Sampson Distance: A New Approach to Improving Visual-Inertial Odometry's Accuracy

He Zhang, Cang Ye, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a new scheme based on the Sampson distance (SD) to describe visual feature residuals for visual-inertial odometry (VIO). Unlike the epipolar-constraint-based SD for visual odometry (VO), the proposed SD is formulated based on the perspective projection constraint. We proved in theory that the proposed SD retains the good properties of those earlier SD criteria in the literature of VO and it represents a visual feature residual more accurately than the prevailing transfer distance (TD) in existing VIO methods. We formulate three distance criteria, including TD, reprojection error (RE), and SD, and compared their performances by simulation. The results show that the SD is much more accurate than the TD and it is a very accurate estimate of the gold standard criteria—RE. Based on the SD, we modified VINS-Mono by replacing its TD-based visual residuals with the SD-based residuals and study the SD's efficacy in pose estimation by experiments with several public datasets. The results reveal that the SD-based VINS-Mono has a substantial improvement over the original VINS-Mono in pose estimation accuracy. This indicates that the SD is a better distance criterion than the TD for representing visual feature residuals. The proposed SD may find its applications to broader areas in computer vision and robotics.**

I. Introduction

Over the last decade, visual-inertial navigation systems (VINS) have been widely used in virtual/augmented reality [1], autonomous navigation [2], [3], [4], 3D reconstruction [5], assistive navigation [6], etc. VINS, consisting of a camera and an inertial measurement unit (IMU), estimates its 6-DOF pose by visual-inertial odometry (VIO) that fuses the visual and inertial data to compute the pose estimation. Both sensory data are used to compute the system's egomotion by tracking the visual features over image frames and by integrating the IMU measurements over time. However, they contribute to pose estimation in a complementary fashion: the inertial data tracks the system's motion when the visual odometry (VO) malfunctions in a feature-sparse scene while the VO-estimated motion helps to reduce the IMU bias and thus lower its drift error. The net effect is that the VIO-estimated pose is more accurate and reliable than that estimated by VO.

The state-of-the-art VIO methods [7], [8], [9] tightly couple the residuals of the inertial data and the visual measurements by using a filtering or a smoothing strategy to estimate the VINS' motion state. They formulate the residual for each visual measurement as the transfer distance (TD). Given a pair of matched visual features between two images, the TD is computed on the 2nd image as the squared distance between the feature and the forward-projected feature (i.e., the perspective projection of the feature from the 1st image onto the 2nd image). This computation assumes that the location of the feature on the 1st image is error-free. The assumption does not hold in the real world due to image noise, uncalibrated optical distortion, etc. The use of TD results in an estimate of the fundamental matrix (FM) that is not optimal. To overcome this problem, a more accurate criterion—reprojection error (RE) —is introduced [10]. RE takes into account the perspective projection errors of the pair of visual features on both images (i.e., both forward-/backward-projection errors). Assuming a Gaussian noise for image points, the maximum likelihood estimate (MLE) of the FM is the one that minimizes the RE. This solution is regarded as the gold standard method [10] in the literature of computer vision. However, the minimization of RE is computationally expensive because it involves an iterative optimization process where the relevant camera poses and the associated visual features' locations must be updated simultaneously at each iteration. A more time-efficient alternative is to use the Sampson distance (SD). SD is computed based on the epipolar constraint and it can be viewed as the first-order approximation of RE [10]. In existing research, SD has been used to estimate the FM between two camera views and achieves better accuracy [15], [16], [18] than other distance criteria. From the estimated FM, the relative camera pose change can be retrieved readily—an essential step for VO [11] to estimate the camera pose. Inspired by this, we hypothesize that using SD as the distance criteria for the visual measurements' residuals in VIO will result in more accurate pose estimation. To validate the hypothesis, we formulate the residual of a visual feature by using the SD and introduce a new VIO method based on the SD. Since the SD is computed differently in our case (based on the perspective projection constraint instead of the epipolar constraint), we must prove/demonstrate that it retains its superiority as demonstrated in the existing work [16], [18]. To the best of our knowledge, this paper is the first work of its kind in the literature of VIO. Our main contributions are summarized as follows:

- We derive the formulas for computing SD based on the perspective projection constraint. This way, the SD functions can be used in the context of VIO for pose estimation.

This work was supported by the National Eye Institute of the NIH under Award R01EY026275. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agency.

The authors are with the Computer Science Department, Virginia Commonwealth University, Richmond, VA 23284, USA. (e-mail: cye@vcu.edu).

- We prove in theory TD > SD ⩾ RE, i.e., SD is a more accurate estimation for RE compared with TD.
- We compare the SD-based VIO with the TD-based VIO by simulation and by experiments with two real-world datasets, EuRoc MAV [12] and TUM VI [13], to demonstrate the superiority of SD in formulating the visual residuals.

II. Related Work

SD was originally introduced by Sampson [14] for conic fitting to scattered data, where an iterative weighted least-square method was used to estimate the parameters of a conic model. The method finds the refined parameters that reduce the overall fitting errors iteratively. The fitting error of a data point is weighted proportionally to the inverse gradient of the fitting error norm. This way, the influence of a data point with large noise is reduced and the fitting accuracy is improved. Later, SD was introduced to computer vision to model the measurement residuals of the visual feature correspondences between two camera views to compute the 2D homography, FM, or the trifocal tensor [10], [15]. In [16], SD is used to estimate the essential matrix and its accuracy is compared with that of other distance criteria, including the algebraic distance, the symmetric squared geometric distance (SED), and the squared reprojection distance. The method computes multiple essential matrix candidates by using the classic five-points method [17] and then selects the one with the smallest distance. The comparative studies in [16] reveal that the use of SD/RE results in a more accurate estimation result than other distance criteria and SD is more computationally efficient than RE. A similar study was conducted in [18], where it was proved in theory that SED (i.e., the sum of the TDs in both images) is larger than RE/SD. In addition, it was demonstrated by simulation that the values of SD are almost equal to that of RE. In other words, the use of SD as a cost function to be minimized results in a more accurate FM estimate when compared to SED. In these methods, the SD for each visual feature is formulated as the distance between the estimated visual measurement and the epipolar line, i.e., the error representing a deviation from the epipolar constraint.

Existing VIO methods [1-5], [7-9] use TD to express the residuals for the visual features and include the visual residuals in the cost function to be minimized. It is natural to hypothesize that by replacing the TD-based visual residual with an SD-based one we can improve the pose estimation accuracy. In the VIO framework, the depth of a visual feature is computed when it is first observed and tracked onto the next image frame. Its coordinates on the subsequent images may be determined via perspective projection. Therefore, the SD, in this case, is computed as the geometric distance between the actual visual measurement and the estimated visual measurement. To validate the hypothesis, we derive the new formulas for the SD as well as the visual residual and prove that the property of the SD in the earlier works can be translated to its VIO application. Based on the new SD-based visual residual, we modify a state-of-the-art VIO method—VINS-Mono [9]—by replacing its TD with SD for computing the visual residuals and then compare the modified VINS-Mono with the original method to demonstrate the benefit of using the SD in VIO.

III. Sampson Distance in VIO

Our previous work [19] reveals that VINS-Mono has the most accurate pose estimation result because: 1) Its robust initialization allows for a more accurate estimation of the initial system state; 2) Its inverse-depth parameterization of 3D visual features results in Gaussian-distributed position uncertainties for low parallax visual features [20], which have non-Gaussian position uncertainties if described by using their XYZ coordinates. Therefore, we use VINS-Mono as a benchmark and modify it by using the SD. We first introduce the principle of the nonlinear state estimation of VINS-Mono and the definition of the SD, and we then derive the residual functions for the visual-feature-pairs by using the SD.

A. State Estimator

A sliding window-based nonlinear optimization process is employed for state estimation. The full state vector within the sliding window is defined as $\boldsymbol{\chi} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M, \lambda_1, \lambda_2, \dots, \lambda_K\}$ where $\mathbf{s}_i = \{\mathbf{t}_{b_i}^w, \mathbf{v}_{b_i}^w, \mathbf{q}_{b_i}^w, \mathbf{b}_a, \mathbf{b}_g\}$ ($i \in [1, M]$) is the IMU's motion state (position, velocity, orientation, accelerometer bias, and gyroscope bias) at the time when the i^{th} keyframe is captured. Superscript w is used to denote the world coordinate system while subscripts b and c represent the IMU and the camera coordinate systems, respectively. M is the size of the sliding window ($M = 10$ in this work) and K is the total number of the visual features inside the sliding window. λ_k ($k = 1, \dots, K$) is the estimate for the inverse depth of the k^{th} visual feature at the time it is first observed. In the world coordinate system, the IMU's pose is denoted by $\boldsymbol{\xi}_{b_i}^w = \{\mathbf{t}_{b_i}^w, \mathbf{q}_{b_i}^w\}$ and the rotation matrix corresponding to quaternion $\mathbf{q}_{b_i}^w$ is denoted as $\mathbf{R}_{b_i}^w$. The state vector $\boldsymbol{\chi}$ is computed by minimizing the following cost function, representing the sum of the Mahalanobis distances for all residuals of the visual-inertial measurements:

$$\boldsymbol{\chi}^* = \underset{\boldsymbol{\chi}}{\operatorname{argmin}}\left(||^p\mathbf{r}||^2 + \sum_j ||^u\mathbf{r}_j||^2 + \sum_k \sum_{(i,j)} \rho\left(||^v\mathbf{r}_{ij}^k||^2\right)\right) \quad (1)$$

where $\rho()$ is the Cauchy loss function [21]. $^p\mathbf{r}$, $^u\mathbf{r}_j$, and $^v\mathbf{r}_{ij}^k$, represent the residuals for the prior information from marginalization, IMU preintegration (between keyframes $j-1$ and j), and the visual feature k that is first observed at keyframe i and tracked onto keyframe j, respectively. The modified VINS-Mono computes $^v\mathbf{r}_{ij}^k$ by using the SD criterion, which is described later. Readers are referred to [9] for the details on the computation of the other residuals.

B. Sampson Distance

The problem of estimating the camera's pose change between two keyframes given several matched visual-feature-pairs $\mathbf{x}_p \leftrightarrow \mathbf{x}'_p$ for $p = 1, \cdots, P$ can be fitted into a common framework with two components [10]:

- a *measurement space* $\mathbb{R}^N$ consisting of measurement vectors $\mathbf{X} = \{\mathbf{x}_p, \mathbf{x}'_p\}$
- a *model* H, representing a subset S of points in $\mathbb{R}^N$. A measurement vector $\mathbf{X}$ that lies on S is said to satisfy the

model. The subspace that satisfies the model is called a *variety* $\mathcal{V}_H$ in $\mathbb{R}^N$

Differing from the previous works [15], [16], [18], that use epipolar constraint, H is defined by using the perspective projection constraint in our work. This constraint ensures the perspective projection error is zero for a visual-feature-pair in $\mathbf{X} \in \mathcal{V}_H$ satisfying $\mathcal{C}_H(\mathbf{X}) = 0$, where $\mathcal{C}_H(\mathbf{X})$ is the cost function. Due to image noise or uncalibrated optical distortion, $\mathcal{C}_H(\mathbf{X}) \neq 0$ for a real-world scenario. The measurement residual can be modeled by finding a vector $\hat{\mathbf{X}}$, which is the closest to $\mathbf{X}$ and satisfies H, i.e. $\mathcal{C}_H(\hat{\mathbf{X}}) = 0$. For a nonlinear $\mathcal{C}_H(\mathbf{X})$, $\hat{\mathbf{X}}$ can be obtained by an iterative process. In each iteration, $\mathcal{C}_H(\mathbf{X})$ is approximated by a Taylor expansion:

$$\mathcal{C}_H(\mathbf{X} + \boldsymbol{\delta}_\mathbf{X}) \approx \mathcal{C}_H(\mathbf{X}) + \frac{\partial \mathcal{C}_H}{\partial \mathbf{X}} \boldsymbol{\delta}_\mathbf{X} \tag{2}$$

Here, $\boldsymbol{\delta}_\mathbf{X} = \hat{\mathbf{X}} - \mathbf{X}$ quantifies the measurement residual, and $\hat{\mathbf{X}}$ is the corrected measurement vector on $\mathcal{V}_H$, satisfying $\mathcal{C}_H(\hat{\mathbf{X}}) = 0$. Letting $J = \frac{\partial \mathcal{C}_H}{\partial \mathbf{X}}$ and $\boldsymbol{\epsilon} = \mathcal{C}_H(\mathbf{X}) - \mathcal{C}_H(\hat{\mathbf{X}})$, we have

$$J\boldsymbol{\delta}_\mathbf{X} = -\boldsymbol{\epsilon} \tag{3}$$

The minimization of cost $\mathcal{C}_H(\mathbf{X})$ is equivalent to finding $\boldsymbol{\delta}_\mathbf{X}$ that minimizes $||\boldsymbol{\delta}_\mathbf{X}||$ subjects to (3). The problem can be solved by using Lagrange Multipliers [10] and the solution is given by:

$$\boldsymbol{\delta}_\mathbf{X} = -J^T(JJ^T)^{-1}\boldsymbol{\epsilon} \tag{4}$$

The SD is defined as the squared norm of $\boldsymbol{\delta}_\mathbf{X}$:

$$||\boldsymbol{\delta}_\mathbf{X}||^2 = \boldsymbol{\epsilon}^T(JJ^T)^{-1}\boldsymbol{\epsilon} \tag{5}$$

C. SD Computation

Let a pair of matched visual-features be denoted by $\mathbf{x}^{c_i} = (x_i, y_i, 1)^T \leftrightarrow \mathbf{x}^{c_j} = (x_j, y_j, 1)^T$ and the inverse of the 3D feature point's depth in the camera coordinate system C_i be denoted by λ_i. We define a measurement vector in $\mathbb{R}^4$ based on the visual-feature pair by $\mathbf{X} = (x_i, y_i, x_j, y_j)^T$, where (x_i, y_i) and (x_j, y_j) are the normalized coordinates of the visual feature on images i and j, respectively. The reprojected visual feature can be obtained by $\tilde{\mathbf{x}}^{c_j} = \hat{\mathbf{x}}^{c_j}/\hat{z}_j$, where $\hat{\mathbf{x}}^{c_j} = (\hat{x}_j, \hat{y}_j, \hat{z}_j)^T$ represents the 3D feature point described in the camera coordinate system C_j. $\hat{\mathbf{x}}^{c_j}$ can be computed from $\mathbf{x}^{c_i}$ via a rigid transformation given by $\hat{\mathbf{x}}^{c_j} = \mathbf{R}_{c_i}^{c_j}\frac{\mathbf{x}^{c_i}}{\lambda_i} + \mathbf{t}_{c_i}^{c_j}$, where $\mathbf{R}_{c_i}^{c_j}$ and $\mathbf{t}_{c_i}^{c_j}$ are the rotation matrix and translation vector from C_i to C_j, respectively. The perspective projection constraint for a pair of error-free visual features is given by $\mathcal{C}_H(\mathbf{X}) = \frac{\hat{\mathbf{x}}^{c_j}}{\hat{z}_j} - \mathbf{x}^{c_j} = 0$. Due to image noise, uncalibrated optical distortion, etc., $\mathcal{C}_H(\mathbf{X}) \neq 0$. An error is thus defined by $\boldsymbol{\epsilon} = \mathcal{C}_H(\mathbf{X}) = (\hat{x}_j/\hat{z}_j - x_j, \hat{y}_j/\hat{z}_j - y_j, 0)^T$, based on which the SD can be computed by using (5). However, due to pose estimation error, $\hat{z}_j$ can be very close to zero, resulting in a singular point and an infinity value of $\mathcal{C}_H$. To avoid this case, we opt to redefine the error as $\boldsymbol{\epsilon} = \hat{z}_j\mathcal{C}_H(\mathbf{X}) = (\hat{x}_j - \hat{z}_j x_j, \hat{y}_j - \hat{z}_j y_j, 0)^T$. In other words, we compute the geometric distance between the feature and the reprojected feature on the image plane with $z = \hat{z}_j$ instead of $z = 1$. Since the third element of $\mathcal{C}_H(\mathbf{X})$ is zero, we rewrite $\boldsymbol{\epsilon}$ by:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \hat{x}_j - \hat{z}_j x_j \\ \hat{y}_j - \hat{z}_j y_j \end{bmatrix} \tag{6}$$

From (5), it can be seen that the redefinition of $\boldsymbol{\epsilon}$ does not change the value of SD while removing the singular point. The 2×4 Jacobian matrix, the partial derivative of $\boldsymbol{\epsilon}$ with respect to $\mathbf{X}$, can be computed by:

$$J = \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{X}} = \left[\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}^{c_i}}, \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}^{c_j}}\right] \tag{7}$$

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}^{c_i}} = \begin{bmatrix} 1,0,-x_j \\ 0,1,-y_j \end{bmatrix} \mathbf{R}_{c_i}^{c_j} \begin{bmatrix} \frac{1}{\lambda_i}, 0 \\ 0, \frac{1}{\lambda_i} \\ 0,0 \end{bmatrix}, \quad \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}^{c_j}} = \begin{bmatrix} -\hat{z}_j, 0 \\ 0, -\hat{z}_j \end{bmatrix} \tag{8}$$

After computing J, the measurement residual ($\mathbf{r}_{ij}^v = \boldsymbol{\delta}_X$) and the SD can be calculated by using (4) and (5). The Jacobian matrices of ${}^v\mathbf{r}_{ij}^k$ with respect to pose variables $\boldsymbol{\xi}_{b_i}^w$ and $\boldsymbol{\xi}_{b_j}^w$ are derived in the APPENDIX.

IV. Comparison between SD and TD

The accuracy of the estimated FM may change when a different error/distance criterion is used. In [18], it is mathematically proved that the symmetric epipolar distance (SED) overestimates the RE and it is larger than the SD as well (SED $\geq \sqrt{2}$RE, SED $\geq \sqrt{2}$SD). The proof was obtained by using the SD criterion based on epipolar constraint. Following this result, our hypothesis for the perspective-projection-based distance criteria for VIO is TD>RE and TD>SD. In this section, we will first prove in theory that TD is larger than both RE and SD and then show by simulation that TD is consistently much larger than RE whereas SD is almost equal to RE.

For a pair of matched visual features $\mathbf{x}^{c_i} = (x_i, y_i, 1)^T \leftrightarrow \mathbf{x}^{c_j} = (x_j, y_j, 1)^T$ with an inverse depth λ_i, the RE is defined as the perpendicular distance in $\mathbb{R}^4$ between $\mathbf{X} = (x_i, y_i, x_j, y_j)^T$ and $\mathcal{V}_H$, where $\mathcal{V}_H$ is implicitly defined by equation $\hat{\mathbf{x}}^{c_j} - \hat{z}_j\bar{\mathbf{x}}^{c_j} = 0$, where $\hat{\mathbf{x}}^{c_j} = (\hat{x}_j, \hat{y}_j, \hat{z}_j)^T = [\mathbf{R}_{c_i}^{c_j}\frac{\bar{\mathbf{x}}^{c_i}}{\lambda_i} + \mathbf{t}_{c_i}^{c_j}]$. RE can be obtained by solving the optimization problem:

$$\begin{aligned} \text{RE} = & \min_{\bar{\mathbf{x}}^{c_i}, \bar{\mathbf{x}}^{c_j}} (||\mathbf{x}^{c_i} - \bar{\mathbf{x}}^{c_i}||^2 + ||\mathbf{x}^{c_j} - \bar{\mathbf{x}}^{c_j}||^2) \\ s.t. \quad & \mathcal{C}_H(\bar{\mathbf{X}}) = \hat{\mathbf{x}}^{c_j} - \hat{z}_j\bar{\mathbf{x}}^{c_j} = 0 \end{aligned} \tag{9}$$

RE represents the needed minimum distance to move point $\mathbf{X} = (x_i, y_i, x_j, y_j)^T$ onto point $\bar{\mathbf{X}}^* = (\bar{x}_i^*, \bar{y}_i^*, \bar{x}_j^*, \bar{y}_j^*)^T$ that lies on $\mathcal{V}_H$. Assuming that the features' coordinates are independent and identically-distributed random variables with a Gaussian distribution, the corrected feature correspondence $(\bar{\mathbf{x}}^{*c_i}, \bar{\mathbf{x}}^{*c_j})$ is the MLE of its true value [10].

A. Compare SD with RE and TD

According to the definition of the SD in section III.B, $||\boldsymbol{\delta}_X||^2 = ||\mathbf{X} - \hat{\mathbf{X}}||^2$, where $\hat{\mathbf{X}}$ is a point on $\mathcal{V}_H$. Since $\bar{\mathbf{X}}$ is the MLE of $\mathbf{X}$ on the variety $\mathcal{V}_H$, RE= $||\mathbf{X} - \bar{\mathbf{X}}||^2$ is the squared perpendicular distance of $\mathbf{X}$ to $\mathcal{V}_H$ [10], meaning $\bar{\mathbf{X}} \in \mathcal{V}_H$ is the closest point to $\mathbf{X}$. Therefore, we can infer $||\mathbf{X} - \hat{\mathbf{X}}||^2 \geqslant ||\mathbf{X} - \bar{\mathbf{X}}||^2$ and SD $\geqslant$ RE.

Assuming that the location of the visual feature is perfect on the 1st image, the TD is calculated as the distance on the 2nd image between the corresponding feature and the forward-projected feature. Given $\mathbf{x}^{c_i} \leftrightarrow \mathbf{x}^{c_j}$ and λ_i, TD is the squared distance between $\mathbf{x}^{c_j}$ and the projection of $\mathbf{x}^{c_i}$ on frame j, denoted $\tilde{\mathbf{x}}^{c_j}$, and it is given by:

$$\text{TD} = ||\mathbf{x}^{c_j} - \tilde{\mathbf{x}}^{c_j}||^2 = (x_j - \frac{\hat{x}_j}{\hat{z}_j})^2 + (y_j - \frac{\hat{y}_j}{\hat{z}_j})^2 = \frac{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}{\hat{z}_j^2} \quad (10)$$

$\tilde{\mathbf{x}}^{c_j}$ has been defined in III.C and vector $\boldsymbol{\epsilon}$ is given in (6). Equation (7) can be rewritten as:

$$J = \left[K, \begin{bmatrix} -\hat{z}_j, 0 \\ 0, -\hat{z}_j \end{bmatrix}\right] \quad (11)$$

where $K = \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{x}^{c_i}}$. Then

$$JJ^T = \hat{z}_j^2\mathbf{I}_2 + KK^T \quad (12)$$

where $\mathbf{I}_2$ is a 2×2 identity matrix. According to Hua's equality [22], $(A+B)^{-1} = A^{-1} - (A + AB^{-1}A)^{-1}$, we can derive:

$$(JJ^T)^{-1} = (\hat{z}_j^2\mathbf{I}_2 + KK^T)^{-1}$$
$$= \frac{1}{\hat{z}_j^2}\mathbf{I}_2 - (\hat{z}_j^2\mathbf{I}_2 + \hat{z}_j^4(KK^T)^{-1})^{-1} = \frac{1}{\hat{z}_j^2}\mathbf{I}_2 - M \quad (13)$$

where $M = (\hat{z}_j^2\mathbf{I}_2 + \hat{z}_j^4(KK^T)^{-1})^{-1}$. Substituting (13) into (5), we have:

$$\text{SD} = ||\boldsymbol{\delta}_X||^2 = \boldsymbol{\epsilon}^T(JJ^T)^{-1}\boldsymbol{\epsilon} = \frac{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}{\hat{z}_j^2} - \boldsymbol{\epsilon}^T M\boldsymbol{\epsilon} = \text{TD} - \boldsymbol{\epsilon}^T M\boldsymbol{\epsilon} \quad (14)$$

Since matrix KK^T is semi-positive, it is obvious that matrix M is positive, meaning $\boldsymbol{\epsilon}^T M\boldsymbol{\epsilon} > 0$. Therefore, we have $\text{TD} = \text{SD} + \boldsymbol{\epsilon}^T M\boldsymbol{\epsilon} > \text{SD}$. Because $\text{TD} > \text{SD} \geqslant \text{RE}$, SD is a better approximation of RE than TD.

V. Simulation and Experimental Results

We first compared the accuracy of SD and TD in estimating the RE and the computational times of SD, TD and RE by simulation. Then, we then modified the TD-based VINS-Mono by using SD. The modified method is called VINS-Mono-SD. We compared the two methods by simulation and by experiments with real-world datasets. A Lenovo ThinkPad T430 (with a duo-core Intel Core i5-3320M CPU and 8 GB RAM) was used for the work.

A. Comparison of distance criteria by simulation

We conducted a simulation to compare TD, SD, and RE computed for a visual feature with a known depth. A thousand feature points were uniformly generated in a 3D space $\{(x,y,z)|_{|x|\leq 5,|y|\leq 5,|z|\leq 5}\}$ (unit: meter) and the 2D visual features were obtained by projecting these feature points onto the 640×480-pixel image plane of a simulated camera with focal-length f=525 and principal point (320,240). The first image frame I_1 was obtained when the camera was at the origin and facing forward, and the second image frame I_2 was obtained after the camera underwent a random pose change $[\mathbf{R};\boldsymbol{t}]$. The projection produces a set of visual-feature-pairs on the two images. Their true locations $\bar{\mathbf{X}} = (\bar{\mathbf{x}}^{c_1}, \bar{\mathbf{x}}^{c_2})$ were computed by using the camera parameters. A zero-mean Gaussian noise ($\sigma_u = \sigma_v = \sigma$) was then added to $\bar{\mathbf{X}}$ to generate the real measurement $\mathbf{X} = (\mathbf{x}^{c_1}, \mathbf{x}^{c_2})$. Given $(\mathbf{x}^{c_1}, \mathbf{x}^{c_2})$ and $[\mathbf{R},\boldsymbol{t}]$, we estimated the 3D feature point by triangulation and use the resulted point to generate $\hat{\mathbf{X}} = (\hat{\mathbf{x}}^{c_1}, \hat{\mathbf{x}}^{c_2})$ by perspective projection. We then used equations (8), (9), and (13) to compute the values for RE, TD, and SD, respectively. In our simulation, we used 12 different σ, ranging from 0.2 to 2.4 pixels (step-size: 0.2-pixel). For each σ value, we repeated the computation 500 times and computed the mean squared residuals for RE, TD, and SD. Fig. 1 plots the mean squared residuals against σ. It can be seen that: 1) the mean squared residual for SD is almost equal to that of RE, and it is much smaller than that of TD, meaning that the estimate $\hat{\mathbf{X}}$ by using SD is more accurate than that by using TD; 2) TD values are consistently much larger than both the RE and SD values; 3) each SD value is very close to the RE value because the initial measurement is almost equal to the optimal solution to the problem defined in (9) and thus the linear approximation is accurate. Simulation results also show that the mean runtimes of RE, SD, and TD (coded with Matlab 2018b) are 6.912, 0.037, and 0.039 milliseconds, respectively. The computation of SD takes much shorter time than that of the RE. These results validate our hypothesis in section IV.

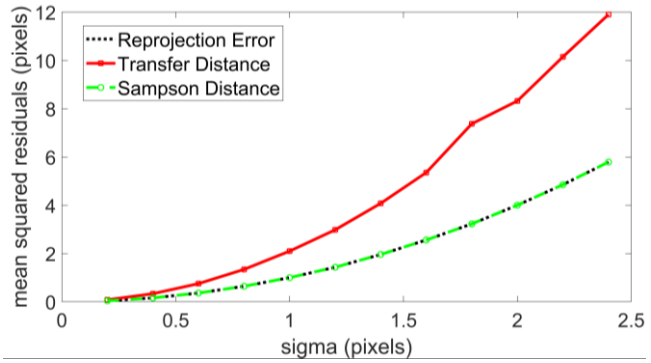

Fig. 1 Comparison of mean squared residuals.

B. SD-based and TD-based VIOs on Synthetic Data

We employed the open-source code [23] to generate simulated visual-inertial data by moving the simulated VINS in a specified trajectory. When running the simulator, we used the default statistical properties for the IMU and changed the image measurement noise ($\sigma_u = \sigma_v = \sigma$) from 0.3 to 2.4 pixels (step-size: 0.3-pixel). For σ value, we ran VINS-Mono and VINS-Mono-SD 10 times. For each run, we first calculated the root mean square error (RMSE) between the estimated trajectory and the ground truth trajectory and then computed the mean RMSE over the 10 runs. Table I summarize the mean RMSE for various σ values. It can be observed that VINS-Mono-SD achieves a smaller mean RMSE than VINS-Mono in all cases and a significant error reduction in 5 out of the 8 cases. The improved pose accuracy results from the utilization of SD to compute the visual feature residuals. Compared to TD, the SD can better estimate the RE, increasing the pose estimation accuracy.

Table I: Mean RMSE of VINS-Mono and VINS-Mono-SD (Pixels)

Noise (σ)	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4
VINS-Mono	0.021	0.036	0.050	0.076	0.081	0.114	0.190	0.243
VINS-Mono-SD	**0.015**	**0.035**	**0.041**	**0.068**	**0.08**	**0.111**	**0.129**	**0.205**
Error Reduction	**28.6%**	**2.8%**	**18.0%**	**10.5%**	**1.2%**	**2.6%**	**32.1%**	**15.6%**

C. *Experimental results with real-world datasets*

We compared the performances of VINS-Mono-SD and VINS-Mono on two public datasets: EuRoc MAV [12] and TUM-VI [13]. Since the ground truth is available, we evaluated the VIO's performance by using the absolute position (3D) error (APE), the RMSE between the ground-truth positions, and the corresponding estimated positions along the entire trajectory. To compute the RMSE, the trajectory produced by each method is aligned with the ground truth trajectory by a full SE3 alignment. The results are summarized in Tables II and III. It can be seen that VINS-Mono-SD achieves a smaller APE than VINS-Mono in seven of the eleven experiments for EuRoc MAV and thirteen of the seventeen experiments for TUM-VI. Its performance degradation in one case for EuRoc and three cases for TUM VI is modest compared with the performance improvement in the other cases. This demonstrates that the VINS-Mono's pose estimation accuracy can be significantly improved by replacing the TD-based residuals with the SD-based residuals for the visual features. Fig. 2 compares the trajectories estimated by the two methods for three selected data sequences, from which it can be observed that VINS-Mono-SD results in a more accurate trajectory.

Table II: Results on the EuRoc MAV Dataset: RMSE of the estimated trajectories of VINS-Mono and VINS-Mono-SD.

Sequence	VINS-Mono	VINS-Mono-SD	RMSE reduction
MH_01_easy	0.16	**0.14**	**12.5%**
MH_02_easy	0.18	**0.13**	**27.8%**
MH_03_medium	**0.19**	0.20	-5.4%
MH_04_difficult	0.35	0.35	0%
MH_05_difficult	0.30	0.30	0%
V1_01_easy	0.09	**0.08**	**11.1%**
V1_02_medium	0.11	0.11	0%
V1_03_difficult	0.19	**0.18**	**5.26%**
V2_01_easy	0.09	**0.08**	**11.1%**
V2_02_medium	0.16	**0.15**	**6.25%**
V2_03_difficult	0.28	**0.26**	**7.14%**

Table III: Results on the TUM VI Dataset: RMSEs of the estimated trajectories of VINS-Mono and VINS-Mono-SD.

Sequence	VINS-Mono	VINS-Mono-SD	RMSE reduction
Room 1	0.07	0.07	0.0%
Room 2	0.07	**0.05**	**28.6%**
Room 3	0.12	**0.10**	**16.7%**
Room 4	0.04	0.04	0.0%
Room 5	0.21	**0.18**	**14.3%**
Room 6	0.07	0.07	0.0%
Corridor 1	0.98	**0.82**	**16.3%**
Corridor 2	0.95	**0.91**	**4.2%**
Corridor 3	1.33	**1.28**	**3.8%**
Corridor 4	0.31	**0.21**	**32.3%**
Corridor 5	0.72	**0.66**	**8.3%**
Magistrale 1	2.18	**2.14**	**1.8%**
Magistrale 2	3.14	**2.69**	**14.3%**
Magistrale 3	**0.41**	0.42	-2.4%
Magistrale 4	4.29	**3.79**	**11.7%**
Magistrale 5	0.86	**0.65**	**24.4%**
Magistrale 6	2.34	**1.60**	**31.6%**

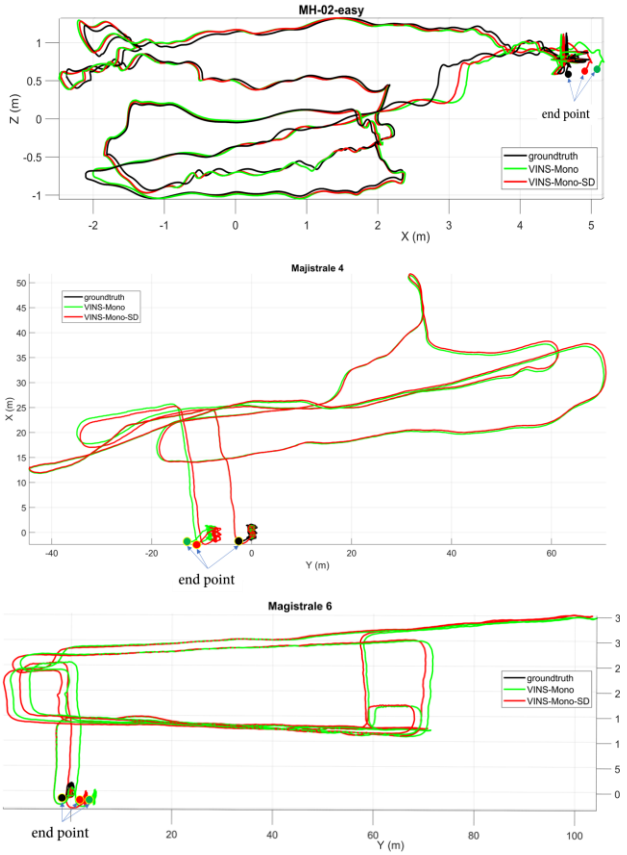

Fig. 2 Comparison between the trajectories estimated by VINS-Mono and VINS-Mono-SD for MH_02_easy, Magistrale 4, and Magistrale 6 data sequences. VINS-Mono-SD results in a trajectory with the endpoint closer to the ground truth for each case, indicating a smaller accumulated positioning error. It is noted that each of the last two datasets only contains ground truth trajectory data for the very beginning/ending parts.

VI. Conclusion

In this paper, we proposed a new SD formulation based on the perspective projection constraint to describe the visual feature residuals for VIO. We proved in theory that the property of the SD in VO translates well into the proposed SD for VIO and the proposed SD provides a more accurate description for a visual feature residual than the prevailing TD criterion used in the state-of-the-art VIO methods. We validated by simulation result that SD is much more accurate than TD and it is a very accurate estimate of the RE, which is the gold standard representation for visual residual. Based on the SD, we modified VINS-Mono by replacing its TD-based visual residuals with the SD-based ones and carried out experiments to compare the performances of the modified VINS-Mono and VINS-Mono and investigate the effectiveness of SD. The experimental results demonstrated that the SD-based VINS-Mono has a significant performance improvement over VINS-Mono in terms of pose estimation accuracy, indicating that SD is a better distance criterion than TD. The work presented in this paper serves as a foundation for its future extension into broader areas such as SLAM in computer vision and robotics.

APPENDIX. JACOBIAN MATRICES FOR SD COMPUTATION

For simplicity, we use ξ_i and ξ_j to represent $\boldsymbol{\xi}_{b_i}^{w}$ and $\boldsymbol{\xi}_{b_j}^{w}$, respectively, in this section. Equation (4) can be re-written by $\mathbf{r}_{i,k}^{v} = \boldsymbol{\delta}_X = -J^T\boldsymbol{\beta}$, where $\boldsymbol{\beta} = A\boldsymbol{\epsilon}$, $A = (B)^{-1}$, $B = JJ^T$. Given $J = \begin{bmatrix} J_{11}, J_{12}, J_{13}, 0 \\ J_{21}, J_{22}, 0, J_{24} \end{bmatrix}$ computed by equations (7)-(9), Jacobian matrices $\frac{\partial J}{\partial \xi_i}$, $\frac{\partial J}{\partial \xi_j}$, $\frac{\partial J^T}{\partial \xi_i}$, and $\frac{\partial J^T}{\partial \xi_j}$ can be computed by using the chain rule as follows:

$$\begin{bmatrix} \frac{\partial J_{11}}{\partial \xi_i} \\ \frac{\partial J_{21}}{\partial \xi_i} \end{bmatrix} = [\mathbf{0}_{2\times 3}, C\mathbf{R}_{b_i}^{c_j}[-\mathbf{R}_c^b\mathbf{d}_1]_\times], \begin{bmatrix} \frac{\partial J_{11}}{\partial \xi_j} \\ \frac{\partial J_{21}}{\partial \xi_j} \end{bmatrix} = [\mathbf{0}_{2\times 3}, C\mathbf{R}_b^c\left[\mathbf{R}_{c_i}^{b_j}\mathbf{d}_1\right]_\times]$$

$$\begin{bmatrix} \frac{\partial J_{12}}{\partial \xi_i} \\ \frac{\partial J_{22}}{\partial \xi_i} \end{bmatrix} = [\mathbf{0}_{2\times 3}, C\mathbf{R}_{b_i}^{c_j}[-\mathbf{R}_c^b\mathbf{d}_2]_\times], \begin{bmatrix} \frac{\partial J_{12}}{\partial \xi_j} \\ \frac{\partial J_{22}}{\partial \xi_j} \end{bmatrix} = [\mathbf{0}_{2\times 3}, C\mathbf{R}_b^c\left[\mathbf{R}_{c_i}^{b_j}\mathbf{d}_2\right]_\times]$$

$$\begin{bmatrix} \frac{\partial J_{13}}{\partial \xi_i} \\ \frac{\partial J_{24}}{\partial \xi_i} \end{bmatrix} = \begin{bmatrix} -\mathbf{e}\mathbf{R}_w^{c_j}, \mathbf{e}\mathbf{R}_{b_i}^{c_j}[\mathbf{p}_{b_i}]_\times \\ \mathbf{e}\mathbf{R}_w^{c_j}, -\mathbf{e}\mathbf{R}_{b_i}^{c_j}[\mathbf{p}_{b_i}]_\times \end{bmatrix}, \begin{bmatrix} \frac{\partial J_{13}}{\partial \xi_j} \\ \frac{\partial J_{24}}{\partial \xi_j} \end{bmatrix} = \begin{bmatrix} \mathbf{e}\mathbf{R}_w^{c_j}, -\mathbf{e}\mathbf{R}_b^c\left[\mathbf{p}_{b_j}\right]_\times \\ -\mathbf{e}\mathbf{R}_w^{c_j}, \mathbf{e}\mathbf{R}_b^c\left[\mathbf{p}_{b_j}\right]_\times \end{bmatrix}$$

where $C = \begin{bmatrix} 1,0,-x_j \\ 0,1,-y_j \end{bmatrix}$, $[\mathbf{d}_1, \mathbf{d}_2] = \begin{bmatrix} \frac{1}{\lambda_i}, 0 \\ 0, \frac{1}{\lambda_i} \\ 0,0 \end{bmatrix}$, $\mathbf{e} = [0,0,1]$, $\mathbf{p}_{b_i} = \mathbf{R}_c^b\frac{\mathbf{x}^{c_i}}{\lambda_i} + \mathbf{t}_c^b$, $\mathbf{p}_{b_j} = \mathbf{R}_{b_i}^{b_j}\mathbf{p}_{b_i} + \mathbf{t}_{b_i}^{b_j}$. Letting $JJ^T = B$, the Jacobian matrices $\frac{\partial B}{\partial \xi_i}$, and $\frac{\partial B}{\partial \xi_j}$ are computed by:

$$\frac{\partial B_{(r+1)(c+1)}}{\partial \xi_i} = \sum_{k=0}^{3} J^T(k,c)\frac{\partial J}{\partial \xi_i}(4r+k,:) + J(r,k)\frac{\partial J^T}{\partial \xi_i}(c+2k,:), r,c = 0..1$$

$$\frac{\partial B_{(r+1)(c+1)}}{\partial \xi_j} = \sum_{k=0}^{3} J^T(k,c)\frac{\partial J}{\partial \xi_j}(4r+k,:) + J(r,k)\frac{\partial J^T}{\partial \xi_j}(c+2k,:), r,c = 0..1$$

Letting $A = (B)^{-1}$, the Jacobian matrices, $\frac{\partial A}{\partial \xi_i}$ and $\frac{\partial A}{\partial \xi_j}$, can be obtained by: $\frac{\partial A}{\partial \xi_i} = F\frac{\partial B}{\partial \xi_i}, \frac{\partial A}{\partial \xi_j} = F\frac{\partial B}{\partial \xi_j}$, where $F = -\begin{bmatrix} A_{11}^2, A_{11}A_{21}, A_{11}A_{12}, A_{12}A_{21} \\ A_{11}A_{12}, A_{11}A_{22}, A_{12}^2, A_{12}A_{22} \\ A_{11}A_{21}, A_{21}^2, A_{11}A_{22}, A_{21}A_{22} \\ A_{12}A_{21}, A_{21}A_{22}, A_{12}A_{22}, A_{22}^2 \end{bmatrix}$. This is achieved by using the chain rule that the partial differential of a matrix's inverse can be computed by $\partial_\xi(inv \circ B) = -B^{-1}\partial_\xi(B)B^{-1}$.

Given $\boldsymbol{\beta} = A\boldsymbol{\epsilon}$, the Jacobian matrices, $\frac{\partial \boldsymbol{\beta}}{\partial \xi_i}$ and $\frac{\partial \boldsymbol{\beta}}{\partial \xi_j}$, are computed by:

$$\frac{\partial \boldsymbol{\beta}}{\partial \xi_i}(r,:) = \sum_{c=0}^{1} \boldsymbol{\epsilon}(c)\frac{\partial A}{\partial \xi_i}(2r+c,:) + A(r,c)\frac{\partial \boldsymbol{\epsilon}}{\partial \xi_i}(c,:), r = 0..1$$

$$\frac{\partial \boldsymbol{\beta}}{\partial \xi_j}(r,:) = \sum_{c=0}^{1} \boldsymbol{\epsilon}(c)\frac{\partial A}{\partial \xi_j}(2r+c,:) + A(r,c)\frac{\partial \boldsymbol{\epsilon}}{\partial \xi_j}(c,:), r = 0..1$$

where $\boldsymbol{\epsilon}$ is defined in (6) and its Jacobian matrices w.r.t. ξ_i and ξ_j are given by:

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \xi_i} = \begin{bmatrix} 1,0,0 \\ 0,1,0 \end{bmatrix}[-\mathbf{x}^{c_j}]_\times[\mathbf{R}_w^{c_j}, \quad -\mathbf{R}_{b_i}^{c_j}[\mathbf{p}_{b_i}]_\times]$$

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \xi_j} = \begin{bmatrix} 1,0,0 \\ 0,1,0 \end{bmatrix}[-\mathbf{x}^{c_j}]_\times[-\mathbf{R}_w^{c_j}, \quad \mathbf{R}_b^c\left[\mathbf{p}_{b_j}\right]_\times]$$

Finally, given $\mathbf{r}_{i,k}^{v} = -J^T\boldsymbol{\beta}$ and the above formulas, the Jacobian matrices, $\frac{\partial \mathbf{r}_{i,j}^{v}}{\partial \xi_i}$ and $\frac{\partial \mathbf{r}_{i,j}^{v}}{\partial \xi_j}$, are computed by:

$$\frac{\partial \mathbf{r}_{i,j}^{v}}{\partial \xi_i}(r,:) = -\sum_{c=0}^{1} \boldsymbol{\beta}(c)\frac{\partial J^T}{\partial \xi_i}(2r+c,:) + J^T(r,c)\frac{\partial \boldsymbol{\beta}}{\partial \xi_i}(c,:), r = 0..3$$

$$\frac{\partial \mathbf{r}_{i,j}^{v}}{\partial \xi_j}(r,:) = -\sum_{c=0}^{1} \boldsymbol{\beta}(c)\frac{\partial J^T}{\partial \xi_j}(2r+c,:) + J^T(r,c)\frac{\partial \boldsymbol{\beta}}{\partial \xi_j}(c,:), r = 0..3$$

REFERENCES

[1] P. Li, *et al*., "Monocular visual-inertial state estimation for mobile augmented reality," *Proc. IEEE Int. Sym. on Mixed and Aug. Reality*, 2017.

[2] S. Weiss, *et al.*, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," *Proc. IEEE International Conference on Robotics and Automation*, 2012.

[3] F. Ma, *et al*., "ACK-MSCKF: Tightly-Coupled Ackermann Multi-State Constraint Kalman Filter for Autonomous Vehicle Localization," *Sensors* 19.21 (2019): 4816.

[4] Z. Shan, R. Li, and S. Schwertfeger, "RGBD-Inertial Trajectory Estimation and Mapping for Ground Robots," *Sensors* 19.10, 2019: 2251.

[5] A. Rosinol, M. Abate, Y. Chang, L. Carlone, "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," *Proc. EEE Int. Conf. Robot. Autom.*, 2020.

[6] H. Zhang and C. Ye, "A Visual Positioning System for Indoor Blind Navigation", *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9079-9085.

[7] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013.

[8] Liu, Haomin, *et al*. "Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics,* vol. 34, no. 4, pp. 1004-1020, 2018.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University press, 2003.

[11] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine,* vol. 19, no. 2, pp. 78-90, 2012.

[12] M. Burri, *et al*., "The euroc micro aerial vehicle datasets," *The Int. Journal of Robotics Research*, 35(10):1157–1163, 2016.

[13] D. Schubert, *et al*.,"The tum vi benchmark for evaluating visual-inertial odometry," *IROS,* 2018, pp. 1680–1687.

[14] P. D. Sampson, "Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm," *Computer graphics and image processing*, vol. 18, no. 1, pp. 97–108, 1982.

[15] Y. Dai, H. Li, and L. Kneip, "Rolling shutter camera relative pose: Generalized epipolar geometry," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4132–4140.

[16] M. Bruckner, F. Bajramovic, and J. Denzler, "Experimental evaluation of relative pose estimation algorithms." *Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2008, pp. 431–438.

[17] D. Nister, "An efficient solution to the five-point relative pose problem," *Proc. IEEE TPAMI*, 2003.

[18] Fathy, Mohammed E., Ashraf S. Hussein, and Mohammed F. Tolba. "Fundamental matrix estimation: A study of error criteria," *Pattern Recognition Letters*, 32.2 (2011): 383-391.

[19] He Zhang, L. Jin, Hao Zhang, and C. Ye, "A comparative analysis of visual-inertial slam for assisted wayfinding of the visually impaired," *Proc. IEEE WACV*, 2019, pp. 210–217.

[20] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans on Robotics,* 24(5): 932-945, 2008.

[21] Peter J Huber, "Robust estimation of a location parameter," *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[22] P. M. Cohn, *Further Algebra and Applications*. Springer Science and Business Media, 2011.

[23] Available online, https://github.com/HeYijia/vio_data_simulation.