon^2)$, with the weight function $\phi(\varepsilon) = \frac{\beta^2}{\beta^2+\varepsilon^2}$, where ε denotes the residual for each observation and β is the loss width. Furthermore, when errors in normal vectors become significantly large for low parallax angles, estimating relative translations or verifying

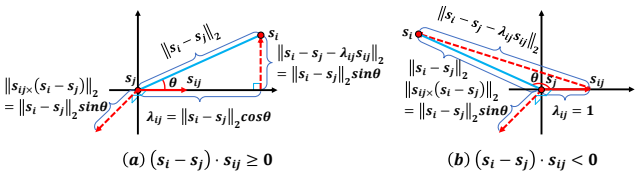

Figure 4. Residuals of two linear objection functions: $||\boldsymbol{s}_{ij\times}(\boldsymbol{s}_i - \boldsymbol{s}_j)||_2$ and $||\boldsymbol{s}_i - \boldsymbol{s}_j - \lambda_{ij}\boldsymbol{s}_{ij}||_2$ under different circumstances.

feature matches based on coplanarity consistency becomes invalid. Therefore, prior to estimating relative translations, we initially filter feature matches with parallax angles below a predefined threshold, denoted as A.

4.2. Feature Tracks Selection

With the re-estimated relative translations, we filter out feature matches that violate coplanarity or cheirality constraints [23] and construct feature tracks using the union-find algorithm [34]. As numerous feature tracks may contain a high ratio of feature ray outliers, only a selected subset of feature tracks is employed to enhance both efficiency and robustness. According to Eq. (8), the coplanarity consistency of feature matches with larger parallax angles is more reliable. Therefore, all feature tracks are sorted in descending order based on their maximum parallax angles. We then examine each feature track to determine whether it can establish a connection between images that lack sufficient coverage times. This process continues until the selected subset of tracks covers all cameras at least N times.

4.3. Definition of Objective Function

Both camera-to-camera and camera-to-point constraints can be represented as the formulation: $\boldsymbol{s}_i - \boldsymbol{s}_j = ||\boldsymbol{s}_i - \boldsymbol{s}_j||_2 \cdot \boldsymbol{s}_{ij}$, where $\boldsymbol{s}_i, \boldsymbol{s}_j$ represent cameras or points and $\boldsymbol{s}_{ij}$ represents a known normalized vector from $\boldsymbol{s}_j$ to $\boldsymbol{s}_i$, *e.g.* a feature ray or a relative translation. We compare two types of linear objective functions, including the cross-product-form $||\boldsymbol{s}_{ij\times}(\boldsymbol{s}_i - \boldsymbol{s}_j)||_2$ and the scale-form $||\boldsymbol{s}_i - \boldsymbol{s}_j - \lambda_{ij}\boldsymbol{s}_{ij}||_2$, where λ_{ij} is a scale variable. To remove scales and directions ambiguity, inequality constraints $\boldsymbol{s}_{ij} \cdot (\boldsymbol{s}_i - \boldsymbol{s}_j) \geq 1$ and $\lambda_{ij} \geq 1$ are respectively utilized for the cross-product-form and the scale-form objective function.

Let $\boldsymbol{s}_{ij}^G$ be the ground truth of $\boldsymbol{s}_{ij}$. For most cases when $\boldsymbol{s}_{ij} \cdot \boldsymbol{s}_{ij}^G \geq 0$, both inequality constraints define correct feasible regions. In the case of an optimal solution, as illustrated in Fig. 4 (a), the magnitudes of residuals in both objective functions are identical. Meanwhile, λ_{ij} in the scale-form equals $\boldsymbol{s}_{ij} \cdot (\boldsymbol{s}_i - \boldsymbol{s}_j)$, representing the magnitude of the projection of $\boldsymbol{s}_i - \boldsymbol{s}_j$ on $\boldsymbol{s}_{ij}$. Therefore, λ_{ij} is a redundant variable since it is entirely determined by the current $\boldsymbol{s}_i, \boldsymbol{s}_j$ and known $\boldsymbol{s}_{ij}$. Moreover, when scale variables exhibit a wide range of variation, such as in cases with disparate lengths of baselines or depths of feature rays, they often struggle to converge to the optimum. This significantly impacts the overall accuracy and efficiency of practical optimization.

When $\boldsymbol{s}_{ij}$ has a significant error resulting in $\boldsymbol{s}_{ij} \cdot \boldsymbol{s}_{ij}^G < 0$, λ_{ij} in the scale-form equals the low bound 1 to minimize penalization as shown in Fig. 4 (b). The inequality constraint for the cross-product-form offers an incorrect feasible region, leading to a biased solution. However, with our relative translation re-estimation, the accuracy of overall relative translations is improved. Moreover, the issue of significant direction errors is well-solved in 1DSfM [54] by multiple random 1-dimension projections, enabling the filtration of most relative translations with significant errors. Hence, the cross-product-form objective function is used in our method for better convergence. A detailed comparison between these two forms is conducted in the experiments.

4.4. Optimization Framework

To avoid redundant and incorrect constraints from feature rays, only the camera-to-camera inequality constraints are utilized to remove the inherent positional and scale ambiguity. The convex objective function is optimized under the L_1 norm for robustness, as demonstrated below:

$$\min_{\substack{\boldsymbol{t}_i, i\in V;\\ \boldsymbol{P}_k, k\in P}} \sum_{ij\in E_v} ||\boldsymbol{v}_{ij\times}(\boldsymbol{t}_i - \boldsymbol{t}_j)||_1 + \sum_{ki\in E_p} ||\boldsymbol{f}_{ki\times}(\boldsymbol{P}_k - \boldsymbol{t}_i)||_1,$$
$$\text{s.t.} \quad \sum_{i\in V} \boldsymbol{t}_i = 0, \quad \boldsymbol{v}_{ij} \cdot (\boldsymbol{t}_i - \boldsymbol{t}_j) \geq 1, \quad \forall ij \in E_v. \tag{10}$$

However, as mentioned in [59], the solution of Eq. (10) is biased for disparate scales of camera baselines and feature point depths and the non-convex angle-based objective function needs a good initialization. Therefore, an unbiased angle-based objective function is utilized to refine the solution of Eq. (10) with a robust IRLS scheme as below:

$$\min_{\substack{\boldsymbol{t}_i, \boldsymbol{P}_k\\ k\in P;\\ i\in V;}} \sum_{ij\in E_v} \rho(\boldsymbol{H}(\boldsymbol{v}_{ij}, \frac{\boldsymbol{t}_i - \boldsymbol{t}_j}{||\boldsymbol{t}_i - \boldsymbol{t}_j||_2})) + \sum_{ki\in E_p} \rho(\boldsymbol{H}(\boldsymbol{f}_{ij}, \frac{\boldsymbol{P}_k - \boldsymbol{t}_i}{||\boldsymbol{P}_k - \boldsymbol{t}_i||_2})),$$
$$\text{s.t.} \sum_{i\in V} \boldsymbol{t}_i = 0, \text{ where } \boldsymbol{H}(\boldsymbol{s}_{ij}, \hat{\boldsymbol{s}}_{ij}) = \begin{cases} ||\boldsymbol{s}_{ij\times}\hat{\boldsymbol{s}}_{ij}||_2, & \boldsymbol{s}_{ij}^T\hat{\boldsymbol{s}}_{ij} \geq 0; \\ 1, & \boldsymbol{s}_{ij}^T\hat{\boldsymbol{s}}_{ij} < 0. \end{cases} \tag{11}$$

The robust estimator function $\rho(\cdot)$ is the same as defined in Eq. (9). The entire framework of HETA is shown below:

Algorithm 1 Hybrid explicit translation averaging method

Input: Pairwise feature matches and global camera rotations.
Output: Camera positions $\boldsymbol{t}_i, \forall i \in V$; 3D points $\boldsymbol{P}_i, \forall i \in P$.
1: Filter out feature matches with low parallax angle;
2: Re-estimate relative translations with known rotations;(Sec. 4.1)
3: Remove image matches that do not align with relative translations;
4: Select feature tracks to build view track graph G; (Sec. 4.2)
5: Estimating camera translations and 3D points (Sec. 4.4)

5. Experiments

Our method is demonstrated through experiments on both sequential dataset KITTI [19] and unordered dataset 1DSfM [54]. For the 1DSfM dataset, view graphs and

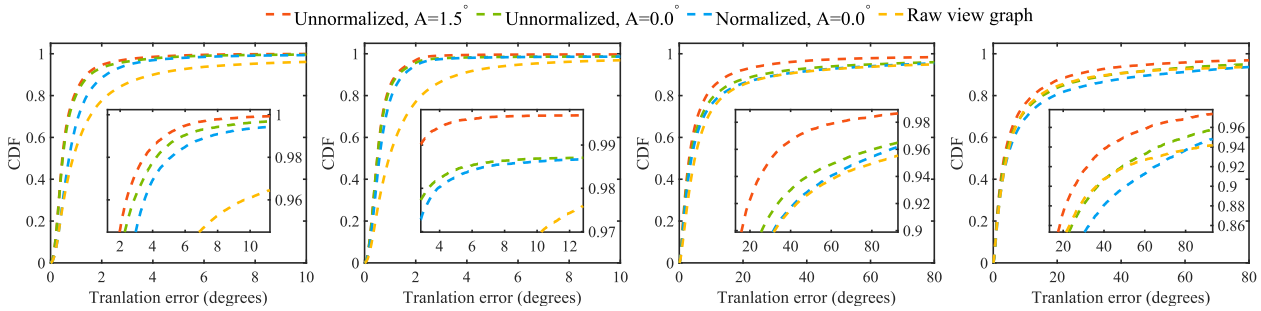

Figure 5. This figure shows the cumulative distribution functions of the relative translation angle errors for KITTI-06, KITTI-09, 1DSfM-PIC and 1DSfM-ROF (from left to right). The setting of 'Normalized, A=0.0° ' corresponds to the traditional method in [38].

Data	LUD[38]				CReTA-BATA [32]				LiGT[8]				PGILP[30]				1DSfM[54]				HETA					
	Init		BA		Init		BA		Init		BA		Init		BA		Init		BA		L_1		L_2		BA	
Name(N)	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$
00(9082)	29.8	49.3	26.9	51.7	16.9	35.9	14.4	35.5	85.6	2e2	85.0	2e2	7.4	14.6	7.0	14.1	1e2	1e2	1e2	1e2	3.0	7.7	2.6	7.6	**2.4**	**7.3**
01(2202)	93.8	**1e2**	76.0	2e2	70.1	2e2	55.4	9e2	55.4	9e2	41.5	1e3	53.8	**1e2**	35.6	**1e2**	4e2	2e3	4e2	7e2	**29.6**	**1e2**	32.9	**1e2**	36.1	**1e2**
02(9322)	22.5	26.5	20.9	26.1	18.2	23.5	16.9	22.4	2e2	2e2	1e2	2e2	30.1	43.1	29.0	43.1	2e2	2e2	2e2	2e2	4.5	**5.7**	**4.2**	6.3	4.6	6.5
03(1602)	9.5	14.3	5.8	21.9	7.5	17.4	5.7	15.5	23.1	78.9	22.1	75.0	5.8	30.3	5.7	28.9	1e2	1e2	1e2	1e2	0.2	0.4	0.2	0.4	**0.1**	**0.3**
04(542)	2.9	9.9	**0.1**	0.3	4.1	19.4	**0.1**	0.3	7.2	44.1	**0.1**	0.3	1.4	2.0	**0.1**	0.3	3.9	5e2	**0.1**	**0.2**	0.4	1.2	0.2	1.1	**0.1**	**0.2**
05(5522)	10.1	24.6	9.3	23.6	11.7	18.2	11.1	17.7	78.5	1e2	67.1	1e2	7.5	11.3	6.0	10.6	1e2	1e2	1e2	1e2	**1.6**	2.8	**1.6**	2.7	**1.6**	**2.5**
06(2202)	21.6	46.6	20.2	46.5	16.1	38.6	14.9	39.6	26.6	61.0	18.1	64.0	6.0	20.1	2.9	17.4	86.1	1e2	89.3	1e2	0.6	1.1	0.6	1.0	**0.2**	**0.4**
07(2202)	9.0	12.5	7.4	9.6	9.4	13.0	7.9	11.5	27.6	35.4	22.0	43.1	5.0	8.8	1.1	6.2	74.4	5e2	72.7	3e2	0.8	1.1	0.7	1.0	**0.5**	**0.9**
08(8142)	22.0	26.4	20.4	30.6	13.8	20.3	12.3	19.3	1e2	4e2	1e2	3e2	18.2	21.1	17.1	21.7	2e2	2e2	2e2	3e2	**5.6**	**6.9**	**5.6**	**6.9**	**5.6**	**6.9**
09(3182)	12.3	43.8	9.9	41.9	12.5	29.9	9.9	31.2	53.1	1e2	63.5	1e2	7.7	17.8	6.6	18.8	1e2	2e2	1e2	2e2	**1.7**	**3.7**	**1.7**	**3.7**	1.8	3.8
10(2402)	8.8	14.5	8.1	23.1	8.3	9.2	6.2	8.1	30.7	52.3	27.4	51.2	8.6	12.5	7.8	12.1	64.1	1e2	64.6	97.9	0.5	**4.2**	**0.4**	**4.2**	0.6	**4.2**

Table 1. Camera position accuracy before and after BA for different methods on the KITTI dataset. N represents the number of cameras in the view graph, and $\tilde{e}$ and $\bar{e}$ respectively denote the median and mean distance error in meters. The best results are shown in bold.

camera pose references were estimated by Bundler [48]. For a more accurate comparison, we employ the method COLMAP [44] to re-estimate view graphs and camera poses. To obtain the real scales for the reconstructions, we utilize the similarity transform [53] and RANSAC [11] algorithms to align the re-estimated camera poses with the results presented in [48] and consider the transformed camera poses as the new references. For the KITTI dataset, the ground truth camera poses are released in [19] and the view graphs are constructed by COLMAP, where similar image pairs are searched via the retrieval method NetVlad [2].

The method of Chatterjee [9] is used to estimate global camera rotations. To demonstrate the superiority of HETA, we conduct a comparison with several methods which include implicit methods like LiGT [8] and PGILP [30], a typical explicit method 1DSfM [54], as well as the methods that rely exclusively on relative translations, like LUD [38] and CReTA [32]. The methods LUD and 1DSfM are implemented by Theia library [49], while CReTA and LiGT are implemented by the authors respectively in MATLAB and OpenMVG library [36]. A revised PGILP is implemented by us, where implicit 3D points are represented based on Eq. (5). In HETA, the ADMM [6] method is employed to solve L_1 norm optimization, and the Ceres solver [1] is used for the BA. During the relative translation re-estimation, the threshold A used for filtering feature matches is set to $1.5°$. To ensure fairness, all methods use the same view graphs, feature matches, and global camera rotations as input.

5.1. Relative Translation Re-estimation

We conduct experiments to evaluate the impact of both using unnormalized normal vectors and filtering out feature matches with low parallax angles on the re-estimated relative translations. In Eq. (9), the loss width β is set to $\sin 1° \cdot \sin 5°$, indicating an expectation that the error angle γ should be less than $5°$ when the parallax angle α equals $1°$. The cumulative distribution functions (CDF) of the relative translation errors, as presented in Fig. 5, demonstrate that the use of unnormalized normal vectors significantly enhances the accuracy of relative translations. Further improvements are achieved by filtering feature matches with parallax angles below $A = 1.5°$. Additionally, the compromised accuracy of relative translations estimated using normalized normal vectors in the 1DSfM-ROF data is attributed to the sub-optimal global camera rotations. However, our method still yields a superior outcome in this case.

5.2. Evaluation on Sequential Data

The KITTI dataset is collected using two cameras mounted on a driving car, where most parallax angles of the feature matches are limited and the camera motion trajectories tend to be approximately collinear. Two cameras are used but considered independently in all experiments, which raises a significant challenge for the global translation estimation system. The calibration results are presented in Tab. 1, where HETA achieves the highest accuracy. The

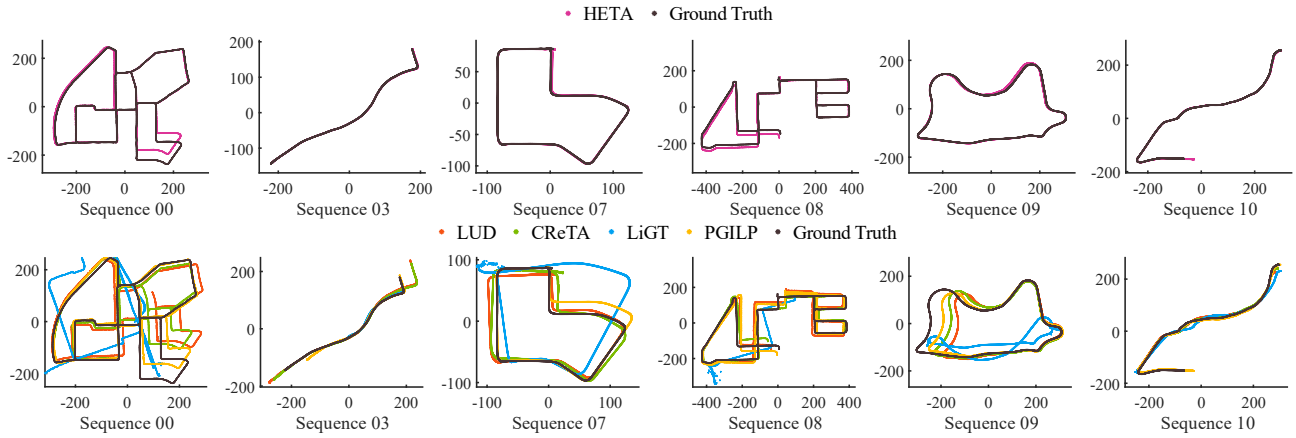

Figure 6. Comparison of camera motion trajectories on a part of KITTI[19] odometry benchmark. The sample state-of-the-art global SfM methods include LUD [38], CReTA [32], PGILP [30] and LiGT [8].

Data		LUD[38]			CReTA-BATA [32]			LiGT[8]			PGILP[30]			1DSfM[54]			HETA						
		BA			BA			BA			BA			BA			L_1		L_2		BA		
Name	N_t	$\tilde{e}$	$\bar{e}$	N_c	$\tilde{e}$	$\bar{e}$	N_c	$\tilde{e}$	$\bar{e}$	N_c	$\tilde{e}$	$\bar{e}$	N_c	$\tilde{e}$	$\bar{e}$	N_c	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	N_c
ALM	497	0.1	0.5	483	0.1	0.4	487	0.3	1.8	422	0.1	0.5	486	0.3	4.6	389	0.5	1.2	0.5	1.2	0.1	0.4	**488**
ELS	217	0.2	0.4	212	0.2	0.4	215	0.2	0.4	204	0.2	0.4	**216**	0.2	0.4	196	2.4	3.9	2.1	3.8	0.2	0.4	**216**
GDM	590	0.1	3.4	560	0.1	3.6	561	5.1	1e3	504	0.2	4.1	556	0.4	66.5	475	2.8	10.4	2.2	10.1	0.2	2.7	**564**
MDR	178	0.2	6.3	168	0.2	5.6	170	8.6	16.4	137	0.2	9.7	170	0.8	9.8	122	1.4	9.7	1.4	9.6	0.2	7.0	**174**
MND	403	0.1	0.1	399	0.1	0.1	399	0.1	0.1	383	0.1	0.1	398	0.1	0.3	363	0.5	1.0	0.5	1.0	0.1	0.1	**400**
ND	479	0.1	0.6	457	0.1	0.3	468	6.2	7.3	397	0.1	0.3	462	0.1	47.5	374	0.3	1.4	0.3	0.9	0.1	0.3	**476**
NYC	296	0.1	0.2	290	0.1	0.3	**294**	0.1	1.5	222	0.1	0.2	285	0.1	5.7	261	0.7	1.8	0.5	1.5	0.1	0.1	290
PDP	295	0.1	0.4	287	0.1	0.1	286	7.4	2e2	108	0.1	0.3	290	0.1	1.9	249	1.1	2.9	1.1	2.9	0.1	0.2	**291**
PIC	1838	0.1	0.5	1797	0.1	0.5	**1811**	12.3	81.8	649	0.1	0.5	1774	0.5	3.1	1621	0.9	1.9	0.7	1.7	0.1	0.4	1807
ROF	918	0.1	0.2	875	0.1	0.2	899	0.6	3.1	732	0.1	0.5	892	0.8	23.5	725	1.9	3.9	1.2	3.3	0.1	0.1	**907**
TFG	3989	0.9	2.5	3864	0.7	1.8	3913	37.5	45.8	789	1.2	4.3	3860	12.1	18.9	3348	3.3	6.4	2.6	5.8	0.7	2.4	**3951**
TOL	396	0.5	3.6	**391**	0.2	4.8	387	70.3	76.0	152	0.3	4.3	380	3.2	7e2	276	2.5	4.9	2.1	4.5	0.4	1.5	387
USQ	637	0.3	2.8	582	0.3	4.4	603	6.7	1e2	336	0.5	4.5	602	0.4	1e2	505	4.2	7.5	3.6	7.2	0.2	2.1	**619**
VNC	713	0.2	5.7	672	0.2	9.7	**702**	20.5	28.2	474	0.2	9.1	664	0.2	4.5	556	1.8	4.2	1.7	4.0	0.1	0.8	686
YKM	337	0.1	0.1	327	0.1	0.1	329	0.1	0.3	318	0.1	0.2	323	2.9	23.9	262	1.2	2.4	1.1	2.1	0.1	0.2	**333**

Table 2. Camera position accuracy on 1DSfM [54] dataset. N_t is the number of images in the view graph. N_c is the number of registered images after BA, whose best results are shown in bold. $\tilde{e}$ and $\bar{e}$ respectively denote the median and mean distance error in meters.

1DSfM method, which is also an explicit method, fails to reconstruct most of the data. Despite using highly accurate relative translations, as shown in Fig. 5, both LUD and CReTA-BATA struggle to produce accurate results. The method LiGT estimates the camera translations based on matrix decomposition, which enhances efficiency but compromises robustness. In contrast, PGILP produces better results by optimizing each camera-to-point constraint under the L_1 norm. The estimated camera motion trajectories are depicted in Fig. 6. From these comparisons, we can conclude that our method, HETA, surpasses all the compared methods in terms of both accuracy and robustness.

5.3. Evaluation on Unordered Data

The 1DSfM dataset [54] is collected by many different types of cameras. Due to the limited accuracy of provided camera intrinsic parameters and substantial incorrect feature matches, the estimated relative poses have large errors. As a consequence, the accuracy of global rotations estimated by [9] is not as high as that for the KITTI dataset, making global translation estimation more challenging. The calibration results after BA are shown in Tab. 2. From this comparison, the accuracy of estimated camera positions in LUD, CReTA-BATA and HETA is comparable. However, for the majority of the data, HETA registers the highest number of images, indicating its higher robustness compared to LUD and CReTA-BATA. For the implicit methods, LiGT and PGILP, which rely solely on all feature tracks as input, their performances are inferior to HETA. This discrepancy arises due to the incorporation of numerous incorrect constraints stemming from feature track outliers.

5.4. Ablation Study on Objective Function

We analyze the influence of various objective functions, considering different formulations ("Cross" denotes cross-product-form, and "Scale" denotes scale-form), various input observations ("PT" for pure relative translation, "PF" for pure feature tracks, and "H" for using a hybrid combination of both), and different methods for handling 3D points ("E" for explicit and "I" for implicit). For example, "Scale-PT" represents the method that employs the scale-form objective function solely based on relative translations

KITTI	Scale-PT		Cross-PT		Scale-PFI		Scale-HE		Cross-HI		Cross-HE	
Name	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$	$\tilde{e}$	$\bar{e}$
00	12.2	24.2	5.0	13.4	7.4	14.6	5.3	17.9	4.1	9.7	**3.0**	**7.7**
01	82.0	1e2	62.4	1e2	53.8	1e2	88.9	1e2	45.5	1e2	**29.6**	1e2
02	11.1	19.0	8.6	13.0	30.1	43.1	22.9	33.9	**3.6**	7.2	4.5	**5.7**
03	5.7	17.1	2.0	5.2	5.8	30.3	1.5	8.8	1.3	2.1	**0.2**	**0.4**
04	1.5	2.8	1.5	2.0	1.4	2.0	0.5	1.8	1.2	1.9	**0.4**	**1.2**
05	7.5	11.5	3.2	4.7	7.5	11.3	2.9	4.2	**1.4**	3.2	1.6	**2.8**
06	7.4	15.5	4.5	13.3	6.0	20.1	3.3	17.6	1.8	10.6	**0.6**	**1.1**
07	4.7	8.1	1.2	2.4	5.0	8.8	3.3	7.1	1.0	1.4	**0.8**	**1.1**
08	15.0	23.7	**3.8**	7.3	18.2	21.1	26.5	38.4	5.6	8.2	5.6	**6.9**
09	14.4	26.0	7.0	24.4	7.7	17.8	3.7	13.8	1.9	14.8	**1.7**	**3.7**
10	5.4	12.6	0.7	3.6	11.8	24.9	4.5	15.6	0.7	**3.2**	**0.5**	4.2

Table 3. Camera position errors produced by applying various objective functions on the KITTI dataset. The best results are shown in bold and the second-best results are underlined.

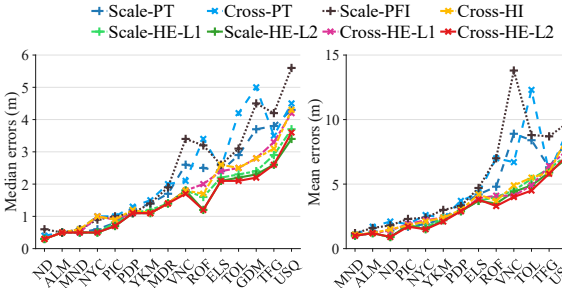

Figure 7. Median and mean camera position errors produced by applying various objective functions on the 1DSfM dataset.

as input, corresponding to the method LUD [38]. Similarly, "Scale-PFI" and "Cross-HE" respectively correspond to the methods PGILP [30] and HETA. All hybrid methods solely use camera-to-camera inequality constraints for L_1 norm optimization. The calibration results for KITTI and 1DSfM datasets are respectively displayed in Tab. 3 and Fig. 7.

When comparing different formulations, cross-product-form methods outperform scale-form methods for better convergence on the KITTI dataset, where the scales of both types of input observations exhibit wide variations. For some data in the 1DSfM dataset, numerous relative translation outliers display substantial angular errors. As a result, the performance of the cross-product-form method under L_1 norm optimization is slightly lower than that of the scale-form. However, after unbiased L_2 norm optimization, the results from both formulations become comparable. In terms of input observations, methods with hybrid input generally outperform those with pure relative translations or pure feature tracks. Pure relative translations lack constraints from 3D points, and pure feature tracks lack robustness. In contrast, methods with hybrid input strike a balance between these two aspects. When considering the handling manner of 3D points, explicit methods outperform implicit methods on both the KITTI and 1DSfM datasets.

The running time of three hybrid methods with different objective functions is illustrated in Fig. 8. We find cross-product-form methods are more efficient, as they avoid redundant scale variable optimization. Furthermore, the efficiency of explicit methods and implicit methods is comparable, dispelling the misconception that explicit methods escalate problem complexity. Complete time comparison with existing methods is shown in supplemental material.

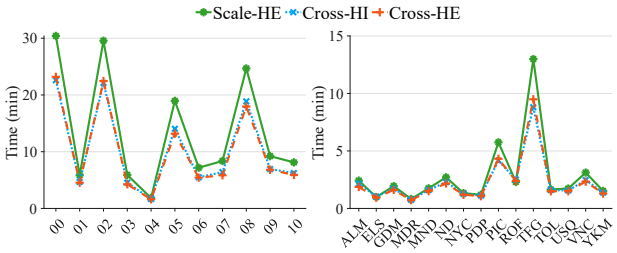

Figure 8. Running time for L_1 norm optimization of three hybrid methods on KITTI dataset (left) and 1DSfM dataset (right).

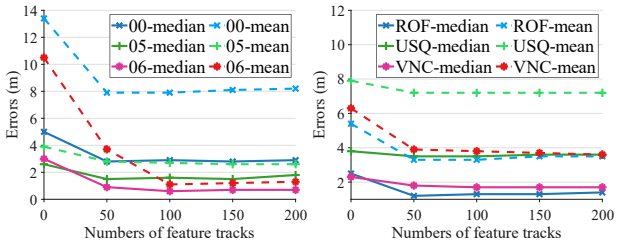

Figure 9. Camera position errors of the HETA method with varying numbers of feature tracks on some data for the KITTI dataset (left) and some data for the 1DSfM dataset (right).

5.5. Discussion on Track Selection

In this section, we explore how the quantity of feature tracks impacts the accuracy of HETA. As depicted in Fig. 9, with an increase in feature tracks, initial reductions in camera position errors are observed, followed by stabilization, although some instances show a slight rise. This indicates that incorporating feature tracks effectively enhances accuracy but excessive tracks may introduce outliers, diminishing accuracy. Given the heightened issue of collinear motion in sequential data, we utilize a larger number of feature tracks. Specifically, parameter N is set to 50 for unordered datasets and 100 for sequential datasets in our study.

6. Conclusion

We revisit the global translation estimation problem with feature tracks and propose a novel hybrid explicit framework. Our approach outperforms many existing state-of-the-art methods on both sequential and unordered datasets. However, the prevalence of feature match outliers still poses a challenge to the broader adoption of global SfM. In the future, we intend to harness the insights gained from neural networks to enhance the performance of feature matching.

Acknowledgments This work was supported by the National Key R&D Program of China (No.2023YFB3906600), the National Natural Science Foundation of China (No.U22B2055, U23A20386, 62273345 and 62073320), and the Beijing Natural Science Foundation (No.L223003).

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver. https://github.com/ceres-solver/ceres-solver, 2023. 6

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 6

[3] Federica Arrigoni, Andrea Fusiello, Elisa Ricci, and Tomas Pajdla. Viewing graph solvability via cycle consistency. In *IEEE International Conference on Computer Vision*, pages 5540–5549, 2021. 1

[4] Federica Arrigoni, Tomas Pajdla, and Andrea Fusiello. Viewing graph solvability in practice. In *IEEE/CVF International Conference on Computer Vision*, pages 8147–8155, 2023. 1

[5] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 14546–14555, 2021. 1

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3 (1):1–122, 2011. 6

[7] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *IEEE International Conference on Computer Vision*, pages 6218–6228, 2021. 1

[8] Qi Cai, Lilian Zhang, Yuanxin Wu, Wenxian Yu, and Dewen Hu. A pose-only solution to visual reconstruction and navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):73–86, 2021. 1, 2, 3, 4, 6, 7

[9] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision*, pages 521–528, 2013. 1, 2, 6, 7

[10] Yu Chen, Ji Zhao, and Laurent Kneip. Hybrid rotation averaging: A fast and robust rotation averaging approach. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10358–10367, 2021. 1, 2

[11] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*, pages 236–243. Springer, 2003. 6

[12] Alejo Concha, Michael Burri, Jesús Briales, Christian Forster, and Luc Oth. Instant visual odometry initialization for mobile ar. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4226–4235, 2021. 1

[13] David J Crandall, Andrew Owens, Noah Snavely, and Daniel P Huttenlocher. Sfm with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2841–2853, 2012. 2

[14] Hainan Cui, Shuhan Shen, and Zhanyi Hu. Robust global translation averaging with feature tracks. In *IEEE International Conference on Pattern Recognition*, pages 3727–3732, 2016. 2, 3

[15] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *IEEE International Conference on Computer Vision*, pages 864–872, 2015. 2, 3

[16] Zhaopeng Cui, Nianjuan Jiang, Chengzhou Tang, and Ping Tan. Linear global translation estimation with feature tracks. In *British Machine Vision Conference*, pages 46.1–46.13, 2015. 2, 3

[17] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the p3p problem. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4872–4880, 2023. 1

[18] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 1

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5, 6, 7

[20] 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. 2

[21] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II–II, 2001. 2

[22] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 1, 2

[23] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[24] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. 4

[25] Aleksander Holynski, David Geraghty, Jan-Michael Frahm, Chris Sweeney, and Richard Szeliski. Reducing drift in structure from motion using extended features. In *IEEE International Conference on 3D Vision (3DV)*, pages 51–60, 2020. 1

[26] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *IEEE International Conference on Computer Vision*, pages 481–488, 2013. 2, 3

[27] Fredrik Kahl and Richard Hartley. Multiple-View Geometry Under the L_∞-Norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008. 2

[28] Qifa Ke and Takeo Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007. 2

[29] Seong Hun Lee and Javier Civera. Hara: A hierarchical approach for robust rotation averaging. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15777–15786, 2022. 1, 2

[30] Liyang Liu, Teng Zhang, Brenton Leighton, Liang Zhao, Shoudong Huang, and Gamini Dissanayake. Robust global structure from motion pipeline with parallax on manifold bundle adjustment and initialization. *IEEE Robotics and Automation Letters*, 4(2):2164–2171, 2019. 1, 2, 3, 4, 6, 7, 8

[31] Weiquan Liu, Cheng Wang, Yu Zang, Shang-Hong Lai, Dongdong Weng, Xuesheng Bian, Xiuhong Lin, Xuelun Shen, and Jonathan Li. Ground camera images and uav 3d model registration for outdoor augmented reality. In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1050–1051, 2019. 1

[32] Lalit Manam and Venu Madhav Govindu. Correspondence reweighted translation averaging. In *European Conference on Computer Vision*, pages 56–72. Springer, 2022. 1, 2, 4, 6, 7

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[34] Pierre Moulon and Pascal Monasse. Unordered feature tracking made fast and easy. In *European Conference on Visual Media Production*, page 1, 2012. 5

[35] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision*, pages 3248–3255, 2013. 2

[36] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *Reproducible Research in Pattern Recognition: First International Workshop*, pages 60–74. Springer, 2017. 6

[37] Carl Olsson, Anders Eriksson, and Richard Hartley. Outlier removal using duality. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1450–1457. IEEE, 2010. 1

[38] Onur Ozyesil and Amit Singer. Robust camera location estimation by convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 1, 2, 4, 6, 7, 8

[39] Zhe Peng, Songlin Hou, and Yixuan Yuan. Epar: An efficient and privacy-aware augmented reality framework for indoor location-based services. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8948–8955, 2022. 1

[40] Rodrigo Chacón Quesada and Yiannis Demiris. Design and evaluation of an augmented reality head-mounted display user interface for controlling legged manipulators. In *IEEE International Conference on Robotics and Automation*, pages 11950–11956, 2023. 1

[41] Jie Ren, Wenteng Liang, Ran Yan, Luo Mai, Shiwen Liu, and Xiao Liu. Megba: A gpu-based distributed library for large-scale bundle adjustment. In *European Conference on Computer Vision*, pages 715–731. Springer, 2022. 1

[42] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 1

[43] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3247–3257, 2021. 1

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 6

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

[46] Kristy Sim and Richard Hartley. Recovering camera motion using l\infty minimization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1230–1237. IEEE, 2006. 2

[47] Kristy Sim and Richard Hartley. Removing outliers using the l\infty norm. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 485–494. IEEE, 2006. 1

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

[49] Chris Sweeney. Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org, 2019. 1, 6

[50] Chris Sweeney, Torsten Sattler, Tobias Hollerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *IEEE International Conference on Computer Vision*, pages 801–809, 2015. 1

[51] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 1

[52] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*, pages 298–372. Springer, 2000. 1

[53] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 6

[54] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014. 2, 4, 5, 6, 7

[55] Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8296–8306, 2023. 1

[56] Luwei Yang, Heng Li, Jamal Ahmed Rahim, Zhaopeng Cui, and Ping Tan. End-to-end rotation averaging with multi-source propagation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11774–11783, 2021. 2

[57] Ganlin Zhang, Viktor Larsson, and Daniel Barath. Revisiting rotation averaging: Uncertainties and robust losses. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 17215–17224, 2023. 2

[58] 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 Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018. 1, 2

[59] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2018. 2, 4, 5