A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry

Zichao Zhang, Davide Scaramuzza

***Abstract*— In this tutorial, we provide principled methods to quantitatively evaluate the quality of an estimated trajectory from visual(-inertial) odometry (VO/VIO), which is the foundation of benchmarking the accuracy of different algorithms. First, we show how to determine the transformation type to use in trajectory alignment based on the specific sensing modality (i.e., monocular, stereo and visual-inertial). Second, we describe commonly used error metrics (i.e., the absolute trajectory error and the relative error) and their strengths and weaknesses. To make the methodology presented for VO/VIO applicable to other setups, we also generalize our formulation to any given sensing modality. To facilitate the reproducibility of related research, we publicly release our implementation of the methods described in this tutorial.**

Open Source Code

A trajectory evaluation toolbox that implements the methods in this tutorial is available at `https://github.com/uzh-rpg/rpg_trajectory_evaluation`.

I. Introduction

Visual(-inertial) odometry (VO/VIO) uses cameras and inertial measurement units (IMUs), which are complementary sensors, to estimate the state (position, orientation and velocity) of the robot. VO/VIO is able to provide robust state estimate for other tasks, such as control and planning, and therefore is widely used in robotic applications. The accuracy of a VO/VIO algorithm is quantified by evaluating the estimated trajectory (i.e., the time history of the state) with respect to the groundtruth, which is necessary to understanding and benchmarking different algorithms.

Quantitatively comparing the estimated trajectory with the groundtruth, however, is not an easy task. There are two major difficulties. First, the estimated trajectory and the groundtruth are usually expressed in different reference frames, and, therefore, cannot be compared directly. Second, a trajectory consists of the states at many different times and, therefore, is high-dimensional data. Thus, how to summarize the information of the whole trajectory into concise accuracy metrics is not trivial. To address the first problem, the estimated trajectory requires to be properly transformed into the same reference frame as the groundtruth, which is often called *trajectory alignment*. To address the second problem, meaningful error metrics need to be used and their properties well understood.

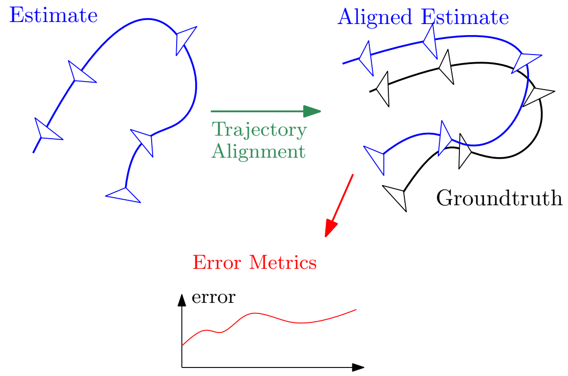


Fig. 1: The process of quantitative trajectory evaluation. First, the estimated trajectory (blue) needs to be aligned with the groundtruth (black), Then, the trajectory estimation error can be calculated from the aligned estimate and the groundtruth using certain error metrics.

To tackle the above difficulties, this tutorial provides principled methods for trajectory alignment with the focus on VO/VIO and discusses different error metrics, as illustrated in Fig. 1. We first detail the trajectory alignment methods for different visual-inertial systems (monocular, stereo and visual-inertial) and discuss the strengths and weaknesses of commonly used error metrics. We then further formulate the trajectory estimation and quantitative evaluation problem in a sensor-agnostic manner, from which we can generalize the methods presented in this tutorial to trajectory evaluation for other sensing modalities. Note that in this tutorial, we assume that the temporal correspondence of the estimate and the groundtruth has already been established.

A. Related Work

Most existing quantitative trajectory evaluation approaches were introduced together with a specific algorithm or a dataset. Sturm *et al.* [1] provided a benchmark for RBG-D simultaneous localization and mapping (SLAM) systems, and proposed to use both the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE). ATE is also widely used to evaluate visual odometry/SLAM algorithms, for example, in [2], [3], [4]. Compared with ATE, relative error, as analyzed in Burgard *et al.* [5] and Kümmerle *et al.* [6], is less sensitive to the specific time the estimation error occurs. Geiger *et al.* [7] further extended the relative error as a function of sub-trajectory length and velocity to provide more informative results.

Despite the rich literature in this field, there is very little work dedicated to the exact problem of quantitative trajectory evaluation for VO/VIO, which leaves many open issues. It is not clear, for example, to what extent the current approaches

The authors are with the Robotics and Perception Group, Dep. of Informatics, University of Zürich , and Dep. of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland—`http://rpg.ifi.uzh.ch`.
This research was supported by the National Centre of Competence in Research (NCCR) Robotics, through the Swiss National Science Foundation, the SNSF-ERC Starting Grant, and the DARPA FLA program.

are applicable: is the method for one sensing modality also suitable for another (e.g., can the same evaluation method be used for both VO and VIO)? More importantly, quantitatively evaluating an estimated trajectory involves many details, which are often described vaguely in the literature but have a big impact on the final result. This severely hinders the reproducibility of related research.

B. Contributions and Outline

The contributions of this tutorial are:

- We derive and describe in details the methods to evaluate an estimated trajectory from VO/VIO, including trajectory alignment (based on the specific sensing modality) and commonly used error metrics.
- We provide a general formulation for quantitative trajectory evaluation, which can be used to generalize the presented methods to other setups.
- We release our implementation of the evaluation methods to the public.

The rest of the tutorial is structured as follows. The formulation of visual(-inertial) odometry as a least squares problem is introduced in Section II. The ambiguity of visual-inertial systems and the trajectory alignment method, which is tightly related to the ambiguity, are detailed in Section III. Commonly used error metrics (absolute and relative errors) are then described in Section IV. In Section V, the presented trajectory evaluation methods are generalized to other setups than VO/VIO. Finally, example VIO evaluation on real data is demonstrated in Section VI.

II. Visual(-inertial) Odometry Formulation

In this section, we first define the states and the noise-free measurement model for a visual-inertial system and then formulate VO/VIO as a least squares problem.

A. States and Measurement Models

States: For a general visual-inertial system, the variables of interest (called *state*) at t_i is

$$\mathbf{x}_i = \{\mathbf{p}_i,\ \mathtt{R}_i,\ \mathbf{v}_i,\ \mathbf{b}_i^a,\ \mathbf{b}_i^g\}, \tag{1}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position of the system, $\mathtt{R}_i \in \mathrm{SO}(3)$ the rotation matrix, $\mathbf{v}_i \in \mathbb{R}^3$ the velocity, and $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ the gyroscope and accelerometer biases. $\mathbf{x}_i$ is expressed in the world frame, except that the biases in the body frame (the IMU frame is assumed to be the same as the body frame for simplicity). It is also common to maintain a map of 3D points (landmarks) as auxiliary states $L = \{\mathbf{l}_j\}_{j=0}^{J}$.

A trajectory can be parameterized either discretely or using continuous-time representations (e.g., [8]), and the former is dominant in VO/VIO. When a discrete parameterization is used, a trajectory can be represented using the states at a set of discrete times $t_s = \{t_i\}_{i=0}^{N-1}$, namely $\mathbf{X} = \{\mathbf{x}_i\}_{i=0}^{N-1}$.

Measurement Models: The measurements of a visual-inertial system come from the cameras and the IMUs. The camera project 3D points to 2D points on the image plane. The pixel coordinates of the tracked features $\tilde{\mathbf{u}}_{ij}$ are usually used as the measurements, and the noise-free measurement model is

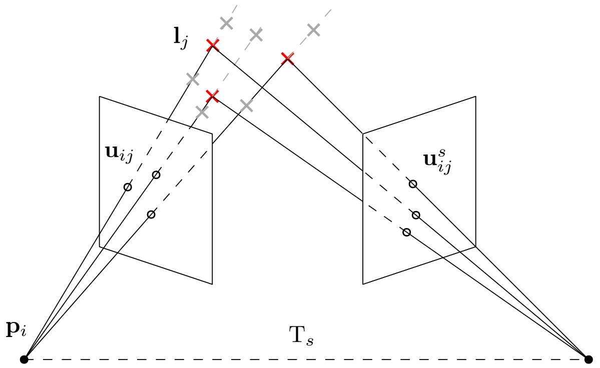


Fig. 2: Camera measurement model and scale ambiguity for a single camera. The camera projects 3D points (red crosses) to 2D points (black circles) on the image plane. For a single camera, 3D points that are in the same direction but at different distances (gray crosses) are projected to the same 2D point, which leads to the scale ambiguity in (9). When a second camera with a constant transformation T_s relative to the first one is added, the scale ambiguity is eliminated.

$$\mathbf{u}_{ij} = \mathrm{proj}(\mathtt{R}_i^\top \mathbf{l}_j - \mathtt{R}_i^\top \mathbf{p}_i), \tag{2}$$

where $\mathrm{proj}(\cdot)$ projects a 3D point in the camera frame to the pixel coordinates. In a stereo configuration, for the same 3D landmark, we also have another measurement $\tilde{\mathbf{u}}_{ij}^s$ with the noise-free measurement model

$$\mathbf{u}_{ij}^s = \mathrm{proj}(\mathtt{R}_i^\top \mathbf{l}_j - \mathtt{R}_i^\top \mathbf{p}_i - \mathbf{t}_{bs}), \tag{3}$$

where $\mathbf{t}_{bs}$ is the baseline between the stereo pair. Note that we made a few simplifications in the above formulations: the camera frame in (2) is assumed to be the same as the body frame, and the stereo cameras in (3) is assumed to be only different by a translation. For a more general setup, it can be shown that the conclusions in this section still hold. The camera measurement model is illustrated in Fig. 2.

The IMU outputs the angular velocity $\tilde{\boldsymbol{\omega}}_i$ and the specific force (acceleration together with gravity) $\tilde{\mathbf{a}}_i$ in the body frame. The measurement model is

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_i^{\mathrm{b}} + \mathbf{b}_i^g, \quad \mathbf{a}_i = \mathtt{R}_i^\top(\mathbf{a}_i^{\mathrm{w}} - \mathbf{g}) + \mathbf{b}_i^a, \tag{4}$$

where $\boldsymbol{\omega}_i^{\mathrm{b}}$ is the angular velocity in the body frame, $\mathbf{a}_i^{\mathrm{w}}$ the acceleration in the world frame, $\mathbf{g}$ the gravity vector in the world frame. The IMU measurement model (4) is illustrated in Fig. 3. The outputs of the gyroscope and the accelerometer (4) are usually at a high frequency and do not directly relate to our states (1). Therefore, a common practice in (keyframe-based) VIO algorithms is to use the integration of (4). In this paper, we use the preintegrated IMU measurements proposed in [9], [10]. Roughly speaking, we integrate the raw IMU measurements to get the relative rotation $\Delta\tilde{\mathtt{R}}_{ik}$, velocity $\Delta\tilde{\mathbf{v}}_{ik}$ and position $\Delta\tilde{\mathbf{p}}_{ik}$ between two states $\mathbf{x}_i$ and $\mathbf{x}_k$, and the integration is formulated to be independent of the states (except for the biases) so that re-integration is not needed when the states change (e.g., during optimization iterations). The corresponding measurement model is

$$\begin{aligned}
\Delta\mathtt{R}_{ik} &= \mathtt{R}_i^\top \mathtt{R}_k, \\
\Delta\mathbf{v}_{ik} &= \mathtt{R}_i^\top(\mathbf{v}_k - \mathbf{v}_i - \mathbf{g}\Delta t_{ik}), \\
\Delta\mathbf{p}_{ik} &= \mathtt{R}_i^\top(\mathbf{p}_k - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ik} - \frac{1}{2}\mathbf{g}\Delta t_{ik}^2),
\end{aligned} \tag{5}$$

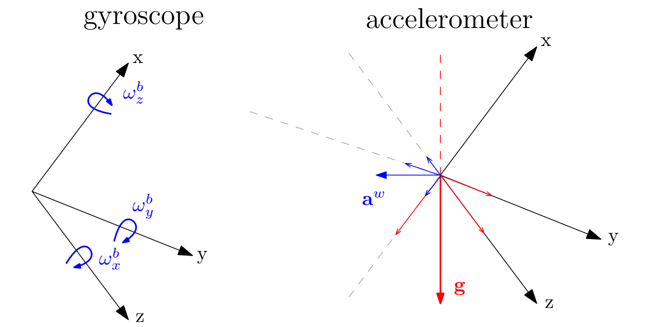


Fig. 3: IMU measurement model (4). The biases are not visualized. In the illustration of the accelerometer, if the body frame (black) is rotated around the gravity direction (red), the gravity components on the axes of the body frame remain unchanged (invariant). The invariance does not hold for rotations around any other axis.

where $\Delta t_{ik} = t_k - t_i$.

B. VO/VIO as a Least Squares Problem

By collecting the visual measurements $\tilde{\mathbf{z}}_\text{V}$ (pixel coordinates of the observed landmarks) and inertial measurements $\tilde{\mathbf{z}}_\text{I}$ (preintegrated IMU measurements e.g., [10]), VO/VIO can be formulated as a nonlinear least squares (NLLS) problem

$$\hat{\mathbf{X}}^* = \arg\min_{\mathbf{X}} J(\mathbf{X}), \tag{6}$$

where

$$J(\mathbf{X}) = \arg\min_{\mathbf{X}} \|\mathbf{f}_\text{V}(\mathbf{X}) \boxminus \tilde{\mathbf{z}}_\text{V}\|^2_{\Sigma_\text{V}} + \|\mathbf{f}_\text{I}(\mathbf{X}) \boxminus \tilde{\mathbf{z}}_\text{I}\|^2_{\Sigma_\text{I}} \tag{7}$$

where $\mathbf{f}_\text{V}(\cdot)$ and $\mathbf{f}_\text{I}(\cdot)$ denote the aforementioned noise-free visual and inertial measurement models respectively, Σ is the measurement covariance and $\|\mathbf{r}\|^2_\Sigma \triangleq \mathbf{r}^\top \Sigma^{-1} \mathbf{r}$ is the squared Mahalanobis distance [1]. In words, (6) aims to find the $\mathbf{X}$ that minimizes the sum of covariance weighted visual and inertial residuals. Note that $\boxminus$ is used because the inertial residual involves rotation. For the complete formulation of the residuals, we refer the reader to [10].

Next, we will show the inherent ambiguity of the NLLS problem (6) and how the trajectory alignment should be performed accordingly.

III. Visual(-inertial) Ambiguity and Trajectory Alignment

In this section, we first discuss the ambiguities in different visual(-inertial) setups and the complication of quantitative trajectory evaluation due to the ambiguities. We then show how to perform trajectory alignment for specific visual(-inertial) setups.

A. Ambiguities and Equivalent Parameters

(6) has infinite solutions that have the same minimum cost. The reason is that the predicted measurements $\mathbf{f}(\mathbf{X})$ are invariant to certain transformations $g(\cdot)$ of the parameter, namely $\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{X}')$ with $\mathbf{X}' = g(\mathbf{X})$. Since the measurements $\tilde{\mathbf{z}}$ are constant, the cost function (7) is also invariant to such transformations. Therefore, the NLLS problem (6) has certain ambiguities related to $g(\cdot)$, and parameters that are different by such transformations are equivalent. Note that in practice, a unique solution can be obtained by enforcing additional constraints [11].

Obviously, the transformations $g(\cdot)$ depend on the specific sensors used. To see this, we now derive the transformations that will not change the predicted measurements (2), (3) and (5). Consider a similarity transformation parameterized by $\mathtt{S} = \{s, \mathtt{R}, \mathbf{t}\}$ as a starting point, where s is a scalar, $\mathtt{R} \in \text{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$. $\mathtt{S}$ transforms the state $\mathbf{x}_i$ and $\mathbf{l}_j$ as

$$\mathbf{p}'_i = s\mathtt{R}\mathbf{p}_i + \mathbf{t},\ \mathtt{R}'_i = \mathtt{R}\mathtt{R}_i,\ \mathbf{v}'_i = s\mathtt{R}\mathbf{v}_i,\ \mathbf{l}'_j = s\mathtt{R}\mathbf{l}_j + \mathbf{t}, \tag{8}$$

and the biases are expressed in the body frame and, thus are not changed by $\mathtt{S}$.

Substituting (8) into the monocular measurement model (2), and it is obvious that

$$\mathbf{u}'_{ij} = \text{proj}(s\mathtt{R}_i^\top \mathbf{l}_j - s\mathtt{R}_i^\top \mathbf{p}_i) = \mathbf{u}_{ij} \tag{9}$$

for any $\mathtt{S}$. For a stereo setup (3), the predicted measurement using the transformed states is

$$\mathbf{u}^{s'}_{ij} = \text{proj}(s\mathtt{R}_i^\top \mathbf{l}_j - s\mathtt{R}_i^\top \mathbf{p}_i - \mathbf{t}_{bs}), \tag{10}$$

and $\mathbf{u}^{s'}_{ij} = \mathbf{u}^{s}_{ij}$ holds only when $s = 1$, and $\mathtt{S}$ becomes a rigid body transformation. The difference of a monocular and a stereo setup is illustrated in Fig. 2.

From the inertial measurement model (5), we have

$$\begin{aligned} \Delta\mathtt{R}'_{ik} &= \mathtt{R}_i^\top \mathtt{R}_k, \\ \Delta\mathbf{v}'_{ik} &= \mathtt{R}_i^\top(s\mathbf{v}_k - s\mathbf{v}_i - \mathtt{R}^\top \mathbf{g}\Delta t_{ik}), \\ \Delta\mathbf{p}'_{ik} &= \mathtt{R}_i^\top(s\mathbf{p}_k - s\mathbf{p}_i - s\mathbf{v}_i\Delta t_{ik} - s\mathtt{R}^\top \frac{1}{2}\mathbf{g}\Delta t_{ik}^2). \end{aligned} \tag{11}$$

Comparing (11) with (5), we can see that the predicted measurements remain unchanged only when $s = 1$ and $\mathtt{R}^\top \mathbf{g} = \mathbf{g}$, which means $\mathtt{R}$ can only be a rotation around z-axis and is parameterized by only one parameter θ:

$$\mathtt{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

This yaw-only rigid body transformation (one DoF rotation plus a translation) corresponds to the four unobservable DoFs for visual-inertial systems [12]. Note that although the above derivation is based on the preintegration measurement model (5), the conclusion is generally applicable for inertial sensors. Intuitively, as illustrated in Fig. 3, the gyroscope and the accelerometer measure the angular velocity and acceleration in the body frame, which are not affected by rigid body transformations. However, the accelerometer additionally measures the gravity, whose projections on the axes of the body frame only remain unchanged when the rotation is around the gravity (i.e., in the form of (12)).

To summarize, for a monocular setup, parameters that are different by a similarity transformation are equivalent. Such transformations for a stereo setup and inertial sensors are rigid body transformations and 4 DoF yaw-only rigid body transformations (i.e., a rotation around the gravity plus a translation) respectively.

[1] Strictly speaking, directly solving (6) and (7) results in a batch optimization approach. Other methods such as filters and sliding window estimators aim to solve the same problem but in a recursive manner.

B. Trajectory Evaluation with Ambiguities

The aforementioned ambiguities complicate the trajectory evaluation process: we cannot directly take the difference (e.g., Euclidean distance of the positions) between an estimate $\hat{\mathbf{X}}$ and the groundtruth $\mathbf{X}_{\text{gt}}$ as the estimation error. To see this, consider the subspaces (in the parameter space) of the equivalent parameters of $\hat{\mathbf{X}}$ and $\mathbf{X}_{\text{gt}}$, denoted as E_{est} and E_{gt} respectively, each of which contains an infinite number of equivalent parameters. For arbitrary $\hat{\mathbf{X}}_a, \hat{\mathbf{X}}_b \in E_{\text{est}}$, the estimation error computed from $\hat{\mathbf{X}}_a$ and $\mathbf{X}_{\text{gt}}$ (or any element in E_{gt}) should be exactly the same as the error with $\hat{\mathbf{X}}_b$ due to the equivalence. This is obviously not the case if we use the difference as an error metric directly.

Therefore, instead of the difference between the estimate and the groundtruth, it is the "distance" between the two corresponding equivalent parameter subspaces that should be used to quantify the estimation error. A common practice is to first find an *equivalent* estimate $\hat{\mathbf{X}}' \in E_{\text{est}}$ that is, by some metric, closest to the groundtruth $\mathbf{X}_{\text{gt}}$ and then calculate the difference from $\hat{\mathbf{X}}'$ and $\mathbf{X}_{\text{gt}}$ (see Section IV). The process of finding $\hat{\mathbf{X}}'$ is referred to as *trajectory alignment*, which we will see next for different sensor combinations.

C. Trajectory Alignment in Visual(-inertial) Systems

To find the equivalent estimate $\hat{\mathbf{X}}'$, we essentially need to find a transformation $g'(\cdot)$, which can be of different types as described in Section III-A, and then calculate $\hat{\mathbf{X}}' = g'(\mathbf{X})$. For both similarity and rigid body transformations, the method proposed in Umeyama *et al.* [13] has become the de-facto standard. In this section, we first present Umeyama's method and then show how it can be adapted to calculate the 4 DoF transformation for visual-inertial systems.

One remaining open choice is which states should be used to calculate the transformation. While there is no "gold standard", two common ways are usually used in practice: 1) using all the estimated states; 2) using only the first one or several initial states. The former tends to give a lower error if later an error metric for the whole trajectory (e.g., ATE) is used, and the latter gives an intuitive error distribution that the estimation error increases over time. We will see the examples about this point on real data in Section VI-B. In terms of computing the alignment transformation, Umeyama's method is only suitable for calculating the transformation using multiple estimated states, and, therefore, we will in addition show how to calculate rigid body and 4 DoF transformations from the first state, which will also be used for calculating the relative error metric in Section IV.

1) Alignment Using Multiple States: As discussed in Salas *et al.* [14], it is usually sufficient to calculate the trajectory alignment transformation using only the translational components of the estimation and the groundtruth. To put it formally, given the estimated positions $\{\hat{\mathbf{p}}_i\}_{i=0}^{N-1}$ and the groundtruth positions $\{\mathbf{p}_i\}_{i=0}^{N-1}$, we want to find a similarity transformation $\mathtt{S}' = \{s', \mathtt{R}', \mathbf{t}'\}$ that satisfies:

$$\mathtt{S}' = \underset{\mathtt{S}=\{s,\mathtt{R},\mathbf{t}\}}{\arg\min} \sum_{i=0}^{N-1} \|\mathbf{p}_i - s\mathtt{R}\hat{\mathbf{p}}_i - \mathbf{t}\|^2 \tag{13}$$

Algorithm 1: Closed-form solution to (13)

```
    Data: estimation $\{\hat{\mathbf{p}}_i\}_{i=0}^{N-1}$, groundtruth $\{\mathbf{p}_i\}_{i=0}^{N-1}$
    Result: $s, \mathtt{R}, \mathbf{t}$ that minimize $\sum_{i=0}^{N-1} \|\mathbf{p}_i - s\mathtt{R}\hat{\mathbf{p}}_i - \mathbf{t}\|^2$
 1  Calculate: $\boldsymbol{\mu}_{\mathbf{p}} = \frac{1}{N}\sum_{i=0}^{N-1} \mathbf{p}_i \quad \boldsymbol{\mu}_{\hat{\mathbf{p}}} = \frac{1}{N}\sum_{i=0}^{N-1} \hat{\mathbf{p}}_i$
      $\boldsymbol{\sigma}_{\mathbf{p}}^2 = \frac{1}{N}\sum_{i=0}^{N-1} \|\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}}\|^2 \quad \boldsymbol{\sigma}_{\hat{\mathbf{p}}}^2 = \frac{1}{N}\sum_{i=0}^{N-1} \|\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}}\|^2$
      $\Sigma = \frac{1}{N}\sum_{i=0}^{N-1} (\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}})(\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}})^\top$
 2  Singular value decomposition: $\Sigma = UDV^\top$
 3  if $det(U)det(V) < 0$ then
 4      $W = \mathrm{diag}(1, 1, -1)$
 5  else
 6      $W = \mathrm{I}_{3\times 3}$
 7  end
 8  $\mathtt{R} = UWV^\top$
 9  $s = \frac{1}{\boldsymbol{\sigma}_{\hat{\mathbf{p}}}^2}\mathrm{trace}(DW)$ or $s = 1$ if the scale is known
10  $\mathbf{t} = \boldsymbol{\mu}_{\mathbf{p}} - s\mathtt{R}\boldsymbol{\mu}_{\hat{\mathbf{p}}}$
```

To solve the least squares problem (13), the method in Umeyama *et al.* [13] is often used, as summarized in Alg. 1. Note that if the scale is known (stereo and inertial setup in Section III-A), we directly set $s = 1$ in line 9 of Alg. 1 After calculating the transformation $\mathtt{S}'$, the aligned estimation is then:

$$\hat{\mathbf{p}}_i' = s'\mathtt{R}'\hat{\mathbf{p}}_i + \mathbf{t}', \ \hat{\mathtt{R}}_i' = \mathtt{R}'\hat{\mathtt{R}}_i, \ \hat{\mathbf{v}}_i' = s'\mathtt{R}'\hat{\mathbf{v}}_i \tag{14}$$

If a yaw-only rigid body transformation is desired, we need to adapt the rotation calculation in Umeyama's method. As proved in [13], the rotation calculated in line 8 of Alg. 1 is the closed-form solution of

$$\mathtt{R}' = \underset{\mathtt{R}\in\mathrm{SO}(3)}{\arg\min} \|\mathtt{P} - \mathtt{R}\hat{\mathtt{P}}\|_F^2, \tag{15}$$

where $\mathtt{P} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{N-1}], \hat{\mathtt{P}} = [\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{N-1}], \mathbf{r}_i = \mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}}, \hat{\mathbf{r}}_i = \hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}}$, and $\|\cdot\|_F$ is the Frobenius norm. The cost in (15) can be further written as

$$\|\mathtt{P} - \mathtt{R}\hat{\mathtt{P}}\|_F^2 \quad = \quad \mathrm{trace}(\mathtt{P}\mathtt{P}^\top + \hat{\mathtt{P}}\hat{\mathtt{P}}^\top - 2\mathtt{R}\hat{\mathtt{P}}\mathtt{P}^\top), \tag{16}$$

and therefore (15) is equivalent to

$$\mathtt{R}' = \underset{\mathtt{R}\in\mathrm{SO}(3)}{\arg\max}\ \mathrm{trace}(\mathtt{R}\hat{\mathtt{P}}\mathtt{P}^\top). \tag{17}$$

If the rotation is of the form (12), we only need to find the following maximum with respect to θ:

$$\theta' = \underset{\theta}{\arg\max}\ (p_{12} - p_{21})\sin\theta + (p_{11} + p_{22})\cos\theta \tag{18}$$

where p_{ij} is the coefficient of $\hat{\mathtt{P}}\mathtt{P}^\top$. With the solution θ' to (18), we can calculate the desired rotation $\mathtt{R}_z'$ using (12) and the translation with line 10 in Alg. 1 (with $s = 1$). The aligned estimation is calculated the same as (14).

It is worth noting that, in this section, the alignment is based on a least squares solution, which is valid only when all the states are of the same uncertainty. If we

TABLE I: Transformations in trajectory alignment for different visual and inertial configurations.

Configuration	Monocular	Stereo	Inertial(+visual)
Type	Similarity	Rigid body	Yaw-only rigid body
Align-Multi	Alg. 1	Alg. 1	Alg. 1 with rotation (18)
Align-Single	×*	(19)	(19) with rotation (20)

* Scale cannot be estimated from a single state.

have the knowledge about the quality of the state estimate, for example, covariance from VO/VIO, more sophisticated methods can be used to account for this (e.g., optimization as in [14]).

2) Alignment Using A Single State: It is possible to calculate a rigid body transformation or a yaw-only transformation with only the first state. Calculating a rigid body transformation is trivially

$$\mathtt{R}' = \mathtt{R}_0\hat{\mathtt{R}}_0^\top, \quad \mathbf{t}' = \mathbf{p}_0 - \mathtt{R}'\hat{\mathbf{p}}_0. \tag{19}$$

Similar to the previous case, computing a yaw-only transformation needs a different treatment. Specifically, the rotation $\hat{\mathtt{R}}_0' = \mathtt{R}_z'\hat{\mathtt{R}}_0$ should be as close to $\mathtt{R}_0$ as possible:

$$\begin{aligned} \mathtt{R}_z' &= \arg\min_{\mathtt{R}_z} \|\mathtt{R}_0 - \mathtt{R}_z\hat{\mathtt{R}}_0\|_F^2 \\ \Rightarrow \quad \mathtt{R}_z' &= \arg\max_{\mathtt{R}_z} \ \mathrm{trace}(\mathtt{R}_z\hat{\mathtt{R}}_0\mathtt{R}_0^\top), \end{aligned} \tag{20}$$

which is of the same form as (17) and can be solved similarly. Once we have $\mathtt{R}_z'$, the translational component $\mathbf{t}'$ is calculated the same as (19).

D. Summary

To summarize, different combinations of visual and inertial sensors result in different ambiguities in VO/VIO. Due to the ambiguities, certain types of transformations should be used to align the estimation with the groundtruth before calculating the estimation error. For various combinations of visual and inertial sensors, we summarize the types of trajectory alignment transformations and the methods to calculate them in Table. I. Using the aligned trajectory estimate, we can now calculate different error metrics to quantify the accuracy of VO/VIO.

IV. Trajectory Error Metrics

To calculate the estimation error from the groundtruth $\mathbf{X}_{\text{gt}}$ and the aligned estimation $\hat{\mathbf{X}}'$, two commonly used error metrics are the absolute trajectory error (ATE) and the relative error (RE). In this section, we will describe them in details and discuss their advantages and disadvantages.

A. Absolute Trajectory Error

For a single state, the error between $\hat{\mathbf{x}}_i'$ and the groundtruth $\mathbf{x}_i$ can be parameterized as

$$\Delta\mathbf{x}_i = \{\Delta\mathtt{R}_i, \Delta\mathbf{p}_i, \Delta\mathbf{v}_i\} \tag{21}$$

and satisfies

$$\mathtt{R}_i = \Delta\mathtt{R}_i\hat{\mathtt{R}}_i', \ \mathbf{p}_i = \Delta\mathtt{R}_i\hat{\mathbf{p}}_i' + \Delta\mathbf{p}_i, \ \mathbf{v}_i = \Delta\mathtt{R}_i\hat{\mathbf{v}}_i' + \Delta\mathbf{v}_i \tag{22}$$

Note that the parameterization of the error (21) and (22) is not unique. For example, $\Delta\mathtt{R}_i$ can also appear on the right side of $\hat{\mathtt{R}}_i'$ in (22). While there is no standard for error parameterization, one must be consistent during the trajectory evaluation. In addition, since the biases are always expressed in the body frame, the biases error is trivially the Euclidean distance of the estimate and the groundtruth.

With (22), we can easily calculate the error $\Delta\mathbf{x}_i$

$$\begin{aligned} \Delta\mathtt{R}_i &= \mathtt{R}_i(\hat{\mathtt{R}}_i')^\top, \\ \Delta\mathbf{p}_i &= \mathbf{p}_i - \Delta\mathtt{R}_i\hat{\mathbf{p}}_i', \\ \Delta\mathbf{v}_i &= \mathbf{v}_i - \Delta\mathtt{R}_i\hat{\mathbf{v}}_i'. \end{aligned} \tag{23}$$

To quantify the quality of the whole trajectory, the root mean square error (RMSE) is usually used

$$\begin{aligned} \text{ATE}_{\text{rot}} &= (\frac{1}{N}\sum_{i=0}^{N-1}\|\angle(\Delta\mathtt{R}_i)\|^2)^{\frac{1}{2}}, \\ \text{ATE}_{\text{pos}} &= (\frac{1}{N}\sum_{i=0}^{N-1}\|\Delta\mathbf{p}_i\|^2)^{\frac{1}{2}}, \end{aligned} \tag{24}$$

where $\angle(\cdot)$ means converting the rotation matrix to angle-axis representation and using the rotation angle as the error. Alternatively, one can also convert $\Delta\mathtt{R}_i$ to other representations (e.g., Euler angles) and get the corresponding rotation errors. The velocity error is defined similarly and omitted here. The calculation of ATE is illustrated in Fig. 4a.

The advantage of ATE is that it gives a single number metric for the position/rotation/velocity estimation, which is easy to compare. However, as recognized by several researchers [5], [6], [7], ATE is sensitive to the time when the error occurs. For example, a rotation estimation error tends to give a larger ATE when it happens at the beginning of the trajectory than the situation when it occurs at the end. Therefore, in addition to ATE, the relative error is also widely used to provide more informative evaluation.

B. Relative Error

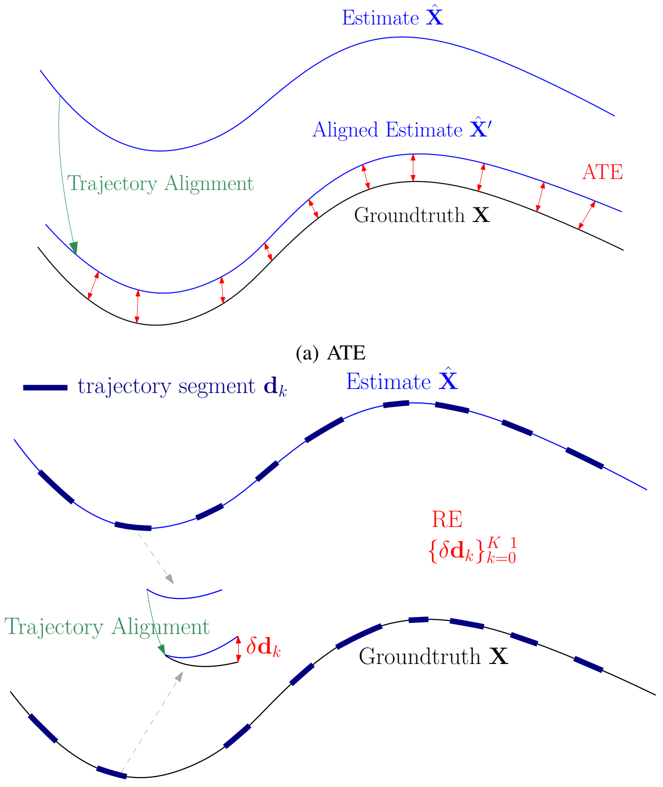
The basic idea of relative error is that, since VO/VIO systems do not have a global reference (global position and yaw), the estimation quality can be evaluated by measuring the relative relations between the states at different times.

To put it formally, first a set of K pairs of states is selected by some criteria (e.g., distance along the trajectory) from $\hat{\mathbf{X}}$:

$$\mathfrak{F} = \{\mathbf{d}_k\}_{k=0}^{K-1}, \quad \mathbf{d}_k = \{\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_e\}, \tag{25}$$

where $e > s$, and each pair defines a sub-trajectory. For each $\mathbf{d}_k$, a relative error $\delta\mathbf{d}_k$ is calculated in a similar way as the absolute error. Specifically, an alignment transformation, depending on the sensor configuration as in Table. I, is computed from the first state $\hat{\mathbf{x}}_s$ and the corresponding groundtruth $\mathbf{x}_s$, and the aligned second state $\hat{\mathbf{x}}_e'$ computed using (14). Then the error $\delta\mathbf{d}_k$ for the state pair $\mathbf{d}_k$ is

$$\begin{aligned} \delta\boldsymbol{\phi}_k &= \angle\,\delta\mathtt{R}_k = \angle\,\mathtt{R}_e(\hat{\mathtt{R}}_e')^\top, \\ \delta\mathbf{p}_k &= \|\mathbf{p}_e - \delta\mathtt{R}_k\hat{\mathbf{p}}_e'\|_2, \\ \delta\mathbf{v}_k &= \|\mathbf{v}_e - \delta\mathtt{R}_k\hat{\mathbf{v}}_e'\|_2, \end{aligned} \tag{26}$$


Estimate $\hat{\mathbf{X}}$
Aligned Estimate $\hat{\mathbf{X}}'$
Trajectory Alignment
ATE
Groundtruth $\mathbf{X}$
trajectory segment $\mathbf{d}_k$
Estimate $\hat{\mathbf{X}}$
RE
$\{\delta \mathbf{d}_k\}_{k=0}^{K\ 1}$
Trajectory Alignment
$\delta \mathbf{d}_k$
Groundtruth $\mathbf{X}$

(a) ATE

(b) RE

Fig. 4: Illustrations of absolute trajectory error and relative error. The error after alignment is exaggerated for visualization. For relative error, the trajectory segments should be all possible pairs of states that satisfy certain criteria, and they are un-overlapped in (b) for the ease of visualization.

which are all scalars. Collecting the error (26) for all the pairs of states (sub-trajectories) in $\mathfrak{F}$ gives

$$\begin{aligned} \text{RE}_{\text{rot}} &= \{\delta \boldsymbol{\phi}_k\}_{k=0}^{K-1}, \\ \text{RE}_{\text{pos}} &= \{\delta \mathbf{p}_k\}_{k=0}^{K-1}, \\ \text{RE}_{\text{vel}} &= \{\delta \mathbf{v}_k\}_{k=0}^{K-1}. \end{aligned} \tag{27}$$

The calculation of RE is illustrated in Fig. 4b.

Since the relative error (27) does not generate a single number but a collection of errors for all the sub-trajectories that satisfy certain criteria, statistics such as the median, average and percentiles can be calculated, which gives more information than ATE. Another advantage is that by selecting the states according to different criteria, RE can have different meanings. For example, a common practice is to select pairs of states that are spaced by a certain distance along the trajectory. The RE from the states pairs that are spatially close reflects the local consistency, while the error for a larger distance reflects more the long-term accuracy. The disadvantage of RE is that it is relatively complicated to calculate, and it is less obvious to rank the estimation quality than using a single number metric as ATE.

C. Discussion and Summary

As discussed above, both ATE and the RE have their own advantages and disadvantages. It is probably not possible to say that a metric should be preferred in all situations over the other one. However, as pointed by [1], the two error metrics are actually highly correlated. In practice, providing both error metrics, if possible, will give a better understanding of the actual estimation quality from different aspects. We summarize the computation and properties of ATE and RE in Table. II.

TABLE II: Comparison of absolute trajectory error and relative error.

	absolute trajectory error	relative error
Compute	**1**. Align the estimated trajectory. **2**. Calculate the RMSE using the aligned estimation and the groundtruth (24)	**1**. Select all sub-trajectory of length *d*. **2**. Align each sub-trajectory using the first state. **3**. Calculate the error of the end state of each sub-trajectory (26). **4**. Collect the errors for all the sub-trajectories (27). **5**. For different lengths *d*, repeat step 1-4.
Pros	• Single number metric, easy for comparison.	• Informative statistics can be computed from the errors of all sub-trajectories. • By changing the length *d*, the relative error can reflect both short and long term accuracy.
Cons	• Sensitive to the time when the estimation error occurs.	• Relatively complicated to compute. • Less straightforward for ranking the estimation accuracy.

Together with the trajectory alignment described in Section III, we can quantify the accuracy of a trajectory estimate from VO/VIO. Before demonstrating the evaluation procedures on real data in Section VI, we first show that the aforementioned methods for VO/VIO can be generalized to arbitrary sensing modalities.

V. General Trajectory Evaluation Problem

A. Trajectory Estimation Problem

Similar to VO/VIO in Section II, we define the estimation problem by specifying the parametrization of the trajectory, the measurements, and the cost function to minimize.

Parameterization: Using discrete parameterization, a trajectory can be represented using the states $\mathbf{X} = \{\mathbf{x}_i\}_{i=0}^{N-1}$ at a set of discrete times $t_s = \{t_i\}_{i=0}^{N-1}$.

Measurements: The measurements are collected at t_s, denoted as $\tilde{\mathbf{M}} = \{\tilde{\mathbf{z}}_i\}_{i=0}^{N-1}$, Note that $\tilde{\mathbf{z}}_i$ can be either the raw readings from the sensors or the output of processing the raw data (e.g., keypoint coordinates (2), preintegrated IMU measurements (5)). The corresponding noise-free measurement model is denoted as $\mathbf{f}(\mathbf{x})$.

Cost: For an estimate of the system parameters $\mathbf{X}$, a commonly used cost is the sum of the squared Mahalanobis distance between the actual and the predicted measurements:

$$c(\mathbf{X}, \tilde{\mathbf{M}}) = \sum_{i=0}^{N-1} \| f(\mathbf{x}_i) - \tilde{\mathbf{z}}_i \|_{\Sigma_i}^2, \tag{28}$$

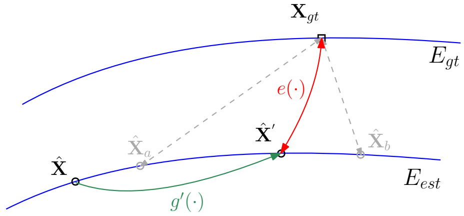


Fig. 5: Illustration of the equivalent subspaces (blue) and the trajectory evaluation process in the parameter space. Directly using the difference (dashed gray line) between an estimation and the groundtruth does not give the same estimation error for equivalent parameters. Instead, the distance between the equivalent subspaces should be used. The first step (green) is to find a unique equivalent estimation $\hat{\mathbf{X}}'$ that is closest to $\mathbf{X}_{gt}$ by a distance metric $d_g(\cdot)$. The second step (red) is to calculate the distance between $\hat{\mathbf{X}}'$ and $\mathbf{X}_{gt}$ using an error metric $e(\cdot)$.

where Σ_i is the measurement covariance. The trajectory estimation is then the process of determining a set of trajectory parameters that minimize the cost (28):

$$\hat{\mathbf{X}}^* = \arg\min_{\mathbf{X}} c(\mathbf{X}, \tilde{\mathbf{M}}). \tag{29}$$

B. Ambiguities and Equivalent Parameters

With only the sensor measurements, the estimation problem (29) usually does not have a unique solution. For example, the absolute position cannot be determined for a visual(-inertial) odometry system. To put it formally, there exist a set of transformations $G = \{g(\cdot)\}$ that satisfy

$$c(g(\mathbf{X}), \tilde{\mathbf{M}}) = c(\mathbf{X}, \tilde{\mathbf{M}}) \quad \forall \mathbf{X}, \ \forall g(\cdot) \in G, \tag{30}$$

where G is determined by the sensor combinations. In other words, for any $\mathbf{X}$, there is a subspace $E_{\mathbf{X}}$ (in the parameter space) where each element has the same cost (28) as $\mathbf{X}$.

Due to this ambiguity, we cannot directly take the difference (e.g., Euclidean distance if the states are vectors) between the estimation $\hat{\mathbf{X}}$ and the groundtruth $\mathbf{X}_{\text{gt}}$ as the estimation error, as illustrated in Fig. 5.

C. Quantitative Trajectory Evaluation

To uniquely define the estimation error of an estimate $\hat{\mathbf{X}}$, the first step is to find an equivalent estimation $\hat{\mathbf{X}}'$ that is closest to $\mathbf{X}_{gt}$ according to a certain distance metric $d_g(\cdot)$:

$$g'(\cdot) = \arg\min_{g(\cdot)\in G} d_g(g(\hat{\mathbf{X}}), \ \mathbf{X}_{gt}), \quad \hat{\mathbf{X}}' = g'(\hat{\mathbf{X}}), \tag{31}$$

which is the trajectory alignment process. Then we can quantify the difference between the estimation and the groundtruth by calculating the error between $\hat{\mathbf{X}}'$ and $\mathbf{X}_{gt}$ using a certain error metric $e(\cdot)$ as $e(\hat{\mathbf{X}}', \mathbf{X}_{gt})$. The above process in the parameter space is illustrated in Fig. 5. We denote the distance metric $d_g(\cdot)$ and error metric $e(\cdot)$ only conceptually, because there is no standard way for defining them, as described in Section III and Section IV.

Therefore, for any sensor combination with ambiguities, to calculate the estimation error, we need to follow similar procedures as VO/VIO : 1) align the estimate with the groundtruth; 2) calculate the estimation error using certain metrics. Importantly, the transformation used for trajectory alignment needs to be computed by considering the properties of the sensors used, as we already see for visual(-inertial) systems in Section III.

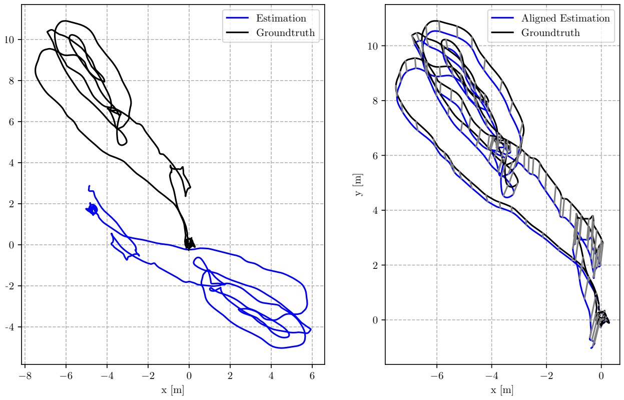


Fig. 6: Trajectory alignment for the estimate from VINS-Mono on *Machine Hall 01*. The left is the top view of the unaligned estimation and the groundtruth, and the right is the aligned trajectory. The states correspondences are shown as gray lines (every 10th is drawn for clear visualization).

VI. Example Quantitative Evaluation

To illustrate the methods described in Section III and Section IV with concrete examples, we first demonstrate the complete process of computing ATE and RE from an unaligned estimation and the groundtruth. Then we show the impact of the number of frames used for trajectory alignment, which seems to be a trivial detail but turns out rather crucial.

A. ATE and RE: a Complete Example

We ran VINS-Mono [15], which is a visual-inertial odometry algorithm, on the *Machine Hall 01* sequence from the EuRoC dataset [16] and evaluated the estimated trajectory.

As discussed above, the first step is to align the estimation with the groundtruth. We used all the states to calculate a yaw-only rigid-body transformation to align the trajectory as described in Section III. The process is illustrated in Fig. 6. We can see the "raw" estimation from VINS-Mono is in a different reference frame as the groundtruth, and therefore cannot be directly compared. We then computed the estimation error using the aligned trajectory and the groundtruth. The absolute error for each state (23) is plotted in Fig. 7, and the ATE (24) described in the caption.

The relative position and rotation errors (27) are plotted in Fig. 8. We calculated the relative errors for sub-trajectories of different lengths. It is clear from Fig. 8 that the estimation error (both translation and rotation) increases with the length of the sub-trajectories.

B. ATE: How Many Frames to Align?

As discussed in Section III, there is no standard for selecting the number of states to be used for trajectory alignment. However, it is of interest to understand how this choice affects the computed estimation error. To this end, we performed the same evaluation as the previous section, but used different states for trajectory alignment: the first Q states are used, where Q varies from 1 to the number of all the states in the trajectory.

We show the ATE of the whole trajectory for five different alignments in Table. III. We can see that the position ATE

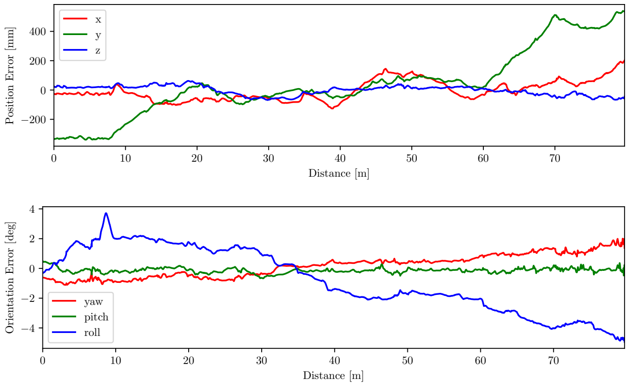


Fig. 7: Absolute position and orientation error (23) with respect to the traveled distance, computed from the aligned trajectory and the groundtruth in Fig. 6. The ATE (24) is 0.2795 m for translation and 2.4935 deg for rotation.

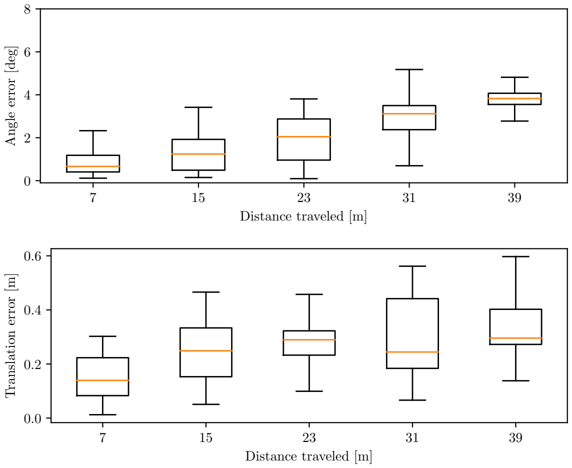


Fig. 8: Relative translation and rotation errors (27) for different sub-trajectory lengths shown as a series of boxplots. The box in the middle indicates the two quartiles of all the estimation errors, the line through the box the median, and the whiskers the upper and lower quartiles.

decreases when more states are used in the alignment, while the rotation ATE does not show a obvious tendency. Intuitively, since the trajectory alignment aims to minimize the least squares position error (13), the more states that are used, the smaller the position ATE is likely to be. The rotation components are not used in computing the alignment transformation and thus are less correlated.

Note that in Table. III, the difference of $\text{ATE}_{\text{trans}}$ between using the first state and all the states for alignment is quite large ($\sim$ 150%). Therefore, in practice, when comparing different algorithms, one needs to be consistent in which states are used for trajectory alignment across different algorithms for a fair comparison. Moreover, this information is quite crucial to reproduce quantitative accuracy evaluations for VO/VIO and should always be presented together with the evaluation results.

VII. Conclusion

In this tutorial, we presented principled approaches for quantitative trajectory evaluation for VO/VIO algorithms. We discussed the ambiguities in visual(-inertial) systems, which is the main source of the complication in trajectory evaluation. Then we detailed the quantitative evaluation methods for VO/VIO, including the trajectory alignment and error metrics. We further showed that similar approaches can be adopted for other sensing modalities that has ambiguities. To benefit the reproducibility of related research, we release our implementation of the methods in this tutorial to the public.

TABLE III: ATE using different states for trajectory alignment. When more states are used in the alignment, the translation ATE tends to be smaller.

States used	ATE_{pos} (m)	ATE_{rot} (deg)
1	0.4383	2.4919
1 - 452	0.4134	2.7427
1 - 904	0.3515	2.7902
1 - 1355	0.3180	2.8365
1 - 1807 (all)	0.2795	2.4935

References

[1] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct. 2012.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PP, no. 99, pp. 1–1, 2017.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, 2017.

[5] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós, "A comparison of SLAM algorithms based on a graph of relations," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct 2009, pp. 2089–2095.

[6] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robots*, vol. 27, no. 4, p. 387, Sep 2009.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012.

[8] T. Barfoot, C. H. Tong, and S. Sarkka, "Batch continuous-time trajectory estimation as exactly sparse gaussian process regression," in *Robotics: Science and Systems (RSS)*, 2014.

[9] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[11] Z. Zhang, G. Gallego, and D. Scaramuzza, "On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, July 2018.

[12] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Research*, vol. 30, no. 1, pp. 56–79, 2011.

[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 4, 1991.

[14] M. Salas, Y. Latif, I. D. Reid, and J. Montiel, "Trajectory alignment and evaluation in SLAM: Horns method vs alignment on the manifold." Robotics: Science and Systems Workshop: The problem of mobile sensors, 2015.

[15] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *arXiv e-prints*, Aug. 2017.

[16] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Research*, vol. 35, pp. 1157–1163, 2015.