2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

17-Point Algorithm Revisited: Toward a More Accurate Way

Chen Xie, Rui Xing, Ning Hao, Fenghua He*

***Abstract*— 17-point algorithm is a popular method in relative pose estimation of multi-cameras. However, the role of overlap in 17-point algorithm remains unexplored. And the relaxed way in solving constrained normal equation leads to sub-optimal results. Both of them influence accuracy of the estimated pose. In this paper, we theoretically analyze the influence of overlap and the solvability of 17-point algorithm. In addition, we show that the abuse of overlap can harm accuracy in practice. In light of these findings, we propose an improved 17-point algorithm, which avoids using overlaps and derives a simple way to solve normal equation on manifold. Both simulations and real world data experiments demonstrate the proposed one outperforms the traditional 17-point algorithm in term of accuracy.**

I. Introduction

Relative pose estimation between two frames is of great significance in computer vision. The problem can be found in various areas, such as structure-from-motion [1], 3D reconstruction [2], SLAM [3, 4]. Moreover, as more robots are being equipped with multi-cameras, the estimation of relative pose from a multi-camera rig has gained increasing popularity in recent years [5]. The multi-camera field is usually used *Generalized Camera Model* (GCM), where rays representing observations [6]. The milestone work [6, 7] derived *generalized epipolar constraint* (GEC) for two-view geometry of GCM. A linear method, called *17-point algorithm*, was proposed to solve the GEC estimation problem. By 17-point algorithm, one can get translation without scale ambiguity, which is different to what in central camera [8].

In practice, it usually has overlapping field of view between cameras. Take Fig. 1 for an example, where the gray area denotes for multi-camera rig (or general camera), C_i or C_i' $(i = 1, 2, 3)$ denotes for camera in this rig. The overlapping field causes the 3D point (represented by P) can be observed by $\{C_1, C_2, C_3\}$ in the first rig, and $\{C_1', C_2'\}$ in the second rig. One can use both image correspondences $(\mathrm{C}_1, \mathrm{P}, \mathrm{C}_1')$ and $(\mathrm{C}_2, \mathrm{P}, \mathrm{C}_1')$ in 17-point algorithm. We call $(\mathrm{C}_2, \mathrm{P}, \mathrm{C}_1')$ an overlap, if we decide to use $(\mathrm{C}_1, \mathrm{P}, \mathrm{C}_1')$. Formally speaking, an *overlap* is the extra image correspondence of a 3D point. Interestingly, none of related works of 17-point algorithm [6, 7, 9–11] explicitly considered overlap. And their experiments consistently adopted omni-directional camera which could naturally prevent overlap. In this article, we theoretically discuss the role of overlap in 17-point algorithm. We mainly focus on following questions: (1) whether overlap is redundant in the meaning of providing constraints?

Chen Xie, Rui Xing, Ning Hao are Ph.D students with Harbin Institute of Technology, Harbin, China. (email:20B904012@stu.hit.edu.cn, 20B904062@stu.hit.edu.cn, 18B904055@stu.hit.edu.cn)
Fenghua He is faculty with Harbin Institute of Technology, Harbin, China. (email: hefenghua@hit.edu.cn)

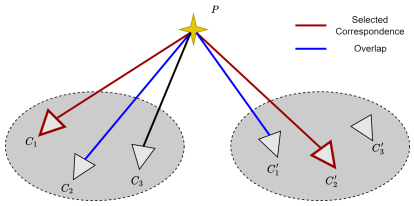

Fig. 1: An example to illustrate the concept of overlap, which denotes for the extra correspondence.

(2) when overlap is allowed, how about the solvability of 17-point algorithm? (3) in practice, is it beneficial to consider overlaps? We answer the above questions in Sec. II, our findings can explain two statements in existing literature but left unexplored: (1) in real world experiment, [9] found that two-camera configuration leads to the degeneracy of 17-point algorithm. (2) many works [10, 12] recommend to use omni-directional camera based on experience.

After revisiting 17-point algorithm with overlap-allowed and giving some theoretical analyses, we propose an improved version of 17-point algorithm in Sec. III. On the one hand, it meets the criterion pointed out by theoretical analysis in Sec. II. On the other hand, it handles the non-relaxed GEC estimation problem, contrast with traditional 17-point algorithm which solves a relaxed one. Moreover, different to existing works [9, 12–14] which solving non-relaxed GEC estimation with complex nonlinear optimization, we derive a simple local optimization way to get solutions on manifold.

To summarize, we make the following contributions:

- We revisit the 17-point algorithm with overlap-allowed. For the first time, we give the necessary and sufficient condition of successfully applying 17-point algorithm for a multi-camera system.
- In light of these analysis to 17-point algorithm, we propose an enhanced version of 17-point algorithm, which can improve accuracy of relative pose estimation.
- Simulations and real data experiments demonstrate that the proposed method outperforms traditional 17-point algorithm in term of accuracy.

II. 17-Point Algorithm with Overlap-Allowed

A. Review to 17-Point Algorithm

In GCM, observations are expressed by Plücker vectors [15]. It is used to represent straight line in a coordinate system O_m which can be chosen arbitrarily. A Plücker vector has six components, usually denoted as $\boldsymbol{l} = [\boldsymbol{q}^{\mathrm{T}}, \boldsymbol{q}'^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$, where $\boldsymbol{q}, \boldsymbol{q}' \in \mathbb{R}^3$. The first three components $\boldsymbol{q}$ is called *direction vector*, it is an unit view ray vector in frame coordinate. The last three components $\boldsymbol{q}'$ is called *moment*

vector. It has the form $\boldsymbol{q}' = \boldsymbol{p} \times \boldsymbol{q}$, where $\boldsymbol{p}$ can be an arbitrary point in the view ray, often taken as the translation part of camera's extrinsic.

Plücker correspondence $\boldsymbol{l_1} = [\boldsymbol{q_1}^{\mathrm{T}}, \boldsymbol{q_1'}^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{l_2} = [\boldsymbol{q_2}^{\mathrm{T}}, \boldsymbol{q_2'}^{\mathrm{T}}]^{\mathrm{T}}$ subjects to GEC [6] is given by

$$\boldsymbol{l_2}^{\mathrm{T}} \cdot \begin{bmatrix} \boldsymbol{E} & \boldsymbol{R} \\ \boldsymbol{R} & \boldsymbol{0} \end{bmatrix} \cdot \boldsymbol{l_1} = 0, \tag{1}$$

where $\boldsymbol{E} = \boldsymbol{t}^{\wedge}\boldsymbol{R}$, $(\cdot)^{\wedge}$ denotes the corresponding skew-symmetric matrix of a 3-dimensional vector [16], rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ transform a vector from O_1 to O_2, where $O_m (m = 1, 2)$ is the origin of general camera. Moreover, (1) can be equivalently transformed to the following form

$$\begin{bmatrix} (\boldsymbol{q_2} \otimes \boldsymbol{q_1})^{\mathrm{T}} \mid (\boldsymbol{q_2} \otimes \boldsymbol{q_1'} + \boldsymbol{q_2'} \otimes \boldsymbol{q_1})^{\mathrm{T}} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{vec}(\boldsymbol{E}) \\ \mathrm{vec}(\boldsymbol{R}) \end{bmatrix} = 0, \tag{2}$$

where $\mathrm{vec}(\cdot)$ denotes vectorization of a matrix. It is important to claim that for programming convenience, the vec operator in our paper denotes vectorization **by row-priority**, **not by column-priority** as usual definition. Due to the symmetry of frame index, this definition does not affect any conclusions.

Stacking all constraints with the form of (2) leads to a linear equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \tag{3}$$

which is also called *normal equation*. In 17-point algorithm, one ignores the constrains in $\boldsymbol{E}$ and $\boldsymbol{R}$ to get a relaxed problem. That is, we use $\boldsymbol{x} \in \mathbb{R}^{18}$ rather than $\boldsymbol{x} = [\mathrm{vec}(\boldsymbol{E})^{\mathrm{T}}, \mathrm{vec}(\boldsymbol{R})^{\mathrm{T}}]^{\mathrm{T}}$. There are 17 degrees of freedom in relaxed $\boldsymbol{x}$, since GEC reduces one freedom. Usually one needs to use at least 17 correspondences to determine $\boldsymbol{E}$ and $\boldsymbol{R}$, where $\boldsymbol{E} = \boldsymbol{t}^{\wedge}\boldsymbol{R}, \boldsymbol{R} \in \mathrm{SO}(3), \boldsymbol{t} \in \mathbb{R}^3$. Followed by recovering $\boldsymbol{R}$ and $\boldsymbol{t}$ based on [6, 7].

B. Solvability of 17-Point Algorithm with Overlap-allowed

In the overlap-allowed scenario, as shown in Fig. 1, any correspondence can be taken as a constraint. When we do not consider noise and there is only one 3D point, if overlap is completely redundant, $\mathrm{rank}(\boldsymbol{A})$ will always be 1. However, in Thm. 1, we prove that $\mathrm{rank}(\boldsymbol{A})$ can be greater than 1, which indicates overlap can provide extra constraints.

Theorem 1: Suppose one point P is observed by two general cameras O_1 and O_2. Denote $\{\boldsymbol{l_i} \mid \boldsymbol{l_i} = [\boldsymbol{q_i}^{\mathrm{T}}, \boldsymbol{q_i'}^{\mathrm{T}}]^{\mathrm{T}}, i = 1, \cdots, M\}$ and $\{\boldsymbol{l_j} \mid \boldsymbol{l_j} = [\boldsymbol{q_j}^{\mathrm{T}}, \boldsymbol{q_j'}^{\mathrm{T}}]^{\mathrm{T}}, j = 1, \cdots, N\}$ are Plücker line observations in each general camera, respectively. And $\boldsymbol{A} \triangleq \{[\boldsymbol{z}^{\mathrm{T}}]_{L\times 18} \mid \boldsymbol{z} = [\boldsymbol{q_j} \otimes \boldsymbol{q_i}; \boldsymbol{q_j} \otimes \boldsymbol{q_i'} + \boldsymbol{q_j'} \otimes \boldsymbol{q_i}], \boldsymbol{A} \cdot [\mathrm{vec}(\boldsymbol{E}); \mathrm{vec}(\boldsymbol{R})] = \boldsymbol{0}\}$. Then $\mathrm{rank}(\boldsymbol{A}) \leq 9$. Moreover, if $\min\{M, N\} = 1$, $\mathrm{rank}(\boldsymbol{A}) \leq 3$.

Proof: See Appendix A. ∎

As we know from Thm. 1 that overlap is not completely redundant, we should include overlap when analyze the solvability of 17-point algorithm. With overlap-allowed, we have Thm. 2 as follows, where K can be arbitrarily large.

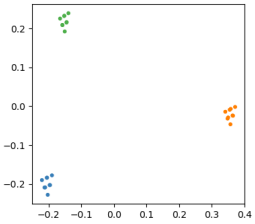

Fig. 2: An example to illustrate the gathering characteristic of $[\boldsymbol{l}_i, \boldsymbol{l}_j]$ when observing 3D points with large depth-baseline ratio.

Theorem 2: Suppose a series of K points $\{P_k \in \mathbb{R}^3 \mid k = 1, \cdots, K\}$ are observed by two general cameras O_1 and O_2, while $O_m (m = 1, 2)$ have more than one cameras $C_n (n = 1, 2, \cdots)$. Denote $\{\boldsymbol{l_{ik}} \mid \boldsymbol{l_{ik}} = [\boldsymbol{q_{ik}}^{\mathrm{T}}, \boldsymbol{q_{ik}'}^{\mathrm{T}}]^{\mathrm{T}}, i = 1, \cdots, M, k = 1, \cdots, K\}$ and $\{\boldsymbol{l_{jk}} \mid \boldsymbol{l_{jk}} = [\boldsymbol{q_{jk}}^{\mathrm{T}}, \boldsymbol{q_{jk}'}^{\mathrm{T}}]^{\mathrm{T}}, j = 1, \cdots, N, k = 1, \cdots, K\}$ are Plücker line observations in each general camera, respectively. And $\boldsymbol{A} \triangleq \{[\boldsymbol{z}^{\mathrm{T}}]_{L\times 18} \mid \boldsymbol{z} = [\boldsymbol{q_{jk}} \otimes \boldsymbol{q_{ik}}; \boldsymbol{q_{jk}} \otimes \boldsymbol{q_{ik}'} + \boldsymbol{q_{jk}'} \otimes \boldsymbol{q_{ik}}], \boldsymbol{A} \cdot [\mathrm{vec}(\boldsymbol{E}); \mathrm{vec}(\boldsymbol{R})] = \boldsymbol{0}\}$.

(1) If $M = N = 1$, $\mathrm{rank}(\boldsymbol{A}) \leq 8$;
(2) If $M = N = 2$, $\mathrm{rank}(\boldsymbol{A}) \leq 16$;
(3) If $M = N \geq 3$, $\mathrm{rank}(\boldsymbol{A}) \leq 17$.

Proof: See Appendix B. ∎

Note that we let $M = N$ because we only care about relative pose estimation of a multi-camera rig frame-by-frame. From Thm. 2, we can easily get the necessary and sufficient condition for applying 17-point algorithm:

Corollary 1: Let N_{cam} be the number of cameras in the multi-camera rig. Then, the necessary and sufficient condition for successfully applying 17-point algorithm is $N_{cam} \geq 3$, which means at least three cameras are needed.

C. Limitation of Overlap in Practice

Actually, overlaps are only useful in analyzing the solvability of 17-point algorithm. Although overlap can provide extra constraints, we find it can harm accuracy of 17-point algorithm in practice. Imagine three very far 3D points are observed by two multi-camera rigs, each has three cameras. It looks like there are more than 17 constraints. However, if we reduce dimension of $[\boldsymbol{l}_i, \boldsymbol{l}_j] \in \mathbb{R}^{12}$ to 2 based on PCA, and depict them on 2D plane, see Fig. 2. We will observe that matches from same 3D point tend to gather together because of **the large depth-baseline ratio, which is usually satisfied in practice.** Clustered matches are just like noised matches of the same 3D point, so there are only 3 useful constraints in Fig. 2. Algebraically, the second smallest singular value of $\boldsymbol{A}$ in (3) will be very small. Therefore, it is not proper to select the unit right-singular vector of the smallest singular value as solution when solving least square problem (3).

III. Improved 17-Point Algorithm

A. Overlap-avoided Track Build

If one 3D point is observed by multiple sub-cameras in a multi-camera system, like what in Fig. 1, the GECs formed by these overlapping observations, such as the GECs

formed by $(\mathrm{C}_1, \mathrm{C}_1')$ and $(\mathrm{C}_1, \mathrm{C}_2')$, may confuse each other in practice. Simply taken 17 these kind of constraints in (3) can harm accuracy.

We propose a way to avoid using overlap. To start with, we introduce the concept of track. Denote a *track* as all 2D correspondences of a 3D point. For example, all 2D correspondences (dashed lines) in Fig. 3 form a track. We build tracks at first, which relates to feature matching. After that, in order to avoid using overlap, we select one cross-frame correspondence from each track, which means we randomly select one camera from $(\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3)$, and randomly select one camera from $(\mathrm{C}_1', \mathrm{C}_2')$ to form a cross-frame correspondence (see red one). As a result, overlap is avoided, which can potentially increase accuracy of 17-point algorithm. The subsequent simulation (Fig. 4) will show that both relative rotation and translation error get minimum when the number of used overlaps per track is zero.

B. Local Optimization of GEC Estimation

From the above, we know not using overlap can potentially lead to better accuracy. In the following, we propose another technique that can improve accuracy. Considering the 17-point algorithm needs to solve a relaxed linear equation (3). However, we want to solve the original GEC estimation problem. That is, $\boldsymbol{x}$ is considered on manifold in (3). Formally, we define the original GEC problem as follows

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \quad \boldsymbol{x} = \begin{bmatrix} \mathrm{vec}(\boldsymbol{E}) \\ \mathrm{vec}(\boldsymbol{R}) \end{bmatrix}, \tag{4}$$

where $\boldsymbol{E} = \boldsymbol{t}^\wedge \boldsymbol{R}$, $\boldsymbol{R} \in \mathrm{SO}(3)$, $\boldsymbol{t} \in \mathbb{R}^3$. Denote $X = \{\boldsymbol{x} \in \mathbb{R}^{18} \mid \boldsymbol{x} = [\mathrm{vec}^{\mathrm{T}}(\boldsymbol{E}), \mathrm{vec}^{\mathrm{T}}(\boldsymbol{R})]^{\mathrm{T}}\}$. The underlying least square problem on manifold is

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in X} \|\boldsymbol{A}\boldsymbol{x}\|_2. \tag{5}$$

Observing that vanilla 17-point algorithm has been able to give reasonable results in most cases, an efficient way is performing local optimization directly on the manifold. This motivates us to disturb rotation and translation, solving (5) on tagent plane. Denote

$$\boldsymbol{R}_{n+1} = \exp(\Delta\boldsymbol{\Phi}_n^\wedge)\boldsymbol{R}_n, \tag{6}$$
$$\boldsymbol{t}_{n+1} = \boldsymbol{t}_n + \Delta\boldsymbol{t}_n, \tag{7}$$

where $n = 0, 1, 2, \cdots$, $\Delta\boldsymbol{\Phi}_n, \Delta\boldsymbol{t}_n \in \mathbb{R}^3$, and $\boldsymbol{R}_0$, $\boldsymbol{t}_0$ can be obtained from vanilla 17-point algorithm. The general epipolar constraints becomes

$$\begin{aligned} &\boldsymbol{q}_j \left(\boldsymbol{t}_n + \Delta\boldsymbol{t}_n\right)^\wedge \exp(\Delta\boldsymbol{\Phi}_n^\wedge)\boldsymbol{R}_n\boldsymbol{q}_i \\ &+ \boldsymbol{q}_j^{\mathrm{T}} \exp(\Delta\boldsymbol{\Phi}_n^\wedge)\boldsymbol{R}_n\boldsymbol{q}_i' + \boldsymbol{q}_j'^{\mathrm{T}} \exp(\Delta\boldsymbol{\Phi}_n^\wedge)\boldsymbol{R}_n\boldsymbol{q}_i = 0. \end{aligned} \tag{8}$$

Here we ignore the track index for simplicity. We use the first-order approximation $\exp(\Delta\boldsymbol{\Phi}_n^\wedge) \approx \boldsymbol{I} + \Delta\boldsymbol{\Phi}_n^\wedge$, and ignore the second-order quantities $\Delta\boldsymbol{t}_n^\wedge \Delta\boldsymbol{\Phi}_n^\wedge \boldsymbol{R}_n\boldsymbol{q}_i \approx 0$. One can get an equation linear on $(\Delta\boldsymbol{t}_n, \Delta\boldsymbol{\Phi}_n)$ as follows:

$$\begin{bmatrix} \left[\boldsymbol{b}_{ij}^1\right]^{\mathrm{T}} & \left[\boldsymbol{b}_{ij}^2\right]^{\mathrm{T}} & e_{ij}(\boldsymbol{x}_n) \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{t}_n \\ \Delta\boldsymbol{\Phi}_n \\ 1 \end{bmatrix} = 0, \tag{9}$$

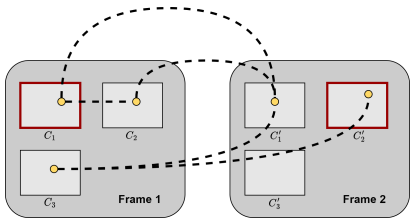

Fig. 3: 2D track, consist of intra-correspondences and cross-correspondences (dashed line), only select one cross-camera pair from it to prevent using overlap (red).

where $e_{ij}(\boldsymbol{x}_n)$ is the algebra error of $(\boldsymbol{l}_i, \boldsymbol{l}_j)$ using $\boldsymbol{x}_n$, and

$$\boldsymbol{b}_{ij}^1 = (\boldsymbol{R}_n\boldsymbol{q}_i)^\wedge \boldsymbol{q}_j, \tag{10}$$
$$\boldsymbol{b}_{ij}^2 = (\boldsymbol{R}_n\boldsymbol{q}_i')^\wedge \boldsymbol{q}_j + (\boldsymbol{R}_n\boldsymbol{q}_i)^\wedge \boldsymbol{q}_j' - (\boldsymbol{R}_n\boldsymbol{q}_i)^\wedge \boldsymbol{t}_n^\wedge \boldsymbol{q}_j. \tag{11}$$

Considering all observations, we have

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{x}_n \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{t}_n \\ \Delta\boldsymbol{\Phi}_n \\ 1 \end{bmatrix} = 0, \tag{12}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \left[\boldsymbol{b}_{ij}^1\right]^{\mathrm{T}} & \left[\boldsymbol{b}_{ij}^2\right]^{\mathrm{T}} \\ \vdots & \vdots \end{bmatrix}_{L \times 6}, \tag{13}$$

with L being the number of all correspondences.

Eq. (12) can be solved linearly by SVD method easily. Once get the residual, we can update state variables by (6) and (7), and repeat several times till convergence. We set the number of iterations to 3 to balance the accuracy and efficiency.

C. The Overall Algorithm

The overall improved 17-point algorithm is shown in Alg. 1, with both overlap-avoided track build and local optimization-based solver. Here RANSAC framework [17] is adopted. We perform all RANSAC iterations and select the result with minimum algebra error. We set the RANSAC iteration times $\mathrm{N}_{\mathrm{iter}} = 10$ in all experiment.

Algorithm 1 Improved 17-Point Algorithm

Require: 2D correspondences $\left\{(\boldsymbol{l}_i, \bar{\boldsymbol{l}}_j)\right\}_L$
Assert($N_{cam} \geq 3$);
Build 2D tracks P for all correspondences;
for $i \leftarrow 1$ to N_{iter} **do**
  for $j \leftarrow 1$ to N_{pt} **do**
    Randomly select one track from P (non-repeat);
    Randomly select one match from this track;
  Construct matrix $\boldsymbol{A}$ in (4);
  Solve relaxed problem (3), get $\boldsymbol{x}_0$;
  for $n \leftarrow 1$ to 3 **do**
    Construct matrix $\boldsymbol{B}$ using $\boldsymbol{x}_{n-1}$ by (10) (11);
    Solve least square problem (12), get $\Delta\boldsymbol{x}_{n-1}$;
    Update: $\boldsymbol{x}_n = \Delta\boldsymbol{x}_{n-1} \boxplus \boldsymbol{x}_{n-1}$;
  Recover $\boldsymbol{t}_i$ and $\boldsymbol{R}_i$ from $\boldsymbol{x}_3$;
Choose the best $\boldsymbol{t}^*$ and $\boldsymbol{R}^*$ from $\{(\boldsymbol{t}_i, \boldsymbol{R}_i)\}$.
return $\boldsymbol{t}^*, \boldsymbol{R}^*$

IV. Experiments

Our simulations and real data experiments are implemented in C++, and all experiments are performed on a 12th Gen Intel Core i9-12900H CPU. We compare our method to traditional 17-point algorithm implemented by ourselves in both cases. Four cameras are used, because Corollary 1 tells us at least three cameras are needed.

For both simulations and real data experiments, we use method in [6] to recover $\boldsymbol{R}$ and $\boldsymbol{t}$. And we measure performance by following criteria [7] [18]:

- Angle difference: $\arccos\left(\frac{\mathrm{tr}(\Delta R_{gt}\Delta R^{\mathrm{T}})-1}{2}\right)$,
- Relative translation error: $\frac{2(\|\Delta t_{gt}-\Delta t\|)}{\|\Delta t_{gt}\|+\|\Delta t\|}$,

where ΔR means relative translation between two frames, Δt means translation between two frames, ΔR_{gt} and Δt_{gt} correspond to its groundtruth counterparts.

A. Simulation Results

In order to test the influence of overlap, we set four cameras at the corners of a square. The square has sides measuring 9cm. The scene consists of 100 randomly placed landmarks within a distance of 2m to 5m from the origin. Each camera has a 180-degree field of view, allowing it to observe all points in front of it. The groundtruth trajectory is a circle with a radius of 1m, starting at $[0\mathrm{m}, 0\mathrm{m}, 0\mathrm{m}]^{\mathrm{T}}$. The multi-camera rig moves forward 0.36 degree per step, taking a total of 1000 steps to complete a full circle. The Plücker vector is perturbed within a cone with a half cone apex angle of $[-\mathrm{noise\_level\ rad}, \mathrm{noise\_level\ rad}]$. Some observations in each frame are selected as outliers, ensuring $\mathrm{outlier\_rate}$ is desired. The results are obtained from 10 trials.

First, we validate that the use of overlap in 17-point algorithm can harm accuracy, details are shown in Fig. 4. We use different number of overlap per track in 17-point algorithm, and report the mean and medium of above metrics. Additionally, we report these metrics of OpenGV[10] as well (red line), which represents a random number of overlap.

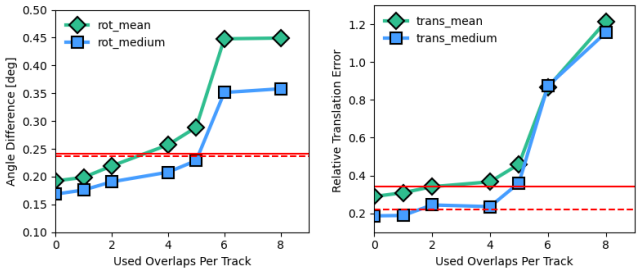

Fig. 4: Mean/medium of angle difference/relative translation error for traditional 17-pt algorithm under different number of used overlaps per track.

From Fig. 4, we can summarize that overlap can harm accuracy under the above simulation setting, where the depth-baseline ratio is large and observation noise exists. Next, we conduct four ablation studies:

- Only allow the number of RANSAC iterations $\mathrm{N_{iter}}$ to change, and fix $\mathrm{N_{pt}} = 27$, $\mathrm{noise\_level} = 0.0015$, $\mathrm{outlier\_rate} = 0.01$. We compare traditional 17-point algorithm with the proposed. The results are presented in Fig 5. One could see that improved 17-pt algorithm can increase accuracy.
- Only allow the used points $\mathrm{N_{pt}}$ to change, and fix $\mathrm{N_{iter}} = 10$, $\mathrm{noise\_level} = 0.0015$, $\mathrm{outlier\_rate} = 0.01$. The results are presented in Fig. 6. Also, as the point number increases, the gap between these two methods becomes smaller.
- Only allow the $\mathrm{noise\_level}$ to change, and fix $\mathrm{N_{iter}} = 10$, $\mathrm{N_{pt}} = 27$, $\mathrm{outlier\_rate} = 0.01$. The results are presented in Fig. 7. Note that the improved version of 17-pt algorithm can resist on noise.

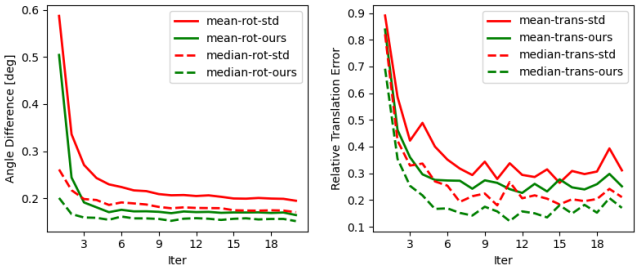

Fig. 5: Mean/medium of pose error for traditional 17-pt (red) and improved 17-pt (green) over different RANSAC iterations $\mathrm{N_{iter}}$, left shows angle difference, right shows relative translation error.

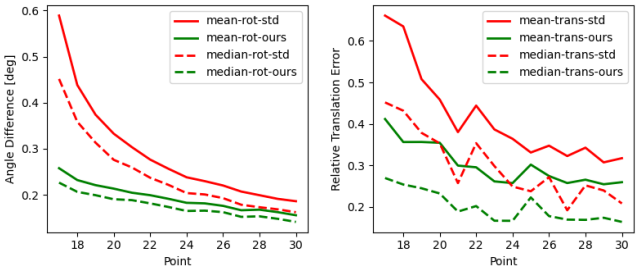

Fig. 6: Mean/medium of pose error for traditional 17-pt (red) and improved 17-pt (green) over different points $\mathrm{N_{pt}}$, left shows angle difference, right shows relative translation error.

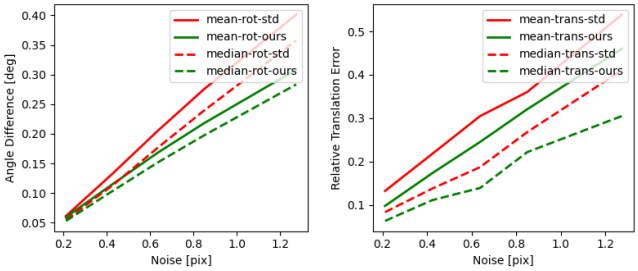

Fig. 7: Mean/medium of pose error for traditional 17-pt (red) and improved 17-pt (green) over different $\mathrm{noise\_level}$, left shows angle difference, right shows relative translation error.

B. Results on Real World Data

We test our method on real data by constructing a multi-camera rig and applying the improved 17-point algorithm to a collected short video sequence. The customized multi-camera rig consists of 4 cameras, which is made of two D435i, each with a pair of gray-scale cameras. The four cameras are synchronized using a microcontroller with a maximum time delay of less than 0.2ms. The multi-camera configuration is depicted in Fig. 8. We employ a frame-to-frame matching within a NOKOV motion capture-enabled environment which can provide the ground truth pose. Our system utilizes the ORB feature [19], employing the feature extraction and matching strategy from ORB_SLAM [4]. And then apply Alg. 1 to estimate relative pose. The frame-to-frame tracking results, obtained through an average of 3 trials, are displayed in Fig. 9. Note that the 17-point algorithm implemented by OpenGV does not consider overlap. Fig. 9 shows our method outperforms 17-point by 10% in both relative rotation and relative translation.

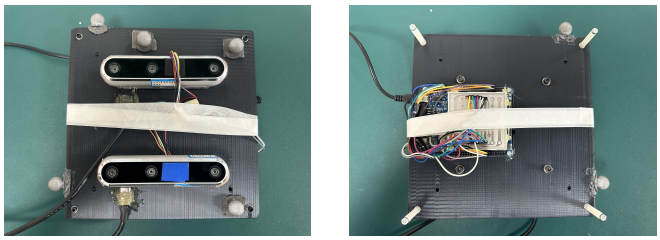

Fig. 8: Customized, synchronized multi-camera rig.

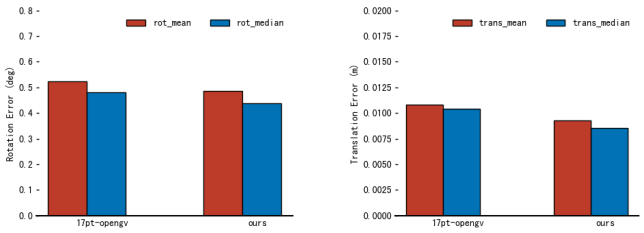

Fig. 9: Mean (red) and medium (blue) pose error for 17-point algorithm and Algorithm 1, left shows rotation error, right shows translation error, both improved by approximate 10%.

V. Conclusion

In this paper, we demonstrate that utilizing overlap can potentially solve the GEC estimation, but harm accuracy in practice. For the first time, we theoretically prove the necessary condition for successfully applying 17-point algorithm is to use a multi-camera rig which has more than 3 cameras. To increase accuracy, an improved 17-point algorithm is proposed, where we avoid to use overlap, and use a simple local optimization to solve the original GEC estimation problem. Both simulation and real data experiment show it outperforms the traditional 17-point algorithm in term of accuracy.

Appendix

A. Proof of Theorem 1

In general cases, assume $\boldsymbol{q_1}, \boldsymbol{q_2}, \boldsymbol{q_3}$ form the standard orthogonal basis of $\mathbb{R}^3$. For any $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$, we can express them by above basis: $\boldsymbol{q_i} = x'\boldsymbol{q_1} + y'\boldsymbol{q_2} + z'\boldsymbol{q_3}, \boldsymbol{q_j} = x\boldsymbol{q_1} + y\boldsymbol{q_2} + z\boldsymbol{q_3}$. We can easily get $\boldsymbol{z} = xx'\boldsymbol{\varepsilon_{11}} + xy'\boldsymbol{\varepsilon_{12}} + xz'\boldsymbol{\varepsilon_{13}} + yx'\boldsymbol{\varepsilon_{21}} + yy'\boldsymbol{\varepsilon_{22}} + yz'\boldsymbol{\varepsilon_{23}} + zx'\boldsymbol{\varepsilon_{31}} + zy'\boldsymbol{\varepsilon_{32}} + zz'\boldsymbol{\varepsilon_{33}}$ (remember $\boldsymbol{z}^{\mathrm{T}}$ is the row of $\boldsymbol{A}$), where

$$\boldsymbol{\varepsilon_{uv}} = \begin{bmatrix} \boldsymbol{q_u} \otimes \boldsymbol{q_v} \\ \boldsymbol{q_u} \otimes (\boldsymbol{t_1} \times \boldsymbol{q_v}) + (\boldsymbol{t_2} \times \boldsymbol{q_u}) \otimes \boldsymbol{q_v} \end{bmatrix},$$

Note that $\boldsymbol{t_1}$ is $\overrightarrow{O_1P}$ in $\mathcal{F}_{O_1}$, $\boldsymbol{t_2}$ is $\overrightarrow{O_2P}$ in $\mathcal{F}_{O_2}$.

We observe that the first nine coordinates of $\{\boldsymbol{\varepsilon_{uv}}\}$ are linearly independent: denote $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q_1} & \boldsymbol{q_2} & \boldsymbol{q_3} \end{bmatrix} \in \mathbb{R}^{3\times3}$, $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{q_1} \otimes \boldsymbol{q_1} & \boldsymbol{q_1} \otimes \boldsymbol{q_2} & \cdots & \boldsymbol{q_3} \otimes \boldsymbol{q_3} \end{bmatrix} \in \mathbb{R}^{9\times9}$. Then $\boldsymbol{S} = \boldsymbol{Q} \otimes \boldsymbol{Q}$, which indicates $\det\boldsymbol{S} = \det^3\boldsymbol{Q} \cdot \det^3\boldsymbol{Q} = \det^6\boldsymbol{Q} \neq 0$. Thus we can get $\{\boldsymbol{\varepsilon_{uv}}\}$ are linear independent to each other. Because there are 9 basis, $\mathrm{rank}(\boldsymbol{A}) \leq 9$. If $\min\{M, N\} = 1$, such as $M = 1$, note that $\boldsymbol{q_i}$ is a fixed vector, we can get $\mathrm{rank}(\boldsymbol{A}) \leq 3$ by similar process. ■

B. Proof of Theorem 2

We assume that every camera can observe all 3D points, which will potentially maximize $\mathrm{rank}(\boldsymbol{A})$.

(1) $M = N = 1$.

Suppose $\boldsymbol{q_{j1}}, \boldsymbol{q_{j2}}, \boldsymbol{e_t}$ are standard orthonormal basis of $\mathbb{R}^3$, in which $\boldsymbol{e_t}$ is the unit direction vector of $\overrightarrow{C_1C_2}$ in $\mathcal{F}_{O_2}$, then for any $\boldsymbol{q_{jk}}$, we can express this unit-vector by the above basis

$$\boldsymbol{q_{jk}} = x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t}. \tag{14}$$

Since $\overrightarrow{C_1P_k}$ is co-planar with $\overrightarrow{C_1C_2}$ and $\overrightarrow{C_2P_k}$, we have

$$\begin{aligned} \boldsymbol{R}\boldsymbol{q_{ik}} &= \alpha\boldsymbol{q_{jk}} + \beta\boldsymbol{e_t} \\ \Rightarrow\ \boldsymbol{q_{ik}} &= \alpha\boldsymbol{R}^T\boldsymbol{q_{jk}} + \beta\boldsymbol{R}^T\boldsymbol{e_t}. \end{aligned} \tag{15}$$

Through (14) and (15), we have follows for $\boldsymbol{z}$, where $\boldsymbol{z}^{\mathrm{T}}$ is the row of $\boldsymbol{A}$:

$$\begin{aligned} \boldsymbol{z} &= \begin{bmatrix} \boldsymbol{q_{jk}} \otimes (\alpha\boldsymbol{R}^{\mathrm{T}}\boldsymbol{q_{jk}} + \beta\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e_t}) \\ \boldsymbol{q_{jk}} \otimes (\boldsymbol{t_i} \times (\alpha\boldsymbol{R}^{\mathrm{T}}\boldsymbol{q_{jk}} + \beta\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e_t})) \\ +(\boldsymbol{t_j} \times \boldsymbol{q_{jk}}) \otimes (\alpha\boldsymbol{R}^{\mathrm{T}}\boldsymbol{q_{jk}} + \beta\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e_t}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_3 \otimes \boldsymbol{R}^{\mathrm{T}} & \boldsymbol{0}_{9\times9} \\ \boldsymbol{0}_{9\times9} & \boldsymbol{I}_3 \otimes \boldsymbol{R}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t}) \\ \otimes(\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t}) \\ (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t}) \\ \otimes(\boldsymbol{t_i} \times (\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t})) \\ +(\boldsymbol{t_j} \times (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t})) \\ \otimes(\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t}) \end{bmatrix} \\ &\simeq \begin{bmatrix} (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t}) \\ \otimes(\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t}) \\ (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t}) \\ \otimes(\boldsymbol{t_i} \times (\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t})) \\ +(\boldsymbol{t_j} \times (x\boldsymbol{q_{j1}} + y\boldsymbol{q_{j2}} + z\boldsymbol{e_t})) \\ \otimes(\alpha x\boldsymbol{q_{j1}} + \alpha y\boldsymbol{q_{j2}} + (\alpha z + \beta)\boldsymbol{e_t}) \end{bmatrix} \\ &= \alpha x^2\boldsymbol{\varepsilon_1} + \alpha xy\boldsymbol{\varepsilon_2} + x(\alpha z + \beta)\boldsymbol{\varepsilon_3} + \alpha y^2\boldsymbol{\varepsilon_4} \\ &\quad + y(\alpha z + \beta)\boldsymbol{\varepsilon_5} + \alpha xz\boldsymbol{\varepsilon_6} + \alpha yz\boldsymbol{\varepsilon_7} + z(\alpha z + \beta)\boldsymbol{\varepsilon_8}, \end{aligned}$$

where $\boldsymbol{t_i}$ is $\overrightarrow{O_1C_1}$, $\boldsymbol{t_j}$ is $\overrightarrow{O_2C_1'}$, and

$$\begin{aligned}
\boldsymbol{\varepsilon_1} &= \begin{bmatrix}(\boldsymbol{q_{j1}}\otimes\boldsymbol{q_{j1}})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{q_{j1}}\otimes\boldsymbol{q_{j1}})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_2} &= \begin{bmatrix}\begin{pmatrix}\boldsymbol{q_{j1}}\otimes\boldsymbol{q_{j2}}\\ +\,\boldsymbol{q_{j2}}\otimes\boldsymbol{q_{j1}}\end{pmatrix}^{\mathrm{T}} & \begin{pmatrix}(\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})\\ \times\,(\boldsymbol{q_{j1}}\otimes\boldsymbol{q_{j2}}+\boldsymbol{q_{j2}}\otimes\boldsymbol{q_{j1}})\end{pmatrix}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_3} &= \begin{bmatrix}(\boldsymbol{q_{j1}}\otimes\boldsymbol{e_t})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{q_{j1}}\otimes\boldsymbol{e_t})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_4} &= \begin{bmatrix}(\boldsymbol{q_{j2}}\otimes\boldsymbol{q_{j2}})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{q_{j2}}\otimes\boldsymbol{q_{j2}})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_5} &= \begin{bmatrix}(\boldsymbol{q_{j2}}\otimes\boldsymbol{e_t})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{q_{j2}}\otimes\boldsymbol{e_t})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_6} &= \begin{bmatrix}(\boldsymbol{e_t}\otimes\boldsymbol{q_{j1}})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{e_t}\otimes\boldsymbol{q_{j1}})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_7} &= \begin{bmatrix}(\boldsymbol{e_t}\otimes\boldsymbol{q_{j2}})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{e_t}\otimes\boldsymbol{q_{j2}})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\
\boldsymbol{\varepsilon_8} &= \begin{bmatrix}(\boldsymbol{e_t}\otimes\boldsymbol{e_t})^{\mathrm{T}} & \left((\boldsymbol{I_3}\otimes\boldsymbol{t_i}^{\wedge}+\boldsymbol{t_j}^{\wedge}\otimes\boldsymbol{I_3})(\boldsymbol{e_t}\otimes\boldsymbol{e_t})\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}.
\end{aligned}$$

Based on similar process in Proof A, we know that $\boldsymbol{\varepsilon_1}$ to $\boldsymbol{\varepsilon_8}$ are linearly independent to each other, therefore $\mathrm{rank}(\boldsymbol{A}) \leq 8$. It is worth noting that we already use epipolar constraint in (15), which equals to the epipolar constraint of $\boldsymbol{A}$.

(2) $M = N = 2$.

We express $\boldsymbol{q_{ik}}$ or $\boldsymbol{q_{jk}}$ under natural basis $\boldsymbol{e_1}, \cdots, \boldsymbol{e_9}$, and denote $\overrightarrow{O_1C_1}$ (in $\mathcal{F}_{O_1}$) or $\overrightarrow{O_2C_1'}$ (in $\mathcal{F}_{O_2}$) as $\boldsymbol{t_1}$, $\overrightarrow{O_1C_2}$ (in $\mathcal{F}_{O_1}$) or $\overrightarrow{O_2C_2'}$ (in $\mathcal{F}_{O_2}$) as $\boldsymbol{t_2}$. Because any correspondence only occurs in $\{(C_1, C_1'), (C_1, C_2'), (C_2, C_1'), (C_2, C_2')\}$, thus the row of $\boldsymbol{A}$ (denote as $\boldsymbol{z}^{\mathrm{T}}$) could be linearly express as follows

$$\begin{aligned}
\boldsymbol{z} = \sum_{k=1}^{9}\alpha_k\begin{bmatrix}\boldsymbol{e_k}\\ (\boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge})\cdot\boldsymbol{e_k}\end{bmatrix} + \beta_k\begin{bmatrix}\boldsymbol{e_k}\\ (\boldsymbol{t_2}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge})\cdot\boldsymbol{e_k}\end{bmatrix}\\
+\,\gamma_k\begin{bmatrix}\boldsymbol{e_k}\\ (\boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_2}^{\wedge})\cdot\boldsymbol{e_k}\end{bmatrix} + \delta_k\begin{bmatrix}\boldsymbol{e_k}\\ (\boldsymbol{t_2}^{\wedge}\oplus\boldsymbol{t_2}^{\wedge})\cdot\boldsymbol{e_k}\end{bmatrix},
\end{aligned}$$

where $\oplus$ means Kronecker sum: $\boldsymbol{X}\oplus\boldsymbol{Y} \triangleq \boldsymbol{X}\otimes\boldsymbol{I}_n + \boldsymbol{I}_m\otimes\boldsymbol{Y}, \boldsymbol{X}\in\mathbb{R}^{m\times m}, \boldsymbol{Y}\in\mathbb{R}^{n\times n}$. We have

$$\begin{aligned}
&\mathrm{rank}(\boldsymbol{A})\\
\leq\ &\mathrm{rank}\left[\begin{array}{c|c|c|c}\boldsymbol{I_9} & \boldsymbol{I_9} & \boldsymbol{I_9} & \boldsymbol{I_9}\\ \hline \boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge} & \boldsymbol{t_2}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge} & \boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_2}^{\wedge} & \boldsymbol{t_2}^{\wedge}\oplus\boldsymbol{t_2}^{\wedge}\end{array}\right]\\
=\ &\mathrm{rank}\left[\begin{array}{c|c|c|c}\boldsymbol{I_9} & \boldsymbol{0}_{9\times9} & \boldsymbol{0}_{9\times9} & \boldsymbol{0}_{9\times9}\\ \hline \boldsymbol{0}_{9\times9} & \boldsymbol{I_3}\otimes(\boldsymbol{t_1}-\boldsymbol{t_2})^{\wedge} & (\boldsymbol{t_2}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & \boldsymbol{0}_{9\times9}\end{array}\right]\\
=\ &9+\mathrm{rank}\left[\begin{array}{c|c}\boldsymbol{I_3}\otimes(\boldsymbol{t_1}-\boldsymbol{t_2})^{\wedge} & (\boldsymbol{t_2}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3}\end{array}\right]_{9\times18}\\
=\ &9+\mathrm{rank}\begin{bmatrix}\boldsymbol{I_3} & & \\ & \boldsymbol{I_3} & \\ & & (\boldsymbol{t_1}-\boldsymbol{t_2})^{\wedge}\end{bmatrix}_{9\times9} = 17, \qquad (16)
\end{aligned}$$

where we use the fact that $\mathrm{rank}[\boldsymbol{X}\mid\boldsymbol{Y}] = \mathrm{rank}(\boldsymbol{X}+\boldsymbol{Y})$ if $\mathrm{col}(\boldsymbol{X})\cap\mathrm{col}(\boldsymbol{Y}) = \boldsymbol{0}$, and also the fact that the rank of a skew-symmetric matrix is 2 at most. Remember we have an epipolar constraint, which indicates the geometry constraints between cameras and points. Therefore, $\mathrm{rank}(\boldsymbol{A})$ will further minus one, which concludes $\mathrm{rank}(\boldsymbol{A}) \leq 16$.

(3) $M = N \geq 3$.

We can easily reform $\boldsymbol{z}$ as follows ($\boldsymbol{z}^{\mathrm{T}}$ is the row of $\boldsymbol{A}$)

$$\begin{aligned}
\boldsymbol{z} &= \begin{bmatrix}\boldsymbol{q_{jk}}\otimes\boldsymbol{q_{ik}}\\ \boldsymbol{q_{jk}}\otimes(\boldsymbol{t_i}\times\boldsymbol{q_{ik}}) + (\boldsymbol{t_j}\times\boldsymbol{q_{jk}})\otimes\boldsymbol{q_{ik}}\end{bmatrix} \qquad (17)\\
&= \begin{bmatrix}\boldsymbol{I_9}\\ \boldsymbol{t_j}^{\wedge}\oplus\boldsymbol{t_i}^{\wedge}\end{bmatrix}\cdot(\boldsymbol{q_{jk}}\otimes\boldsymbol{q_{ik}}) \qquad (18)
\end{aligned}$$

where $\boldsymbol{t_i}$ is $\overrightarrow{O_1C_i}$ (in $\mathcal{F}_{O_1}$), $\boldsymbol{t_j}$ is $\overrightarrow{O_2C_j'}$ (in $\mathcal{F}_{O_2}$). Similar to proof of Thm. 2-(2),

$$\begin{aligned}
&\mathrm{rank}(\boldsymbol{A})\\
=\ &\mathrm{rank}\left(\begin{bmatrix}\begin{bmatrix}\boldsymbol{I_9}\\ \boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge}\end{bmatrix}\cdot\boldsymbol{Q}_{11} & \cdots & \begin{bmatrix}\boldsymbol{I_9}\\ \boldsymbol{t_N}^{\wedge}\oplus\boldsymbol{t_M}^{\wedge}\end{bmatrix}\cdot\boldsymbol{Q}_{MN}\end{bmatrix}\right)\\
=\ &\mathrm{rank}(\boldsymbol{T}\cdot\boldsymbol{Q}) \leq \min\{\mathrm{rank}(\boldsymbol{T}), \mathrm{rank}(\boldsymbol{Q})\} \leq \mathrm{rank}(\boldsymbol{T}),
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{T} &= \begin{bmatrix}\boldsymbol{I_9} & \cdots & \boldsymbol{I_9}\\ \boldsymbol{t_1}^{\wedge}\oplus\boldsymbol{t_1}^{\wedge} & \cdots & \boldsymbol{t_N}^{\wedge}\oplus\boldsymbol{t_M}^{\wedge}\end{bmatrix}_{18\times9MN},\\
\boldsymbol{Q} &= \begin{bmatrix}\boldsymbol{Q}_{11} & & \\ & \ddots & \\ & & \boldsymbol{Q}_{MN}\end{bmatrix}_{9MN\times KMN},\\
\boldsymbol{Q}_{ij} &= \begin{bmatrix}\boldsymbol{q}_{j1}\otimes\boldsymbol{q}_{i1} & \cdots & \boldsymbol{q}_{jK}\otimes\boldsymbol{q}_{iK}\end{bmatrix}_{9\times K}.
\end{aligned}$$

Through the block elementary transformation of matrix,

$$\begin{aligned}
&\mathrm{rank}(\boldsymbol{A}) \leq\\
&\mathrm{rank}\left(\begin{bmatrix}\boldsymbol{I_9} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots\\ \boldsymbol{0} & \cdots & (\boldsymbol{t_j}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & \cdots & \boldsymbol{I_3}\otimes(\boldsymbol{t_i}-\boldsymbol{t_1})^{\wedge} & \cdots\end{bmatrix}\right)\\
&= 9+\mathrm{rank}\left(\begin{bmatrix}\cdots & (\boldsymbol{t_j}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & \cdots & \boldsymbol{I_3}\otimes(\boldsymbol{t_i}-\boldsymbol{t_1})^{\wedge} & \cdots\end{bmatrix}\right)
\end{aligned}$$

Note that while $M = N \geq 3$,

$$\begin{aligned}
&\max\left\{\begin{array}{c}\mathrm{rank}\left(\begin{bmatrix}(\boldsymbol{t_2}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & (\boldsymbol{t_3}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3}\end{bmatrix}\right),\\ \mathrm{rank}\left(\begin{bmatrix}\boldsymbol{I_3}\otimes(\boldsymbol{t_2}-\boldsymbol{t_1})^{\wedge} & \boldsymbol{I_3}\otimes(\boldsymbol{t_3}-\boldsymbol{t_1})^{\wedge}\end{bmatrix}\right)\end{array}\right\}\\
&\leq \mathrm{rank}\left(\begin{bmatrix}\cdots & (\boldsymbol{t_j}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & \cdots & \boldsymbol{I_3}\otimes(\boldsymbol{t_i}-\boldsymbol{t_1})^{\wedge} & \cdots\end{bmatrix}\right)\\
&\leq 9.
\end{aligned}$$

Since $\boldsymbol{X}\otimes\boldsymbol{Y} = \boldsymbol{P}\cdot(\boldsymbol{Y}\otimes\boldsymbol{X})\cdot\boldsymbol{P}^{\mathrm{T}}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are square matrix, $\boldsymbol{P}$ is permutation matrix, we can only consider the first part of max operation. Also based on the block elementary transformation of matrix, if these cameras are not colinear, let

$$\begin{aligned}
\boldsymbol{t_2}-\boldsymbol{t_1} &= (v_{1x}, v_{1y}, v_{1z}),\\
\boldsymbol{t_3}-\boldsymbol{t_1} &= (v_{2x}, v_{2y}, v_{2z}).
\end{aligned}$$

We can verify

$$\begin{aligned}
&\mathrm{rank}\left(\begin{bmatrix}(\boldsymbol{t_2}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3} & (\boldsymbol{t_3}-\boldsymbol{t_1})^{\wedge}\otimes\boldsymbol{I_3}\end{bmatrix}\right)\\
=\ &\mathrm{rank}\left(\begin{bmatrix}\boldsymbol{I_3} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I_3} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \alpha\boldsymbol{I_3} & \beta\boldsymbol{I_3} & \gamma\boldsymbol{I_3}\end{bmatrix}\right)\\
=\ &9,
\end{aligned}$$

where $\alpha = v_{1y}v_{2z} - v_{1z}v_{2y}$, $\beta = v_{1x}v_{2z} - v_{1z}v_{2x}$, $\gamma = v_{1y}v_{2x} - v_{1x}v_{2y}$, they are non-zero under non-colinear case. Hence $\mathrm{rank}(\boldsymbol{A}) \leq 18$, considering the epipolar constraint, we have $\mathrm{rank}(\boldsymbol{A}) \leq 17$. ■

REFERENCES

[1] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[2] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[3] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[4] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[5] Pushyami Kaveti, Arvind Thamilchelvan, and Hanumant Singh. Design and evaluation of a generic visual slam framework for multi-camera systems. *arXiv preprint arXiv:2210.07315*, 2022.

[6] Robert Pless. Using many cameras as one. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–587. IEEE, 2003.

[7] Hongdong Li, Richard Hartley, and Jae-hak Kim. A linear approach to motion estimation using generalized camera models. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[8] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[9] Laurent Kneip and Hongdong Li. Efficient computation of relative pose for multi-camera systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 446–453, 2014.

[10] Laurent Kneip and Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2014.

[11] Banglei Guan and Ji Zhao. Affine correspondences between multi-camera systems for 6dof relative pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 634–650. Springer, 2022.

[12] Ji Zhao, Wanting Xu, and Laurent Kneip. A certifiably globally optimal solution to generalized essential matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12034–12043, 2020.

[13] Henrik Stewénius and David Nistér. Solutions to minimal generalized relative pose problems. Citeseer.

[14] Joao Campos, Joao R Cardoso, and Pedro Miraldo. Poseamm: A unified framework for solving pose problems using an alternating minimization method. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3493–3499. IEEE, 2019.

[15] Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions of the Royal Society of London*, (155):725–791, 1865.

[16] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

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

[18] Gim Hee Lee. A minimal solution for non-perspective pose estimation from line correspondences. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 170–185. Springer, 2016.

[19] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.