A Pose-Only Solution to Visual Reconstruction and Navigation

Qi Cai, Lilian Zhang, Yuanxin Wu, Wenxian Yu, and Dewen Hu

Abstract—Visual navigation and three-dimensional (3D) scene reconstruction are essential for robotics to interact with the surrounding environment. Large-scale scenarios and computational robustness are great challenges facing the research community to achieve this goal. This paper raises a pose-only imaging geometry representation and algorithms that might help solve these challenges. The pose-only representation, equivalent to the classical multiple-view geometry, is discovered to be linearly related to camera global translations, which allows for efficient and robust camera motion estimation. As a result, the spatial feature coordinates can be analytically reconstructed and do not require nonlinear optimization. Comprehensive experiments demonstrate that the computational efficiency of recovering the scene and associated camera poses is significantly improved by 2-4 orders of magnitude.

Index Terms—Visual navigation, 3D reconstruction, global optimization, global translation, pure rotation

1 INTRODUCTION

A VISUAL imaging system maps the 3D real world onto a two-dimensional image camera plane. One essential task of computer vision research is to recover a 3D scene and the camera poses at which the images were taken [1], [2]. As noted by Marr [3], humans perceive the real world through two main processes: image feature correspondence, followed by the computation and understanding of the 3D scene.

Visual computation efficiency and robustness have been long-standing bottleneck problems in 3D computer vision. In the applications of simultaneous localization and mapping or structure from motion, the technique of bundle adjustment (BA) is ubiquitously used in the reverse-imaging process of recovering the scene and the associated camera poses from a set of images. The BA technique plays a prominent role in computer vision, robotics, and digital photogrammetry applications [1], [4], [5], [6], [7]. It is essentially an iterative nonlinear optimization with respect to a highly dimensional parameter space of 3D feature coordinates and camera poses (sometimes including intrinsic camera parameters) [1], [2]; its performance heavily depends on initialization [8], [9], [10], [11], [12], [13]. However, special but not uncommon camera movements, such as collinear or small translations, typically lead to abnormal initialization [2], [12], [14], [15]. For a fine initialization, an incremental BA starting from a two-view optimization can be employed [6], [16], or alternatively, the pairwise relative poses can be used as inputs for optimally solving first the global rotation and then the global translation of each view. There exist a number of efficient and stable global rotation averaging methods [13], [17], [18], [19], [20], [21], [22]. However, the state-of-the-art global translation estimation methods often suffer from inaccurate inputs of relative translation caused by wrong feature matching or critical camera motions [14], [15], [23], [24]. For example, if the camera undergoes a pure rotation, we will obtain the correct relative rotation but a wrong relative translation [25]. It partially implies that, though both relying on relative poses as inputs, the estimation of global rotation is more stable than that of global translation.

These long-standing bottleneck problems might be argued to be insufficient visual geometry representation. The well-known co-planar relationship [26] of the camera baseline and the two projection rays characterizes the two-view imaging geometry but loses the chirality condition and depth information [1], [2], [25], [26], [27]. It may be argued that the multi-view geometry satisfying chirality needs to be further explored [27]. The two-view imaging geometry is actually equivalent to a pair of pose-only (PPO) constraints decoupling camera poses from 3D feature coordinates [25]. This paper is motivated by the PPO constraints and tries to answer the following questions:

1. Is it possible to extend the pose-only equivalent representation to the multiple-view imaging geometry?
2. Is it possible to optimally achieve 3D visual computing in the pose-only parameter space?

Based on the above motivations, this paper proposes a pose-only solution to visual reconstruction and navigation, hopefully easing the challenges of computational efficiency and robustness in 3D visual computing. The contributions of this paper are four-fold:

- **Pose-only imaging geometry.** The multiple-view imaging geometry is equivalently represented by camera
poses and image points, and notably, it is linearly related to camera global translations (see Fig. 1).

- **Efficient and robust global translation estimation.** The linear relationship is found to be instrumental in deriving efficient and robust global translation estimation, which does not rely on pairwise relative translations as the input.

- **Pose adjustment and analytical 3D reconstruction.** In contrast to the gold-standard BA, the so-called pose adjustment (PA), together with an analytical 3D reconstruction, achieves 3D visual computing by optimization on camera poses only.

- **Free of point-scattering phenomenon.** The proposed pose-only solution is free of a serious point-scattering phenomenon constantly evidenced in the state-of-the-art results.

The paper is organized as follows. Section 2 introduces the related works on global translation estimation and the BA parameterization. In Section 3, we present the equivalent pose-only representation for multi-view imaging geometry. It is shown that the representation is linearly related to and enables a linear estimation method of global translation. Section 4 solves the 3D visual computing by a pose-only optimization and a subsequent analytical feature coordinate reconstruction. Section 5 demonstrates the performance of the proposed pose-only algorithms in both simulation and real-world tests. The paper is concluded in Section 6.

## 2 RELATED WORK

The nonlinear optimization of a large-scale BA has been facing two challenges [4], [10], [11], [28]: benign initialization and fast solution to the normal equation.

To determine an initialization of global translation for subsequent BA optimization, the global rotation is usually assumed to be known. The relative translations can be expressed as $\lambda t_{i,j} = R_i(t_i - t_j)$ where $\lambda = \|t_i - t_j\|$. $t_{i,j}$ is a relative translation unit vector between the $i$-th and $j$-th views, and $R_i$ and $t_i$ are the global rotation and translation for the $i$-th view, respectively.

Most of the global translation methods minimize penalties formed by relative and global translations. A direct linear approach by Govindu [19] proposed a least-squared solution of global translation to the linear system $t_{i,j} \times R_i(t_i - t_j) = 0$ and refined the solution by an iterative reweighting scheme. Moreover, the work [29] gave a Lie-algebraic method of pose averaging and its robustness was further improved in [30]. Sim and Hartley [31] utilized the $L_\infty$ norm to minimize the angular error between relative and global translations, where the relative translation was extracted from the trifocal tensor. Moulon et al. [32] estimated the trifocal tensor by minimizing the $L_\infty$ reprojection error among triple views and then extracted reliable relative translations for global translation optimization. The objective function used $\|\lambda t_{i,j} - R_i(t_i - t_j)\| = 0$ under the $L_\infty$ norm. Wilson and Snavely [12] introduced a 1DSM method, which has a one-dimensional preprocessing step to remove relative translation outliers and uses a non-convex optimization in the squared chordal form. Ozyesil [33] pointed out that the $L_\infty$ norm is prone to pairwise translation outliers, and provided a robust penalty with a least unsquared deviations (LUD) form [15]. Goldstein et al. [34] utilized a penalty with the magnitude of the projection of $t_i - t_j$ onto the orthogonal complement of $t_{i,j}$ and minimized it by the alternating direction method of multipliers. Zhuang et al. [35] gave a geometric interpretation of the works [12], [15], [34] and developed a bilinear angle-based translation averaging (BATA) method. However, the above-mentioned methods might suffer from degeneracy. As pointed out by [23] and [24], all methods using the unit relative translations, e.g., [12], [15], [19], [34], [35], are ill-posed under collinear motion except for the trifocal-tensor-based ones [31], [36]. Besides collinear motion, parallel rigidity [33] also affects the global translation averaging methods that are based on the pairwise relative translation [24]. In order to solve the degeneracy problems, Jiang et al. [14] and Cui [23] proposed some kinds of global linear translation methods. Specifically, Jiang et al. [14] employed the triangular relationship among the views of camera triplets to constraint global translations and used the depth ratio to mitigate the problem of collinear motion. Cui [23] reconstructed the feature depth from the relative pose, and utilized the resultant common feature coordinate to enforce the relationship between image pairs. The recent work by Liu et al. [24] showed that the image observations could be used, together with relative poses, to help handle collinear motions, and presented an approximate linear translation form by calculating a non-linear term directly based on $t_{i,j}$.

It should be highlighted that the accuracy of all above-mentioned global translation averaging methods depends on that of pairwise relative translation. There exist a few works that do not require pairwise relative translation as inputs. Arie-Nachimson et al. [37] utilized the two-view co-planar relationship and image observations to directly find the global translation solution, which is unable to handle collinear motion [38]. By assuming that a reference plane is visible in all image, Rother [39] proposed a linear method based on an algebraic equation to directly solve camera translations as well as 3D feature coordinates. The size of the problem could become prohibitive because of large number of feature points [31]. Hartley and Shaffalitsky [40] optimized the $L_\infty$ norm reprojection error to jointly estimate global translations.
and 3D points using second-order cone programming, later generalized in [41], which involves a large number of unknowns and is computationally and memory expensive [31], [32]. Moreover, since the $L_\infty$ norm is sensitive to outliers, these methods need careful outlier removal [42], [43].

The BA parameter space typically includes camera poses and 3D feature coordinates. Civera et al. [44] utilized the parametrization of feature inverse depth to effectively handle far or low-parallax features. Zhao et al. [11] and Liu et al. [24] presented the parallax angle-based feature parametrization to further help deal with collinear features, which leads to better convergence and robustness. Note that these works do not reduce the high dimensionality of the BA parameter space. For a fast solution to the large-scale normal equation challenge of BA, the main ideas in the last two decades have been full utilization of the inherent sparsity property of the BA problem, and speedy computation by the Schur complement [4], [11], [45], [46]. To overcome the memory limit of a computer, the BA problem was transformed into a number of small-scale inter-connected BA sub-problems and then handled by distributed computers [8], [10].

3 Pose-only Imaging Geometry

3.1 Depth-Pose-Only Constraint

Consider a 3D feature point $X^W = (x^W, y^W, z^W)^T$ observed in $n$ images (or views). For $i = 1, 2, \ldots, n$, denote by $X_i = (x_i, y_i, 1)^T$ the normalized image coordinate of the 3D feature point in the $i$-th image (or, alternatively, view $i$), and by $R_i$ and $t_i$ the global rotation and global translation of the camera when taking the $i$-th image, respectively. The projection equation of the 3D feature point $X^W$ for the $i$-th image can be given by [1], [2]

$$X_i = \frac{1}{z_i} X^C_i = \frac{1}{z_i} R_i (X^W - t_i), \quad i = 1, 2, \ldots, n, \quad (1)$$

where $X^C_i = (x^C_i, y^C_i, z^C_i)^T$ is the coordinate of the 3D feature point in the camera frame corresponding to the $i$-th image, and $z^C_i > 0$ is the corresponding depth of the feature point. For $m$ 3D feature points observed in $n$ images, the multiple-view imaging relationship can be represented as [1], [2]

$$f \left( \{ X_k^W \}_{k=1,m}, \{ R_i, t_i \}_{i=1,n}, \{ X_{ki} \}_{k=1,m, i=1,n} \right) = 0, \quad (2)$$

where $X_k^W$ is the world coordinate of the $k$-th 3D feature; $R_i$ and $t_i$ denote the global rotation and translation of the camera when taking the $i$-th image, respectively; and $X_{ki}$ is the normalized image coordinate of the $k$-th 3D feature on the $i$-th image. It is well known that there is a global scale ambiguity in recovering camera poses and the 3D scene structure. For instance, for any rigid transformation $R$ and $t$ at a scale $\alpha$, the projection equation (1) is always valid for substitutions $R_i \rightarrow R_i R^T$, $t_i \rightarrow \alpha R(t_i + t)$, and $X^W \rightarrow \alpha R(X^W + t)$. Therefore, the discussions to follow are based on global scale ambiguity awareness. Denote by $(i, j)$ a pair of views consisting of the $i$-th and $j$-th images. The imaging equation for the view pair $(i, j)$ is [1], [2]

$$z^{C_j} X_j = z^{C_i} R_{ij} X_i + t_{i,j}, \quad (3)$$

where the relative rotation is $R_{ij} = R_i R_j^T$ and the relative translation is $t_{ij} = R_j(t_i - t_j)$. Left multiply the antisymmetric matrix $[X_j]_x$ on both sides of Equation (3),

$$z^{C_i} [X_j]_x R_{ij} X_i = - [X_j]_x t_{i,j}. \quad (4)$$

Taking the magnitude, we get

$$z^{C_i} = \left\| \frac{[X_j]_x t_{i,j}}{\theta_{ij}} \right\| \geq d^{(i,j)}, \quad (5)$$

where $\theta_{ij} = \| [X_j]_x R_{ij} X_i \|$. Similarly, left-multiplying the antisymmetric matrix $[R_{ij} X_i]_x$ on both sides of Equation (3) yields

$$z^{C_i} = \left\| \frac{[R_{ij} X_i]_x t_{i,j}}{\theta_{ij}} \right\| \geq d^{(i,j)} \cdot \quad (6)$$

Combining Equations (3), (5), and (6), the pose-only constraint for the two-view imaging geometry, called a pair of pose-only or PPO constraints [25], is obtained as

$$d^{(i,j)} X_j = d^{(i,j)} R_{ij} X_i + t_{i,j}. \quad (7)$$

Moreover, it can be proved that the PPO constraint is equivalent to the two-view imaging geometry [25]. This equivalency is valid even when there is only a pure rotation between the two views, namely, in the case of $\theta_{ij} = 0$. Regarding the $l$-th image ($l \neq i, j$), the view pair $(i, l)$ also satisfies the PPO constraint

$$d^{(i,l)} X_l = d^{(i,l)} R_{il} X_i + t_{i,l}. \quad (8)$$

and

$$d^{(i,l)} = d^{(i,j)} = z^{C_i}. \quad (9)$$

We name the relationship in Equation (9) as the depth-equal constraint of the 3D feature point on the $l$-th image. Note that for all $n$ images, there are $C_n^2$ PPO constraints and $C_n^3$ depth-equal constraints, which contain a great deal of redundancy.

Substitute Equation (9) into Equation (8),

$$d^{(i,j)} X_l = d^{(i,j)} R_{il} X_i + t_{i,l}. \quad (10)$$

Define a set

$$D(\zeta, \eta) = \left\{ d^{(i,j)} X_i = d^{(i,j)} R_{\zeta}, X_{\zeta} + t_{\zeta,i} | 1 \leq i \leq n, i \neq \zeta \right\}, \quad (11)$$

which represents a set of constraints that take views $\zeta$ and $\eta$ as the left- and right-base views, respectively. As this is related to poses and depths (which are functions of poses) only, we name it the depth-pose-only (DPO) constraint set for the 3D feature point. Note: it can be proved that the DPO constraint set (11) is equivalent to the projection equation (1); see Proposition 3 below. That is, for $m$ 3D feature points observed in $n$ images, the multiple-view imaging relationship (2) can be equivalently expressed in a pose-only form.
The depths $d_{i}^{(i,j)}$ and $d_{j}^{(i,j)}$ are linearly related to translation (Proposition 2), so the two-view PPO constraint (7) can be rewritten as a linear form of relative translation

$$ (X_{i}b_{i,j} - R_{i,j}X_{j}a_{i,j} - \theta_{i,j}t_{i,j})t_{i,j} = 0. $$

By analogy, the multiple-view DPO constraint (11) can also be linearly expressed in terms of relative translation

$$ \theta_{i,j}^{g}R_{i,j}X_{i}a_{i,j}^{g}t_{i,j,n} + \theta_{i,j}^{g}R_{i,j}X_{i}b_{i,j}^{g}t_{i,j,n} = 0, $$

where $a_{i,j}$ and $b_{i,j}$ are functions of rotations to be explicitly given in Proposition 2. Alternatively, the above expressions (13) and (14) can be readily expressed in terms of global translation. In the sequel, we will present another linear expression of the global translation.

### 3.2 Linear Global Translation Constraint

Currently, the global rotation averaging algorithms, such as Chatterjee and Govindu [13], perform fairly well. The remaining sub-section attempts to solve global translations preconditioned on known global rotations $\{R_{i}\}_{i=1..n}$.

Left multiply $[X_{i}]_{x}$ on both sides of the DPO constraint (11),

$$ 0 = [X_{i}]_{x}(d_{i}^{(\xi,n)}R_{i,j}X_{j} + t_{i,j}), \quad 1 \leq i \leq n, \ i \neq \xi. $$

According to Proposition 2 below,

$$ d_{i}^{(\xi,n)} = a_{\xi,n}^{T}t_{i,j,n}. $$

Substituting Equation (16) into Equation (15), we show that global translations satisfy the following linear homogeneous equations

$$ Bt_{n} + Ct_{i} + Dt_{i} = 0, \quad 1 \leq i \leq n, \ i \neq \xi, $$

in which

$$ B = [X_{i}]_{x}R_{i,j}X_{j}a_{i,j}^{T} \quad R_{\xi,n}, $$

$$ C = \theta_{i,j}^{g}[X_{i}]_{x}R_{i}, $$

$$ D = - (B + C). $$

If $[X_{i}]_{x}R_{i,j}X_{j} = 0$ for $1 \leq i \leq n, \ i \neq \xi$, then $B = C = D = 0$ and Equation (17) is always true for any global translation.

For all 3D feature points, denote by $t = (t_{1}^{T}, \ldots, t_{n}^{T})^{T}$ the concatenated global translation of $n$ images and rewrite Equation (17) as

$$ L \cdot t = 0, $$

where $L$ is a matrix comprising global rotations and normalized image coordinates. It can be proved that $\text{rank}(L) = 3n - 4$ when there are at least two 3D feature points satisfying $\theta_{\xi,n} \neq 0$ (see Proposition 6). Equation (19) is called the linear global translation (LiGT) constraint. Choose view $r$ as the global translation reference, namely,

$$ t_{r} = 0. $$

An estimate of global translation $\hat{t}$ ($t_{r}$ removed) can be obtained by solving the linear homogeneous Equation (19).

There are two $t$ with opposite signs; however, the right one can be readily identified by using Equation (16), that is, it should satisfy $a_{\xi,n}^{T}t_{\xi,n} \geq 0$.

Consequently, according to Propositions 3-5 below, the three representations of multiple-view imaging relationship, namely (2), (12), and (19), are equivalent. Equation (19) expresses the multiple-view imaging relationship as a linear constraint. Given global rotations, the LiGT constraint (19) enables a linear solution to global translations, which will be proved in the sequel to be theoretically immune to camera collinear movement and local pure rotation. Certainly, the accuracy of the obtained global translations would be affected by the quality of the given global rotations. If a higher pose accuracy is required, a proposed algorithm of pose adjustment (given in Section 4) can be used to further refine the camera poses. The 3D feature coordinates can be analytically recovered from the camera poses.

### 3.3 Propositions

This subsection collectively presents the important Propositions mentioned above. Specifically, Proposition 1 assures the fact that a 3D feature has one unique depth in a view; Proposition 2 proves the depth can be linearly expressed in terms of translation; Propositions 3-5 show that the three representations of multiple-view imaging relationship in Fig. 1 and the current section are equivalent; Proposition 6 provides a solid foundation of solvability of the LiGT constraint in Equation (19).

**Proposition 1.** $d_{i}^{(i,j)} = d_{j}^{(j,i)}$ for view pair $(i,j)$.

**Proof.** As $[a]_{x}Rb = [R[a]_{x}]_{x}b$ for any two three-dimensional vectors $a$ and $b$, we have from Equations (5) and (6)

$$ d_{i}^{(j,i)} = \frac{||[t_{j,i}]_{x}R_{i,j}X_{j}||}{||X_{j}||R_{i,j}} = \frac{R_{i,j}R_{i,j}^{T}t_{j,i}^{x}X_{j}^{x}}{R_{i,j}R_{i,j}^{T}X_{j}^{x}} = d_{j}^{(i,j)}. $$

Q.E.D.

**Proposition 2.** The depth can be linearly expressed in terms of translation, that is, $d_{i}^{(i,j)} = a_{i,j}^{T}t_{i,j}$ and $d_{j}^{(i,j)} = b_{i,j}^{T}t_{i,j}$, where $a_{i,j}^{T} = ([R_{i,j}X_{i}]_{x}X_{j})^{T}[X_{j}]_{x}$ and $b_{i,j}^{T} = ([R_{i,j}X_{i}]_{x}X_{j})^{T}[R_{i,j}X_{i}]_{x}$.

**Proof.** Left multiply $[X_{j}]_{x}$ on both sides of the PPO constraint (7),

$$ d_{i}^{(i,j)}[X_{j}]_{x}R_{i,j}X_{i} = - [X_{j}]_{x}t_{i,j}. $$

It indicates vectors $[X_{j}]_{x}R_{i,j}X_{i}$ and $[X_{j}]_{x}t_{i,j}$ are on the same line but in opposite directions, i.e.,
When there are at least two 3D feature points \( R_{i,j}X_i \),eq. Assume that the DPO constraint set (11) is valid, then in Equation (5), we have

\[
\frac{d_{i}^{(i,j)}}{\theta_{i,j}^2} t_{i,j} \leq a_{i,j}^T t_{i,j}.
\]  

(24)

Similarly, left multiply \([R_{i,j}X_i]_x \) on both sides of the PPO constraint (7),

\[
d_{j}^{(i,j)} [R_{i,j}X_i]_x X_j = [R_{i,j}X_i]_x t_{i,j}.
\]  

(25)

It means \([R_{i,j}X_i]_x X_j \) has the same direction with \([R_{i,j}X_i]_x t_{i,j} \), i.e.,

\[
\frac{[R_{i,j}X_i]_x X_j}{[R_{i,j}X_i]_x t_{i,j}} = \frac{[R_{i,j}X_i]_x t_{i,j}}{b_{i,j}^T t_{i,j}}.
\]  

(26)

Using \(d_{j}^{(i,j)} \) in Equation (6),

\[
d_{j}^{(i,j)} = \frac{b_{i,j}^T t_{i,j}}{\theta_{i,j}^2}.
\]  

(27)

Q.E.D.

Proposition 3. The DPO constraint set (11) ⇔ the projection Equation (1).

Proof. ⇔: According to the developments in Equations (1)–(10), the necessity is obvious.

⇒: Assume that the DPO constraint set (11) is valid, then

\[
d_{i}^{(i)} X_i = d_{\xi}^{(i)} R_{\xi}X + t_{\xi,i} \quad 1 \leq i \leq n, i \neq \xi.
\]  

(28)

In analogy to the treatment in Equations (3)–(5), left-multiplying \([X_i]_x \) and taking magnitude on both sides of Equation (28), we obtain

\[
d_{\xi}^{(i)} = \frac{||[X_i]_x t_{\xi,i}||}{||[X_i]_x R_{\xi}X||} = d_{\xi}^{(i)}, \quad 1 \leq i \leq n, i \neq \xi.
\]  

(29)

which means that the depth-equal constraint (9) is implicitly embedded in Equation (28).

For proof convenience, we choose the world frame to be the camera frame of view \( \xi \), that is, \([R_{\xi}X]_s = [I_3]_s \). Then

\[
X_\xi = \frac{1}{z_{\xi}^e} X^{\xi} = \frac{1}{z_{\xi}^e} X^W.
\]  

(30)

Express Equation (28) in terms of the global pose,

\[
d_{i}^{(i)} X_i = R_{i} \left( d_{\xi}^{(i)} X_\xi - t_i \right), \quad 1 \leq i \leq n, i \neq \xi.
\]  

(31)

According to Equations (5), (6), and (9), the above equation can be written as

\[
z_{\xi}^i X_i = R_{i} \left( X^W - t_i \right), \quad 1 \leq i \leq n, i \neq \xi.
\]  

(32)

Combining Equations (30) and (32), we obtain the projection Equation (1). Q.E.D.

Proposition 4. The LiGT constraint ⇒ the depth-equal constraint.

Proof. From (15), \( t_{\xi,i} \), \( X_i = d_{\xi}^{(i)} [X_i]_x R_{\xi}X \). Taking the magnitude on both sides,

\[
d_{\xi}^{(i)} = \frac{||[X_i]_x R_{\xi}X||}{||[X_i]_x R_{\xi}X||} = d_{\xi}^{(i)}.
\]  

(33)

which indicates that the depth-equal constraint (9) is embedded in the LiGT constraint. Q.E.D.

Proposition 5. The LiGT constraint ⇔ the DPO constraint.

Proof. ⇔: The necessity is true according to the derivation of the LiGT constraint in Equation (15).

⇒: From Equation (15), we know that \( X_i \) and \( d_{\xi}^{(i)} R_{\xi}X + t_{\xi,i} \) are parallel, that is,

\[
\lambda X_i = d_{\xi}^{(i)} R_{\xi}X + t_{\xi,i}.
\]  

(34)

As the normalized image coordinates \( X_i \) and \( X_\xi \) satisfy

\[
z_{\xi}^i X_i = z_{\xi}^\xi X_\xi + t_{\xi,i}.
\]  

(35)

Left multiply \([X_i]_x \), we obtain

\[
z_{\xi}^i [X_i]_x R_{\xi}X_\xi = -[X_i]_x t_{\xi,i}.
\]  

(36)

Taking the magnitude on both sides, \( z_{\xi}^i = ||[X_i]_x t_{\xi,i}|| / ||[X_i]_x R_{\xi}X_\xi|| = d_{\xi}^{(i)} \). According to Equation (33), \( z_{\xi}^\xi = d_{\xi}^{(i)} = d_{\xi}^{(i)} \).

Equalizing Equations (34) and (35), we obtain \( \lambda = z_{\xi}^i > 0 \). Left multiply \([R_{\xi}X_i]_x \) on both sides of Equation (34) and take the magnitude on both sides,

\[
\lambda = \frac{||R_{\xi}X_i]_x t_{\xi,i}||}{||R_{\xi}X_\xi]_x X_\xi||} = d_{\xi}^{(i)}.
\]  

(37)

Substituting into Equation (34) yields the DPO constraint, namely,

\[
d_{i}^{(i)} X_i = d_{\xi}^{(i)} R_{\xi}X + t_{\xi,i}.
\]  

(38)

Q.E.D.

It is important to stress that Proposition 5 implies the left cross product in Equation (15) does not result in any information loss, in contrast to the loss of the chirality condition and depth information in the essential equation \([1], [2], [25], [26], [27]\). Intuitively, the physical image coordinates in Equation (34) have to satisfy an implicit constraint of the two-view imaging Equation (3). As a result, the \( \lambda \) is not arbitrary but unique.

Proposition 6. When there are at least two 3D feature points with different image points such that \( \theta_{\xi} = \|X_\xi \times R_{\xi}X\| \neq 0 \), \( rank(L) = 3n - 4 \).
Proof. For the true global translation vector \( t_{true} = (t_1^T, \ldots, t_n^T)^T \), the LiGT constraint implies
\[
Bt_n + Ct_i + Dt_\xi = 0, \quad 1 \leq i \leq n, i \neq \xi,
\]
that is,
\[
L \cdot t_{true} = 0.
\]

(40)

Take nonzero constants \( \alpha_i \in \mathbb{R} \) for \( i = 0, 1, 2, 3 \). Denote by \( e_1, e_2, e_3 \) the columns of the three-dimensional identity matrix and by \( \bar{t} = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 \) any three-dimensional vector. For the global translation of each view, make a linear transformation of global scaling and translating as
\[
t'_i = \alpha_0 t_i + \bar{t}.
\]

(41)

Substituting into Equation (39),
\[
Bt'_n + Ct'_i + Dt'_\xi
= B(\alpha_0 t_n + \bar{t}) + C(\alpha_0 t_i + \bar{t}) + D(\alpha_0 t_\xi + \bar{t})
= \alpha_0 (Bt_n + Ct_i + Dt_\xi) + (B + C + D) \bar{t}
= 0,
\]
where \( 1 \leq i \leq n, i \neq \xi \). Applying transformation (41) to each component translation in \( t_{true} \), that is,
\[
t'_i = \alpha_0 t_i + \bar{t},
\]
where \( \bar{T} = w \otimes \bar{t} \) is a \( 3n \times 1 \) vector, \( w = (1, \ldots, 1)^T \) is a \( n \times 1 \) constant vector, and \( \otimes \) denotes the Kronecker product. Note that
\[
\bar{T} = \alpha_1 (w \otimes e_1) + \alpha_2 (w \otimes e_2) + \alpha_3 (w \otimes e_3)
= \alpha_1 \xi_1 + \alpha_2 \xi_2 + \alpha_3 \xi_3.
\]

(43)

Substituting Equation (43) yields
\[
t'_i = \alpha_0 t_{true} + \alpha_1 \xi_1 + \alpha_2 \xi_2 + \alpha_3 \xi_3.
\]

(44)

(45)

Obviously, the nonzero vectors \( \xi_1, \xi_2, \) and \( \xi_3 \) are linearly independent. Now, let us suppose \( t_{true} \) could be linearly represented by \( \xi_1, \xi_2, \) and \( \xi_3 \), that is,
\[
t_{true} = k_1 \xi_1 + k_2 \xi_2 + k_3 \xi_3
= k_1 (w \otimes e_1) + k_2 (w \otimes e_2) + k_3 (w \otimes e_3)
= w \otimes (k_1 e_1 + k_2 e_2 + k_3 e_3)
\]
\[
\Delta = w \otimes \beta
\]
\[
= (\beta^T, \ldots, \beta^T)^T.
\]

(46)

This means that the camera motions of all views are pure rotations, that is, \( \theta_{i,j} = 0 \) for any view pair \((i, j)\), which is a contradiction. Thus, \( t_{true} \) is linearly independent of \( \xi_1, \xi_2, \) and \( \xi_3 \). With Equation (42),
\[
L \cdot t'_{true} = 0.
\]

(47)

From Equations (45) and (47), we can see that the four \( 3n \times 1 \) nonzero vectors \( t_{true}, \xi_1, \xi_2, \) and \( \xi_3 \) constitute a four-dimensional null space. Therefore, \( \text{rank}(L) \leq 3n - 4 \).

Suppose that a vector \( \xi_i \) exists, which is linearly independent of \( t_{true}, \xi_1, \xi_2, \) and \( \xi_3 \), but satisfies \( L \cdot \xi_i = 0 \). Then,
\[
B\xi_{n,i} + C\xi_1,4 + D\xi_2,4 = 0, \quad 1 \leq i \leq n, i \neq \xi
\]
where \( \xi_i \) is the global translation of view \( i \) in \( \xi_i \).

Let \( y_{i,n} = t_i - t_n \), then \( y = (y_{1,n}, \ldots, y_{n,n})^T = t_{true} - (w \otimes t_n) \). For \( i = \eta \) in Equation (48), we have
\[
(B+C)\xi_{n,i} + D\xi_{2,i} = 0.
\]

(49)

In view of \( B + C + D = 0 \), the above equation becomes
\[
D(\xi_{2,i} - \xi_{n,i}) = 0.
\]

(50)

Similarly for \( i = \eta \) in Equation (39), we have
\[
D(t_{\xi - t_n}) = 0.
\]

(51)

According to the precondition we assumed, Equations (50) and (51) are supposed to be valid for at least two 3D feature points. That is to say, there is a \( D' \) satisfying both Equations (50) and (51). This can be collectively written as
\[
G(\xi_{n,i} - \xi_{n,i}) = G(t_{\xi - t_n}) = 0,
\]
where \( G \) is some \( (D' \times 3) \) matrix. The precondition that there are at least two 3D feature points means \( \text{rank}(G) \geq 2 \). If \( \text{rank}(G) = 3 \), then \( t_\xi - t_n = 0 \) and all image points satisfy \( \theta_{i,n} = \|X_{n} \times R_{\xi,n}X_{\xi} \| = 0 \), which contradicts the precondition. Therefore, \( \text{rank}(G) = 2 \). In this regard, Equations (50) and (51) imply
\[
\xi_{n,i} - \xi_{n,i} = \lambda(t_{\xi - t_n}) = \lambda y_{i,n},
\]

in which \( \lambda \) is a scalar.

From Equation (48) and (52), we then obtain
\[
0 = (B + C - C)\xi_{n,i} + C\xi_1,4 + D\xi_2,4
= \lambda Dy_{i,n} + C(\xi_1,4 - \xi_{n,i})
\]

(53)

Similarly, from Equation (39) we obtain
\[
0 = (B - C + D)t_n + Ct_i + Dt_\xi
= Dy_{i,n} + C\xi_{i,n}.
\]

Substituting into Equation (53),
\[
C(-\lambda y_{i,n} + \xi_{i,4} - \xi_{n,4}) = 0.
\]

(55)

Note that, from Equation (18), \( C = g^2\xi_{i,4}[X_{i,4}R_i \]. There are at least two 3D feature points satisfying \( \|X_{n} \times R_{\xi,n}X_{\xi} \| \neq 0 \), as assumed in the precondition; therefore, Equation (55) indicates that \( [X_{i,4}R_i(-\lambda y_{i,n} + \xi_{i,4} - \xi_{n,4})] \) can only be a zero vector, that is,
\[
\xi_{i,4} - \xi_{n,4} = \lambda y_{i,n}.
\]

(56)
Combining Equations (52) and (56), we obtain

\[ \xi_{i;4} = \xi_{5;4} = \lambda y_{\xi;5}. \]  

(57)

So far, we obtain \( \xi_{5;4} = \lambda y_{\xi;4} + \xi_{n;4} \) and \( \xi_{i;4} = \lambda y_{\xi;4} + \xi_{n;4} \) for \( 1 \leq i \leq n, i \neq 5 \). By analogy, for all 3D feature points, we have

\[ \xi_{i} = \lambda y_{\xi} + (w \otimes \xi_{n}) \\
= \lambda (t_{\text{true}} - (w \otimes t_{\theta})) + (w \otimes \xi_{n}) \\
= \lambda t_{\text{true}} + w \otimes (\xi_{n;4} - \lambda t_{\theta}). \]  

(58)

The three-dimensional vector \( \xi_{n;4} - \lambda t_{\theta} = \alpha_{1} e_{1} + \alpha_{2} e_{2} + \alpha_{3} e_{3} \), that is, \( \xi_{i} \) is linearly dependent on \( t_{\text{true}} \), \( \xi_{4} \), and \( \xi_{3} \), which is a contradiction. Therefore, \( \text{rank}(L) = 3n - 4 \). Q.E.D.

For a global pure rotation, \( \theta_{\xi;0} = 0 \) for all 3D feature points and the LiGT constraint can never generate the right global translation; neither can the projection equation, nor the DPO constraint. However, as a scenario violating the Proposition 6 preconditions only occurs theoretically, Proposition 6 actually indicates that the global translation can almost certainly be solved from the LiGT constraint, even under special but common movements such as collinear motion or local pure rotation.

4 POSE-ONLY ALGORITHMS

4.1 Pose Adjustment

The gold-standard BA minimizes the reprojection error formulated by the projection Equation (1). Denoting by \( \hat{X}_{i} \), the error-contaminated normalized image coordinate of a 3D feature point in the \( i \)-th image, the reprojection error is usually defined as

\[ V_{i} = X_{i} - \hat{X}_{i} = \frac{Y_{i}^{BA}}{e_{i}^{T}Y_{i}^{BA}} - \hat{X}_{i}, \]  

(59)

where \( Y_{i}^{BA} = R_{i}(W^{i} - t_{i}) \) and \( e_{i}^{T} = (0, 0, 1) \). For \( m \) 3D feature points observed in \( n \) images, a reprojection error vector \( V_{BA} \) can be formed. The error function in the BA minimization can be expressed as

\[ \varepsilon_{BA}(\{X_{k}^{W}\}_{k=1..m}, \{R_{i}, t_{i}\}_{i=1..n}, \{\hat{X}_{k;i}\}_{k=1..m, i=1..n}) = V_{BA}^{T}V_{BA}. \]  

(60)

The corresponding BA minimization problem is formulated as \([1], [2]\)

\[ \arg\min \varepsilon_{BA} \]  

(61)

As there are typically a large number of 3D features in a scene, we can imagine that the Equation (61) is a nonlinear optimization problem in a high-dimensional parameter space. With the DPO constraint set in Equation (11), the reprojection error for a 3D feature point is given by

\[ V_{i} = X_{i} - \hat{X}_{i} = \frac{Y_{i}^{PA}}{e_{i}^{T}Y_{i}^{PA}} - \hat{X}_{i}, \]  

(62)

where \( Y_{i}^{PA} = \hat{\theta}_{\xi;\eta}((d\xi_{\eta})R_{i}X_{\xi} + t_{\xi;i}) = ||[t_{\xi;i}]_{x}X_{\xi}^{	ext{r}}||R_{i}X_{\xi} + \hat{\theta}_{\xi;\eta}t_{\xi;i}. \) For \( m \) 3D feature points observed in \( n \) images, a reprojection error vector \( V_{PA} \) can be formed. The error function in the minimization can be expressed as

\[ \varepsilon_{PA}(\{R_{i}, t_{i}\}_{i=1..n}, \{\hat{X}_{k;i}\}_{k=1..m, i=1..n}) = V_{PA}^{T}V_{PA}. \]  

(63)

The corresponding minimization problem is formulated as

\[ \arg\min \varepsilon_{PA}, \{R_{i}, t_{i}\}_{i=1..n} \]  

(64)

the unknown parameters of which consist of camera poses only. Therefore, it is referred to as the PA algorithm throughout the paper.

4.2 Global Analytical Reconstruction

The 3D multiple-view scene structure can be analytically reconstructed from the obtained camera poses. For a 3D feature point, its depth in the left-base view is calculated as

\[ z_{\xi}^{W} = \sum_{1 \leq i \leq n, i \neq \xi} \omega_{\xi,i} d_{\xi;i} \]  

(65)

where \( \omega_{\xi,i} \) is the weighting coefficient. According to the two-view case [25], \( \theta_{\xi;i} \) is a quality indicator of reconstruction, and thus, we take the weighting coefficient as \( \omega_{\xi,i} = \theta_{\xi;i}/\sum_{1 \leq i \leq n, i \neq \xi} \theta_{\xi,i} \). Finally, the 3D feature coordinate is given by

\[ X_{i}^{W} = z_{\xi}^{W} R_{\xi} X_{\xi} + t_{\xi}. \]  

(66)

The world coordinates of \( m \) 3D features observed in \( n \) images can be represented entirely by camera poses and image points as \( \{X_{k}^{W}\}_{k=1..m} = z(\{R_{i}, t_{i}\}_{i=1..n}, \{X_{k;i}\}_{k=1..m, i=1..n}) \).

4.3 Pose-only Algorithms for Recovering Camera Poses and the 3D Scene

The algorithm applies to \( m \) 3D feature points and \( N \) images (or views). Note that all feature points are not necessarily observed in each image here, in contrast to the theoretical discussions for brevity there in Section 3. It assumes that the camera global rotations are given beforehand, which are typically acquired by global rotation averaging algorithms from relative rotation estimates.

Input: Normalized image coordinates.
Output: Global poses and 3D feature coordinates.
Step 1. Relative rotation estimation and global rotation averaging.
Step 2. Designate a view, say view \( r \), as the reference view. Set the constraint \( t_{r} = 0 \).
Step 3. For the current 3D feature point \( X_{k}^{W} \) observed in \( n \) (\( \leq N \)) images, select left/right-base views using the following criterion

\[ (\xi, \eta) = \arg\max_{1 \leq i \leq n} \{\theta_{\xi,i}\}. \]  

(67)

Step 4. Build the matrix \( L \) using Equations (17) and (18).
Step 5. For all 3D feature points, repeat Steps 2-3. Obtain the global translation \( \hat{\ell} \) by solving Equation (19).
6. Identify the right global translation solution using \( a_t^T t_{c x} \geq 0 \).

Step 7 (Optional). Implement pose adjustment to further improve camera poses according to Equation (64).

Step 8. Analytically reconstruct all 3D feature coordinates using Equations (65) and (66).

The flow chart of the pose-only solution is given in Fig. 2. The performance of LiGT depends on the quality of given global rotations. For sufficiently good global rotations, the global translation estimates of LiGT would be of comparably good quality, so are the analytically reconstructed 3D feature coordinates. In this regard, the PA adjustment in Step 7 might be optional depending on the requirement of reconstruction accuracy and time.

5 Experiments

The experiment was performed on an Ubuntu 18.04.4 LTS platform, with 128 GB memory and Intel Xeon(R) Platinum 8269CY CPU @ 2.50 GHz, one core. The LiGT algorithm was developed based on Spectra and Eigen C++ libraries, and the PA algorithm was developed using the SparseLM optimization library. The experiment consisted of both real and simulation data tests, throughout which the set of parameters are unchanged. Note that the reprojection errors have been regularized uniformly for all algorithms by way of BA’s minimization function (60), using their own estimates of camera poses and 3D feature coordinates.

We utilized the OpenGV [47] library for relative poses by common two-view processing algorithms [26], [47], [48], [49], [50]. The state-of-the-art libraries of global structure from motion (SfM), such as OpenMVG [36] and Theia Vision [51], are mainly based on the global rotation averaging algorithms proposed by Chatterjee [18], Hartley [20], [21], and Martinet [52]. Comparably, Chatterjee’s newest algorithm [13] has the best accuracy and robustness [12], [15] and thus was used to provide the global rotation of each view.

Recovering the global translation is a key problem in SfM. We compared the most commonly-used global translation averaging algorithms in our tests. LUD is found to be robust and outstanding, and thus, it is mainly shown in real data tests.

Most state-of-the-art libraries incorporate the Google Ceres BA [53], although there have been a number of studies focused on speeding-up BA, such as sBA [4], ssBA [54], and PBA (or PMBA) [11], [24]. This study addresses the standard case of the calibrated camera; therefore, the Google Ceres BA (version 1.14.0) was taken as the benchmark for algorithm assessment.

Since OpenGV provides RANSAC to obtain robust relative estimation, and the global rotation averaging algorithm by Chatterjee uses the iterative reweighted least squares method to depress outliers, no additional outlier handling was employed in the experiment. The running time excludes data file reading and writing. The memory cost is calculated using the Intel VTune Profiler [55].

5.1 Simulation Test Performance

Three types of camera motions (linear, circular, and square) were designed in the simulation test to compare the algorithms in terms of accuracy and robustness. The distance between two adjacent views is denoted by \( D_t \).

Linear motion test: to simulate an open-loop collinear motion condition. The movement is along the x-axis and the distance \( D_t \) for neighbouring views is subject to a Gaussian distribution \( N(D_t, Dt/10) \).

Circular closed-loop test: to simulate a closed-loop, not-collinear condition. The radius of the circle is \( r = D_t \cdot N/2\pi \), where \( N \) is the number of views.

Square closed-loop test: to simulate a closed-loop and collinear condition. The translation is generally along a square but slightly disturbed in three directions. The side of the square is \( s = D_t \cdot N/4 \). The camera translation is perturbed in three directions by adding a small amount of \( N(0, Dt/10) \).

The distance between neighbouring views and the image point noise standard variance (std) are controlling parameters. The inter-distance is used to assess the algorithms under different translation magnitudes, and the noise std is used to evaluate the algorithms’ sensitivity to noise.

There are 100 images and 10000 randomly distributed 3D feature points in each simulation scenario. The distance between two adjacent views is set to \( D_t = [0.1, 5, 10, 20, 30] \) meters (with a fixed image noise std of 1 pixel), and the image noise std is set to \( [0.1, 1, 2, 5, 10] \) pixels (with a fixed distance of \( D_t = 10 \) meters). The 3D feature points were generated uniformly in front of the camera, ranging 0-150 meters.
The state-of-the-art algorithms in Theia were mainly assessed, including the global translation algorithms LUD and 1DSfM, and the Google Ceres BA. The LinearSfM method in Theia is not presented in simulation results because of its instability in our tests. Under collinear motion or small translation, the 1DSfM algorithm was found to delete related images and, thus, has not been included for comparison. The LiGT algorithm, as well as the PA algorithm initialized with the LiGT algorithm (LiGT-PA), are compared against state-of-the-art counterparts: the LUD algorithm, to determine global translations, and the Google Ceres BA algorithm initialized with the LUD algorithm (LUD-BA).

Fig. 4 presents the average algorithm accuracy across 50 Monte Carlo runs. Note that all global translation algorithms take the common input of global rotations, so LiGT, LUD, and 1DSfM have the exact same accuracy in global rotation.

**Accuracy of global translation.** LiGT is superior to LUD and 1DSfM in terms of global translation accuracy for all three types of camera motions.

**Robustness of optimization algorithm.** For normal inter-distance and moderate image noise, LiGT-PA and LiGT-BA are comparable in accuracy, but the former is less affected by image noise. For instance, the accuracy of LiGT-PA is obviously better than that of LiGT-BA if the image noise std surpasses two pixels. LUD-BA does not appear to be particularly stable and is sometimes inferior to LiGT (see the subplots of a3, a6, and c6 in Fig. 4).

**Local small translation.** To demonstrate the performance of LiGT in local pure rotation motion, a circular closed-loop test is noiselessly simulated by 10 images and one thousand 3D points, of which half of the images are located at the 2nd camera’s position (see Fig. 5a). Fig. 5b shows that the LiGT algorithm remarkably outperforms all other global translation algorithms in this local pure rotation case. The LinearSfM algorithm was tested but failed in this case of local pure rotation simulation. Note that the LiGT’s best performance has been achieved without any special outlier handling, which can be well predicted from Proposition 6. In contrast, most other global translation algorithms have spent great efforts on handling outliers brought about by the pairwise relative translation that is error-prone in the case of small translation motions.

**Global small translation.** When the camera motions are all close to pure rotations (e.g., $D_t = 0.1$ meters), all algorithms show divergence trends due to the well-known inherent singularity.

### 5.2 Real Test Performance

We performed a number of real tests on over 60 data from public datasets of Lund, OpenSLAM, and Strecha [56], [57], [58]. Due to page limit, only representative results are presented here. Interested readers are referred to [59] for additional results.

**Computational costs and accuracy.** The computational costs (in terms of running time and memory cost) and the reprojection errors across real data tests of Lund and OpenSLAM are summarized in Fig. 6, clockwise in ascending order of the number of image points. Compared with LUD-BA, LiGT-PA reduces, on average, the running time by approximately 25%.
Strecha dataset, which possesses the ground truth of camera results across a number of well-known algorithms for both computational cost and accuracy. PA considerably outperforms the state-of-the-art algorithms in those of LiGT-PA. Figs. 6 and 7 clearly indicate that LiGT and further reduced by 1-2 orders of magnitude, approaching used as the input instead. The reprojection errors of LiGT are lying on each vertical axis denote the reprojection errors of LUD and LiGT when global rotations refined by LiGT-PA are recovered by LiGT or LiGT-PA, which shows that the probable analytical 3D scene reconstruction is more robust in the LUD result that barely shows the scene outlines. The scene recovery of LUD-BA is considerably inferior to that of LiGT-PA. For example, the long stone wall in King’s College Cambridge, Statue of Liberty, University of Western Ontario, and Monastery in Ystad taken on a curve path. The camera trajectory is relatively complex for University of Western Ontario. In all four data, there exists local small translation or approximate pure rotation, to different extent, while taking photos from various view angles at stops.

The results in Fig. 7a show that the LiGT’s 3D scene recovery is very close to that of LUD-BA or LiGT-PA, in contrast to the LUD result that barely shows the scene outlines. The scene recovery of LUD-BA is considerably inferior to that of LiGT-PA. For example, the long stone wall in King’s College Cambridge, the goddess body in Statue of Liberty, the right-side wall in University of Western Ontario, and the roof in Monastery in Ystad. LUD-BA encounters apparent failure in recovering the lower-right part of camera poses in University of Western Ontario. In all four data, there exists local small translation or approximate pure rotation, to different extent, while taking photos from various view angles at stops.

Table 1 lists the global translation estimation accuracy results across a number of well-known algorithms for the Strecha dataset, which possesses the ground truth of camera poses in favor of direct accuracy comparison. As these algorithms rely on the input of global rotation, both the optimal estimate and the Chatterjee method [13] have been considered as inputs. LiGT stands out among all global translation estimation algorithms and performs consistently and significantly better than the others, followed by LUD in terms of accuracy and reliability. LinearSfM performs quite well in three datasets but fails in others. Among the three optimization methods, all initialized by LiGT, PA performs best in five out of six datasets and is only marginally worse than PMBA and BA in Herz-Jesus-P25. Note that LiGT is very close to the optimization methods, especially when using the optimal rotation as the input. It should be noted that the absolute values in Table 1 are affected by the result of feature matches, which are provided by OpenMVG in this paper.

Robustness. The BA optimization, especially LUD-BA algorithm, exhibits an apparent point-scattering phenomenon, in that many of the recovered 3D feature points are scattered distantly from the main structure (see Fig. 8). This phenomenon is inherent to many types of BA algorithms including PMBA and is considerably more severe under challenging conditions such as small translation or low image quality. In contrast, LUD-PA and LiGT-PA have the smallest and quite similar point-scattering degrees, followed by LiGT; LUD-BA and LiGT-BA yield worse not better situations than LUD and LiGT, respectively. Regarding #38 Pumpkin in Fig. 8, the recovered 3D points are identified as green scattering ones according to the LUD-BA final result, if they locate far away from the center of all camera positions. The numbers in the global view are the maximum absolute coordinate of all recovered 3D feature points along each axis. The smallest number of LUD-PA convincingly demonstrates the strong robustness of PA and the 3D feature coordinate analytical reconstruction. In the local view of LUD-BA, the camera poses at the bottom-right part clustered together, in contrast to those of LUD-PA. The scattering points in LUD or BA are actually useful points in scenes recovered by LiGT or LiGT-PA, which shows that the proposed analytical 3D scene reconstruction is more robust in these unbenign conditions. The global/local views of LiGT-related algorithms are not given because their point-scattering degrees are quite close to each other (see Fig. 8b).

Fig. 8 shows that, even under special camera motions such as small translation or collinear movement, LiGT and PA generally perform the best.
TABLE 1
Translation Accuracy (mm) of Strecha Dataset

| Data/Methods   | OpenMVG [36] | LinearSfM [14] | BATA [35] | 1DSfM [12] | Dalalyan [42] | LUD [15] | LiGT | PA | PMBA [24] | BA [53] |
|---------------|--------------|----------------|-----------|------------|--------------|----------|------|---|-----------|---------|
| Herz-Jesus-P8 | 98.94 (139.62) | 5.78 (5.88) | 473.14 (592.62) | 50.63 (48.94) | 123.71 (504.37) | 76.54 (61.99) | 3.2 (5.01) | 3.38 | 3.45 | 3.45 |
| Herz-Jesus-P25 | 225.97 (223.37) | / | 58.67 (51.82) | 25.75 (27.31) | 124.93 (120.10) | 11.65 (12.29) | 5.23 (6.86) | 5.20 | 5.15 | 5.15 |
| Fountain-P11  | 184.46 (194.23) | 3.95 (4.02) | 53.02 (242.29) | 10.12 (10.09) | 64.60 (63.79) | 19.53 (18.97) | 2.56 (3.17) | 2.48 | 2.54 | 2.54 |
| Entry-P10     | 135.22 (256.87) | 148.97 (149.38) | 409.62 (381.11) | 328.28 (329.45) | 1542.87 (1541.42) | 513.11 (544.45) | 46.45 (44.60) | 108.19 (141.73) | 21.48 (49.84) | 24.23 | 24.23 |
| Castle-P19    | 791.83 (762.84) | / | 1903.77 (2547.66) | 1542.87 (1541.42) | 513.11 (544.45) | 845.17 (828.06) | 25.23 (41.88) | 23.64 | 24.23 | 24.23 |
| Castle-P30    | 789.70 (827.95) | / | 307.94 (316.34) | 379.79 (389.14) | 231.92 (264.46) | 120.45 (121.16) | 21.48 (49.84) | 20.64 | 21.14 | 21.14 |

Results in parentheses take the rotation estimate by [13] as input, while other results use the final optimal rotation by OpenMVG. PA, PMBA, and BA are all initialized by the LiGT results in parentheses. Note that LinearSfM in Theia failed sometimes due to eigendecomposition and marked by ‘/’.

Fig. 7. Recovered camera poses and 3D scenes, and reprojection errors of representative data. King’s-College, Statue-of-Liberty, UWO and Ystad-Monastery are from the Lund dataset. (a) Recovered camera poses and 3D scenes by LiGT, LiGT-PA, LUD, and LUD-BA. Red arrows denote cameras. (b) reprojection errors for LiGT-PA and LUD-BA as a function of the number of iterations (maximum set at 100) performed during the optimization process. The 3D scenes were recovered analytically in LiGT and LiGT-PA, and by traditional triangulation in LUD. The squares on each vertical axis denote the reprojection errors of LUD and LiGT when global rotations refined by LiGT-PA are used as the input instead.
This study presents a pose-only representation for the multiple-view imaging geometry and discovers that it is linearly related to camera translation by the LiGT constraint. The proposed LiGT algorithm not only produces the global translation efficiently and accurately but, together with the PA algorithm, can further enhance the accuracy and robustness (for example, to critical camera motions and point-scattering phenomenon) of recovering the camera pose and 3D scene structure. The proposed pose-only solution is believed to considerably reduce the computational efficiency and robustness challenges currently encountered in 3D vision computation. Taking the tested data with the largest number of image points (#6: Basilica-di-SMF, in the eleven o’clock direction in Fig. 6b) as an example, BA consumes 100GB memory but LiGT and PA require only about 100MB and 5GB, respectively. In applications where global rotations can be provided accurately, nonlinear optimization processes may not be required for camera poses and the 3D scene structure. Consequently, the computational cost would be mitigated by several orders of magnitude, hopefully opening a door to future lightweight 3D visual computation on personal devices or microchips.

There are other aspects of the pose-only solution not yet comprehensively explored in this work, such as the strategy in selecting the reference and base views and in processing outliers. A basic view-selecting strategy is used in this work and could be enhanced for better performance. Two major factors affecting the current pose-only solution performance are the quality of image observations and the given global rotations that has been satisfyingly assured by RANSAC and rotation averaging, respectively, in the reported tests. We believe that a well-designed outlier-handling strategy for the pose-only solution deserves a systematically intensive study.

It is an open question as to why the serious point-scattering phenomenon exists in BA. To our opinion, it is likely due to the insensitivity of the reprojection error to the 3D feature coordinate, as well as the optimization technique over the high-dimensional parameter space of BA. The proposed LiGT and PA algorithms could also benefit simultaneous localization and mapping, and other versions of SfM (incremental, hybrid, and distributed). In this work, we have discussed the application of the proposed pose-only solution in the global SfM, but we believe that the solution...
Qi Cai received the BSc degree from the School of Geosciences and Info-Physics in 2015, and the MS degree from the School of Aeronautics and Astronautics, Central South University, in 2018. He is currently working toward the PhD degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include geodesy, computer vision, and inertial-visual fusion for navigation.

Lilian Zhang received the BS degree from the School of Mechatronics and Automation, National University of Defense Technology, China, in 2007, and the PhD degree from the Institute of Computer Science, University of Kiel, Germany, in 2013. He is currently an associate professor with the National University of Defense Technology. He has authored more than 30 articles in journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and International Journal of Computer Vision. His research interests include robotic vision and bionic navigation.

Yuanxin Wu received the BSc and PhD degrees in navigation from the Department of Automatic Control, National University of Defense Technology, in 1998 and 2005, respectively. He was a lecturer with the National University of Defense Technology, an associate professor with the Department of Geomatics Engineering, University of Calgary, Canada, as a visiting postdoctoral fellow and with Central South University as a professor. He is currently a professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. His research interests include inertial-based navigation system, state estimation, inertial-visual fusion, and wearable humanoid. He is the recipient of NSFC Award for Excellent Young Scientists (2014), Natural Science and Technology Award in University (2008, 2016) and the Elsevier’s Highly Cited Chinese Researchers (2015-2020). He is currently an associate editor for The Journal of Navigation and IEEE TRANSACTION ON AEROSPACE AND ELECTRONIC SYSTEMS, Program Committee for DGN Inertial Sensors and Systems (since 2019) and Saint Petersburg International Conference on Integrated Navigation Systems (since 2021) and the IEEE AESS distinguished lecturer from 2020 to 2022.

Wenxian Yu received the BSc, MSc, and the PhD degrees from the National University of Defense Technology, Changsha, China, in 1985, 1988, and 1993, respectively. From 1996 to 2008, he was a professor with the College of Electronic Science and Engineering, National University of Defense Technology, where he was the deputy dean with the School and assistant director with the National Key Laboratory of Automatic Target Recognition. In 2008, he joined the School of Electronics, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He was the vice dean of the School. He is currently a distinguished professor under the Yangtze River Scholar Scheme and the vice dean of the Advanced Industrial Technology Research Institute.

Dewen Hu received the BSc and MSc degrees from Xian Jiaotong University, China, in 1983 and 1986, respectively, and the PhD degree from the National University of Defense Technology, in 1999. In 1986, he was with the National University of Defense Technology. From October 1995 to October 1996, he was a visiting scholar with the University of Sheffield, U.K. In 1996, he was promoted as a professor. He has authored more than 300 articles in journals, including Brain, PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA, NeurImage, Human Brain Mapping, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MEDICAL IMAGING, and IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING. His research interests include pattern recognition and cognitive neuroscience. He is currently an action editor for Neural Networks, and an associate editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.