IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 22, 2020, accepted February 29, 2020, date of publication March 9, 2020, date of current version March 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978589

The WHU Rolling Shutter Visual-Inertial Dataset

LIKE CAO, JIE LING, (Member, IEEE), AND XIAOHUI XIAO, (Member, IEEE)
Hubei Key Laboratory of Waterjet Theory and New Technology, Wuhan University, Wuhan 430072, China
Corresponding author: Xiaohui Xiao (xhxiao@whu.edu.cn)

This work was supported by the National Key R&D Program of China (2018YFB2100903).

ABSTRACT The vast majority of modern consumer cameras employ a rolling shutter (RS) mechanism which has a price and electronic advantage to global shutter (GS). However, in geometric computer vision applications such as visual simultaneous localization and mapping (VSLAM), performances of accuracy and robustness are usually deteriorated due to the rolling shutter effect when using the RS cameras. This paper introduced the Wuhan University Rolling Shutter Visual-Inertial (WHU-RSVI) synthetic dataset for evaluating VSLAM and VI-SLAM (visual-inertial SLAM) methods in which RS cameras or IMU data are typically used. The proposed synthetic dataset contains RS images, time-synchronized GS images, inertial measurement unit (IMU) measurements, and accurate ground truth. It provides camera images with 640×480 resolution at 30 Hz and IMU measurements from 90 Hz to 14400 Hz. The cubic B-spline curves are used to model the motion of trajectories. Based on the known trajectories, an image of each pose can be rendered, and the corresponding IMU measurement model is then established. The dataset provides realistic images and IMU measurements by modeling the sensor noise in RGB and IMU data. Two trajectories with three sequences of different motion speeds (i. e., slow, medium and fast corresponding to different rolling shutter effects) are contained in the proposed dataset. Herein, the proposed dataset can be applied to compare the impact of different rolling shutter effects on a specific method.

INDEX TERMS Dataset, IMU, rolling shutter camera, SLAM, visual-inertial.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) has been a hot research area for computer vision, robotics, and remote sensing. It is the basic module for many real-time location applications, such as mobile robots, autonomous driving, Virtual Reality (VR), Augmented Reality (AR) and Micro Aerial Vehicle (MAV). SLAM methods can be implemented with a variety of sensors, such as GPS or LiDAR, but cameras and IMUs have been widely studied and applied in recent years for their low cost. There are two types of cameras: RS camera and GS camera. RS cameras are more widely used consumer cameras and have a price and electronic advantage to GS cameras. However, its line-by-line or column-by-column scanning characteristic cause image distortion during it is moving, which is called the rolling shutter effect as in Figure 1.

There are various datasets for evaluating VSLAM related methods such as visual odometry, struct from motion, 3D reconstruction, etc. There are some commonly used datasets in VSLAM related field in the literature. The TUM dataset is used for evaluating the monocular [1], RGB-D [2], and visual-inertial [3] situations respectively. The EuRoC

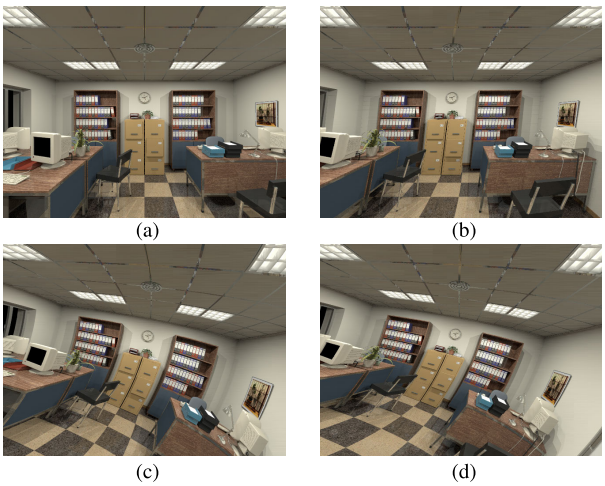

FIGURE 1. Rolling shutter camera with different motion modes. (a) Camera without any motion. (b) Camera with translation. (c) Camera with rotation. (d) Camera with translation and rotation.

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

datasets [4] present a visual-inertial dataset that contains synchronized stereo images and IMU measurements. The KITTI datasets [5], [6] are usually used for autonomous driving. The ICL-NUIM RGB-D dataset [7] for the Evaluation of RGB-D SLAM Systems. Besides, the Oxford RobotCar Dataset [8] is used for different environments and times. Other datasets include the University of Michigan North Campus long-term vision and lidar dataset [9], the Canadian planetary emulation terrain 3D mapping dataset [10] and the Chilean underground mine dataset [11].

All of the above datasets are based on GS images. In the RS situations, the Zurich Urban Micro Aerial Vehicle Dataset [12] uses an RS camera to evaluate appearance-based SLAM and online 3D reconstruction algorithms for MAVs, but it is not specifically for evaluating the RS situations. Kerl *et al.* [13] provide four synthetic RGB-D sequences and four real RGB-D sequences with GS camera models and RS camera models. The synthetic sequences are extended from the ICL-NUIM dataset, the real data sequences are recorded along with the ground truth trajectory from a motion capture system. Schubert *et al.* [14] provide a real dataset that contains GS and RS images, IMU data and ground truth pose for ten different sequences.

The rolling shutter effect is still a big challenge to VSLAM related methods [15]. For a mobile robot using RS cameras, it will seriously reduce its location accuracy and robustness when ignoring the rolling shutter effect. Our synthetic framework is inspired by Handa *et al.* [7] who used a Ray tracing software and synthetic trajectories to obtain ground truth depth maps and color images. To obtain the motion trajectory in real space, we modeled the motion curve through B-Splines. After obtaining the poses at a different position on the trajectories, all images from GS cameras can be rendered by POV-Ray. Each row of an RS image can be obtained by rendering it line-by-line. A complete RS image is obtained by joining every row of an RS image. Through the actual camera noise model, simulated image noise is added to each image to make it more similar to realistic images. By modeling the motion curves and the IMU coordinates system, measurements of IMU is obtained. The simulated IMU noise is also added according to the noise model of realistic IMU. This dataset provides two trajectories, each of which provides three speeds of motion: slow, medium and fast. Each frame of an RS image has a corresponding GS image with the same starting time. Hence, this dataset can also be used to evaluate the differences between GS sequences and RS sequences. All sequences provide accurate ground truth, thus researchers can accurately know the error of a specific method. Different from Kerl *et al.* [13] the dataset of this paper contains IMU data, different from Schubert *et al.* [14], the dataset in this paper is synthetic and each trajectory contains three sequences of different motion speeds (i. e., slow, medium and fast corresponding to different rolling shutter effects), and the IMU data in this dataset are recorded at various rates from 90Hz to 14400 Hz.

This paper generates a dataset for evaluating VSLAM or VI-SLAM methods when researchers use RS cameras. All data, documents, programs, scripts are available online under the Creative Commons Attribution-ShareAlike 3.0 Unported License at http://aric.whu.edu.cn/rsvi-dataset.html

II. SENSORS

A. CAMERA

Sensors used in our dataset include cameras and IMU. There are two types of camera used, GS camera and RS camera.

The dataset was generated by POV-Ray (Persistence of Vision Raytracer),[1] an open-source Ray tracing software. The 3D model for the dataset is the ''office'' model from ignorancia.[2] The size of the office model is $800 \times 500 \times 250\ cm^3$.

All cameras have a resolution of 640×480 and a field angle of 90 degrees. The POV-Ray's coordinates are left-hand, while the right-hand coordinates are often used in the field of VSLAM, so the left-hand coordinates are converted to the right-hand coordinates. After conversion, the camera has an intrinsic matrix shown as

$$\boldsymbol{K} = \begin{bmatrix} 320.0 & 0.0 & 319.5 \\ 0.0 & 320.0 & 239.5 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{1}$$

The ray-tracing is a discrete, digital sampling of the image, usually one sample per pixel. But such sampling can introduce all sorts of errors. A jagged, stepped appearance may appear in a sloping or curved line, or it may result in the loss of details between adjacent pixels. This effect is called aliasing. techniques used to help eliminate these errors or reduce their negative impact on the image is called anti-aliasing. Anti-aliasing is provided in POV-Ray, and all the image in the dataset is rendered at the Anti-aliasing threshold 0.3.

1) MODEL

The image obtained by CCD image sensor is always GS image, and a CMOS image sensor can obtain both GS image and RS image. All pixels are exposed at the same time in a GS camera. In contrast, the pixels in an RS image are scanned row-by-row (or column-by-column) so that pixels in different rows (columns) are not acquired at the same time. The row-by-row scanned mode is used in this dataset. Differences between RS camera and GS camera [16] are shown in Figure 2.

As a result, when the camera or objects are in motion, the image will be distorted. Distortion of the image is related to the way of motion, shown in Figure 3.

In a GS camera, a point $\boldsymbol{P_w}$ in world coordinates and its pixel coordinates P_{uv} satisfy the following projection model:

$$\boldsymbol{P}_{uv} = \boldsymbol{K}(\boldsymbol{R}\boldsymbol{P}_w + \boldsymbol{t}), \tag{2}$$

where $\boldsymbol{K}$ is the intrinsic matrix, $\boldsymbol{R}$ is the rotation matrix, and $\boldsymbol{t}$ is the translation vector.

[1] http://www.povray.org/
[2] http://www.ignorancia.org/

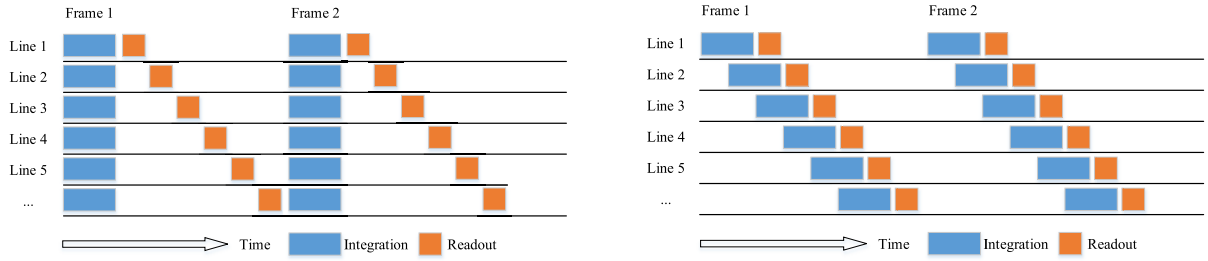

FIGURE 2. The scanning principle of GS camera and RS camera. (a) GS camera model. All rows of an image are exposed simultaneously during a fixed exposure time. (b) Rolling shutter camera model. Each raw of the sensor is sequentially exposed during a fixed exposure time.

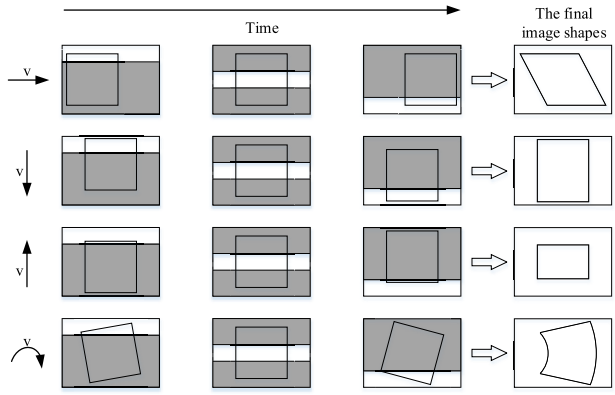

FIGURE 3. Image distortion in different motion modes. v is the direction of camera motion, and the final image shapes are shown on the right side of the figure.

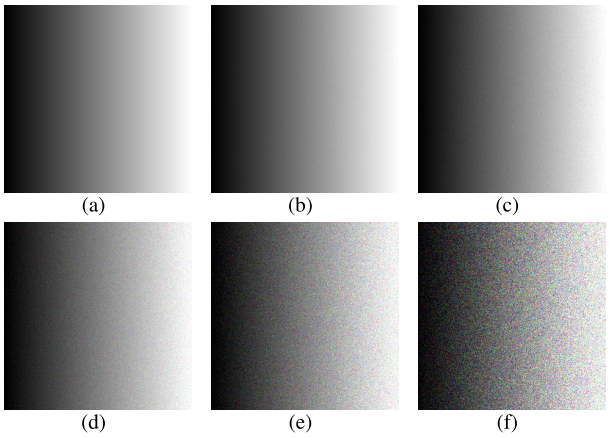

FIGURE 4. Images with different noise levels. (a) Image without any noise. (b) Image with noise of $\sigma_s = 0.01$ and $\sigma_c = 0.005$. (c) Image with noise of $\sigma_s = 0.02$ and $\sigma_c = 0.01$. (d) Image with noise of $\sigma_s = 0.04$ and $\sigma_c = 0.02$. (e)Image with noise of $\sigma_s = 0.08$ and $\sigma_c = 0.04$. (f) Image with noise of $\sigma_s = 0.16$ and $\sigma_c = 0.08$.

In an RS camera, a point $\boldsymbol{P_w}$ in world coordinates and its pixel coordinates P_{uv} satisfy the following projection model:

$$\boldsymbol{P}_{ui} = \boldsymbol{K}\left[\delta \boldsymbol{R}_i \boldsymbol{R} \quad \boldsymbol{t} + \delta \boldsymbol{t}_i\right] \boldsymbol{P}_w, \tag{3}$$

where $\delta \boldsymbol{R}_i$ and $\delta \boldsymbol{t}_i$ are the increments of the rotation matrix and translation vector at the scan of row i relative to the scan of the first row.

2) NOISE

The function representing the relationship between the brightness and irradiance of the image pixel value is called the Camera Response Function (CRF). It is the variety of the linear and nonlinear relations that the camera receives during imaging. Grossberg and Nayar [17] analyzed the real CRF in detail and collected a diverse database of real-world camera response functions (DoRF).

Images generated by POV-Ray are clean and without any noise, whereas the real images are noisy. There are various types of noise in an image, including photon shot noise, dark current Fixed Pattern Noise, dark current shot noise, offset Fixed Pattern Noise, source follower noise, sense node reset noise, and quantization noise [18]. The noise model of CCD image sensor and the CMOS image sensor is a little different, such as in the dark current Fixed Pattern Noise [18], [19]. This dataset only models the most general noise in an image. The noise model was simplified by Liu *et al.* [20], and it can be expressed as

$$I = f\left(L + n_s + n_c\right) + n_q, \tag{4}$$

where I is the image brightness, $f(\cdot)$ denotes the CRF. Linear CRF is a common response model without gamma correction [21]. Images in this dataset are not gamma-corrected, so a Linear CRF is also used in our generated images. n_s represents all the noise components that depend on irradiance L, n_c is the independent noise before gamma correction, and n_q is additional quantization and amplification noise. Since most cameras can achieve very low quantization and amplification noise, so the n_q is ignored in our model. It is assumed that the mean and variance of noise satisfy the following statistics: $\mathrm{E}\left(n_s\right) = 0$, $\mathrm{Var}\left(n_s\right) = L\sigma_s^2$, $\mathrm{E}\left(n_c\right) = 0$, $\mathrm{Var}\left(n_c\right) = \sigma_c^2$.

According to equation (4), realistic noise can be added to the image. Images can be obtained with different noise levels by using different values of σ_s and σ_c. Images with different noise levels are shown in Figure 4. In this dataset, σ_s is set to 0.04, σ_c is set to 0.02, where these values are in line with realistic image noise [20].

B. IMU

1) MODEL

The IMU data is obtained from the trajectories' pose ground truth. However, the ground truth pose is represented in the world coordinates where the acceleration and angular velocity in IMU measurements are represented in the IMU body

coordinates. Therefore, pose in the world coordinates must convert to the IMU measurements in the body coordinates. In this dataset, the origin of world coordinates coincides with that of body coordinates. When the position of sensors in the world coordinates is $\boldsymbol{s} = [x, y, z]^T$, its velocity is $\boldsymbol{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ and its acceleration are $\boldsymbol{a}_w = [\ddot{x}, \ddot{y}, \ddot{z}]^T$.

The special orthogonal group, quaternions, rotation vector and Euler Angles can all representing the attitude [22]. In general, using Euler angles is a bad practice because of the singularities. For instance, in a $z-y-x$ rotation sequence, the gimbal lock occurs when the rotation around the Y-axis is equal to 90 degrees. However, the Euler angles are used to represent rotations in this paper for the reason that compared with other representations, Euler Angles are more simple and intuitive, which makes them well to analyze and control. Since the trajectories in the dataset are artificially designed, so the Euler angles that can be visually interactive are chosen in this paper, and the trajectories in this dataset are designed carefully that avoided the singularities.

The rotations of Euler angles can be extrinsic or intrinsic. The extrinsic means rotation around the axis of the world coordinates, while intrinsic means rotation around the axis of the body coordinates. The results of $[\psi, \theta, \phi]^T$ order of the extrinsic rotations are the same as $[\phi, \theta, \psi]^T$ order of the intrinsic rotations. In POV-Ray, a camera's attitude is representing by Euler angles of extrinsic rotations, while in SLAM applications, the intrinsic rotations are generally used. After conversion, the rotations of Euler angles in this dataset is the $z-y-x$ order of intrinsic rotations, and the rotation matrix corresponding to the Euler angles is expressed as

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_z(\psi)\,\mathbf{R}_y(\theta)\,\mathbf{R}_x(\phi)\\ &= \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}\\ &\quad\times\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi & \cos\phi\end{bmatrix}\\ &= \begin{bmatrix}\cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi\\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi+\cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi\end{bmatrix}.\end{aligned} \tag{5}$$

Gyroscope measurements are the angular velocity of the IMU body coordinates. The Euler angles' derivatives of time are $\left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T$. Assume that the IMU body angular velocity is $g = \left[g_x, g_y, g_z\right]^T$. As described in [9], the first Euler angle ψ undergoes two additional rotations, the second θ undergoes one additional rotation, and the third ϕ without any additional rotations, by considering the inverse relationship where the Euler angle rotation sequence $z-y-x$ is used to map Euler rates to gyroscope measurements as

$$\begin{aligned}\begin{bmatrix}g_x\\ g_y\\ g_z\end{bmatrix} &= \begin{bmatrix}\dot{\phi}\\ 0\\ 0\end{bmatrix} + \mathbf{R}_x^T(\phi)\begin{bmatrix}0\\ \dot{\theta}\\ 0\end{bmatrix} + \mathbf{R}_x^T(\phi)\mathbf{R}_y^T(\theta)\begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix}\\ &= \begin{bmatrix}1 & 0 & -\sin\theta\\ 0 & \cos\phi & \sin\phi\cos\theta\\ 0 & -\sin\phi & \cos\phi\cos\theta\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}.\end{aligned} \tag{6}$$

The acceleration measured by IMU $\boldsymbol{a} = \left[a_x, a_y, a_z\right]^T$ is also the acceleration in the body coordinates. Besides, the acceleration of gravity is considered to be constant as $\mathbf{a}_{gw}$. Therefore the acceleration between body coordinates and world coordinates can be expressed as

$$\mathbf{a} = \mathbf{R}^T(\mathbf{a}_w + \mathbf{a}_{gw}). \tag{7}$$

According to equation (6, 7), IMU measurements can be obtained without any noise.

2) NOISE

Based on the IMU ground truth, the simulated realistic noise can be added to the clean IMU data. Assume that the additional noise of both accelerometer and gyroscope is Gaussian white noise. In addition to the Gaussian white noise, IMU measurements also affected by the accelerometer bias b_a and the gyroscope bias b_w. The IMU measurement model is shown as

$$\begin{aligned}\hat{\mathbf{a}}(t) &= \mathbf{a}(t) + \mathbf{n}_a(t)\\ \hat{\mathbf{g}}(t) &= \mathbf{g}(t) + \mathbf{n}_g(t),\end{aligned} \tag{8}$$

where $\hat{\mathbf{a}}(t)$ denotes the accelerometer raw data, and $\hat{\mathbf{g}}(t)$ denotes the gyroscope raw data. $\mathbf{a}(t)$ is the ground truth of accelerometer. $\mathbf{n}_a(t)$ denotes the noise of the accelerometer. $\mathbf{g}(t)$ is the ground truth of gyroscope and $\mathbf{n}_g(t)$ denotes the noise of gyroscope. The noise process of accelerometer and gyroscope can be written as [4]

$$\begin{aligned}\mathbf{n}_a(t) &= \mathbf{b}_a(t) + \mathbf{w}_a(t)\\ \mathbf{n}_g(t) &= \mathbf{b}_g(t) + \mathbf{w}_g(t),\end{aligned} \tag{9}$$

where $\mathbf{w}_a$ and $\mathbf{w}_g$ are continuous Gaussian white noise process with a strength of σ_a and σ_g. The larger σ_a and σ_g indicate that IMU measurements have more noise. It satisfies the following conditions:

$$\begin{aligned}\mathrm{E}\left[\mathbf{w}_a(t_1)\,\mathbf{w}_a(t_2)\right] &= \sigma_a^2\mathbf{I}\delta(t_1 - t_2)\\ \mathrm{E}\left[\mathbf{w}_g(t_1)\,\mathbf{w}_g(t_2)\right] &= \sigma_g^2\mathbf{I}\delta(t_1 - t_2),\end{aligned} \tag{10}$$

where $\delta(\cdot)$ is the Dirac delta function. It can be seen that the Gaussian white noise at different times is independent of each other.

The discrete-time model of Gaussian white noise is defined as

$$\mathbf{n}_d[k] = \boldsymbol{\sigma}_{nd}\mathbf{w}[k], \tag{11}$$

where $\mathbf{w}[k] \sim \mathcal{N}(0, 1)$,$\boldsymbol{\sigma}_{nd} = \boldsymbol{\sigma}_n\frac{1}{\sqrt{\Delta t}}$.

The accelerometer bias and the gyroscope bias are modeled as a random walk. The random walk is a discrete model while the corresponding continuous model is called the Wiener Process or Brownian motion. Formally, this process is generated by integrating "white noise" of strength σ_{ba}(accel) or σ_{bg}(gyro):

$$\begin{aligned}\dot{\mathbf{b}}_a &= \mathbf{w}_{ba}\\ \dot{\mathbf{b}}_g &= \mathbf{w}_{bg},\end{aligned} \tag{12}$$

where $\mathbf{w}_{b_a} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{ba}^2\right)$, $\mathbf{w}_{b_g} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{bg}^2\right)$. The discrete-time model of the random walk process is as follows:

$$\mathbf{b}_d[k] = \mathbf{b}_d[k-1] + \boldsymbol{\sigma}_{bd}\mathbf{w}[k], \tag{13}$$

where $\mathbf{w}[k] \sim \mathcal{N}(0, 1)$,$\boldsymbol{\sigma}_{db} = \boldsymbol{\sigma}_b\sqrt{\Delta t}$.

Combine the IMU ground truth and its noise model, the realistic noise of the IMU is simulated. Discrete-time random walk noise of the accelerometer and gyroscope at each step is as follows:

$$\begin{aligned}\mathbf{b}_{dbg}[k] &= \mathbf{b}_{dbg}[k-1] + \boldsymbol{\sigma}_{dbg}\sqrt{\Delta t}\mathbf{w}[k]\\ \mathbf{b}_{dba}[k] &= \mathbf{b}_{dba}[k-1] + \boldsymbol{\sigma}_{dba}\sqrt{\Delta t}\mathbf{w}[k].\end{aligned} \tag{14}$$

Therefore, The discrete-time model of IMU can be obtained at time k, and the realistic noise can be simulated step by step according to

$$\begin{aligned}\hat{\mathbf{a}}[k] &= \mathbf{a}[k] + \boldsymbol{\sigma}_a\frac{1}{\sqrt{\Delta t}}\mathbf{w}[k] + \mathbf{b}_{dba}[k-1] + \boldsymbol{\sigma}_{ba}\sqrt{\Delta t}\mathbf{w}[k]\\ \hat{\mathbf{g}}[k] &= \mathbf{g}[k] + \boldsymbol{\sigma}_g\frac{1}{\sqrt{\Delta t}}\mathbf{w}[k] + \mathbf{b}_{dbg}[k-1] + \boldsymbol{\sigma}_{bg}\sqrt{\Delta t}\mathbf{w}[k].\end{aligned} \tag{15}$$

III. DATASET

A. TRAJECTORIES

The movement of mobile robots is usually continuous and smooth. To get a trajectory closer to the real trajectory in simulation, an appropriate trajectory generation method is needed. B-splines are a widely used tool in the configuration of curves and it is proposed by [23]. In this dataset, the cubic B-Spline is used to construct the trajectories curves.

The B-spline curve of n orders is defined as

$$P_{k,n}(t) = \sum_{i=0}^{n} P_{i+k}B_{i,n}(t), \quad t \in [0, 1], \tag{16}$$

where P_{i+k} are control points at time $t_i, i \in [0, \ldots, n]$, and $B_{i,n}(t)$ denote the basis functions [24], where $B_{i,n}(t)$ are defined as follows:

$$B_{i,n}(t) = \frac{1}{n!}\sum\nolimits_{j=0}^{n-i}(-1)^j C_{n+1}^{j}(t+n-i-j)^n \quad t \in [0, 1], \quad i = 0, 1, \ldots, n. \tag{17}$$

The curve is called the cubic B-spline when $n = 3$, and its basis functions are derived as

$$\begin{cases} B_{i,3}(t) = \frac{1}{6}\left(-t^3 + 3t^2 - 3t + 1\right)\\ B_{i+1,3}(t) = \frac{1}{6}\left(3t^3 - 6t^2 + 4\right)\\ B_{i+2,3}(t) = \frac{1}{6}\left(-3t^3 + 3t^2 + 3t + 1\right)\\ B_{i+3,3}(t) = \frac{1}{6}t^3 \end{cases} \quad t \in [0, 1]. \tag{18}$$

A segment of the B-spline is controlled by 4 points: P_i, P_{i+1}, P_{i+2}, P_{i+3}. According to equation (16, 18), the cubic B-spline is expressed in the form of the matrix can be deduced as

$$P_{i,3}(t) = \frac{1}{6}\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}1 & 4 & 1 & 0\\ -3 & 0 & 3 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}P_i\\ P_{i+1}\\ P_{i+2}\\ P_{i+3}\end{bmatrix}, \quad t \in [0, 1]. \tag{19}$$

The locality of cubic B-Spline is mostly considered among its many properties. The locality of B-splines means that every point on the curve is only related to its corresponding control points but nothing to do with others. When the curve needs to be modified, only the local control points need to be modified, and modifications do not affect the rest of the curve. The trajectories will inevitably encounter extreme situations during the generation process. For example, the camera may interfere with the object, which will cause the image to be completely blocked. At this time, it is necessary to modify the local trajectory and re-render the image related to the local trajectory.

The trajectories in this dataset have six variables, namely the translation along the axes of x, y, z, and the rotation around the axes of x, y, z. Since each variable is independent of others, the cubic B-spline interpolation is carried out for each variable respectively. By taking the first-order and second-order derivatives of the curve in time, the velocity and acceleration can be obtained

The trajectory generation steps of this dataset in this paper are as follows:

(1) Set the basic control points manually to get the general outline of the curve.

(2) Perform cubic B-spline interpolation on each control point to get the frame control points for each frame of data.

(3) Perform cubic B-spline interpolation on the frame control points to obtain the pose of each scan line in an image.

Frame control point always covers an average of 1 frame while a basic control point covers multiple frames. In the slow trajectories, a basic control point covers an average of 20 frames, and in the case of medium and fast, it is 10 and 5 frames, respectively.

The reason to perform cubic B-spline interpolation on each basic control point to get the frame control points first is to control the curve more precisely and reduce re-rendering

FIGURE 5. Images of the office room scene taken at different camera pose. Images in the first row are clean and taken by a GS camera, images in the second row are clean and taken by an RS camera, and the images with simulated noise are displayed in the third row.

calculations. For a cubic B-spline curve, when a control point is modified, the adjacent 4 curve segments will change, i.e., if a frame control point is modified, there will be 4 frames that need to be re-rendered. When only basic control points are used, taking the slow trajectories as the example, modification of a basic control point will cause 80 frames to be re-rendered (the number of interpolations needs to be increased by 20 times to get the same number of frames), which greatly increases the calculations. Therefore, in this paper, frame control points are obtained first, and then the pose of each scan line in an image is obtained.

This method allows for a flexible trajectory in three-dimensional space with only a few control points. Then the IMU ground truth can be obtained by equation (6, 7).

In the same trajectory, if the camera or object moves at different speeds, the image will be obtained at different rolling shutter effects. The faster the camera or object moves, the more obvious the rolling shutter effect is. Conversely, an RS camera is identical to a GS camera when everything is static.

The dataset is obtained in an office room scene. The model of the scene is modified to meet the requirements of the dataset. The overview of the dataset is shown in Figure 5.

Each trajectory contains three sequences of different motion speeds (fast, medium and slow). Although the shapes of three speeds are slightly different at the micro-level, they look very similar overall. The detail information of the trajectory is shown in Table 1.

The trajectories estimation overview is shown in Figure 6 and the absolute trajectory error (ATE) results are shown in Table 2. The evo tools [25] are used to plot the trajectories estimation figures and the VINS-Mono [26] is used to estimate the detailed results. The bold numbers represent the best estimation of each trajectory, while the red numbers represent the worst. Results generally show that the faster a camera moves, the better the estimation is when using a GS camera. On the contrary, the faster a camera moves, the worse the estimation is when using an RS camera.

TABLE 1. Trajectories characteristics.

Sequences	Frames& Duration	Length	Avg.Vel.
t1-slow	2895&96.50s	50.1071m	0.5192m/s
t1-medium	1515&50.20s	50.0052m	0.9902m/s
t1-fast	825 &27.50s	49.7071m	1.8099m/s
t2-slow	3000&100.00s	53.1682m	0.5317m/s
t2-medium	1570&52.33s	53.1245m	1.0151m/s
t2-fast	855 &28.50s	53.0256m	1.8605m/s

B. COORDINATE FRAME CONVENTIONS

In this section, we describe the coordinates frame conventions used in the WHU-RSVI dataset.

The rotation around the fixed coordinates axis used in the POV-Ray when an image is rendering. The POV-Ray uses a left-hand coordinates system where in the area of VSLAM, the right-hand coordinates are used mostly. In this case, all the coordinates used are right-hand coordinates after the images are rendered. When the coordinates are converted, the X-axis remains unchanged in the new coordinates system, while the Y-axis direction in the new coordinates is consistent with the Z-axis direction of the previous coordinates, and the Z-axis direction in the new coordinates is consistent with the Y-axis direction of the previous coordinates system. Moreover, the rotation around the left-hand coordinates becomes a rotation of the right-hand coordinates of equal magnitude and opposite directions. The origin of the IMU coordinates

TABLE 2. Trajectories estimation.

trajectories	max error (m)	mean error (m)	median error (m)	min error (m)	rmse error (m)	std error (m)
t1 global fast	**0.119917**	**0.043134**	**0.0354994**	**0.0051823**	**0.0502126**	**0.0256642**
t1 rolling fast	1.394272	0.580265	0.541298	0.1105736	0.6590849	0.3123358
t1 global medium	0.159047	0.05668	0.0488485	0.006317	0.0659048	0.0335315
t1 rolling medium	0.38173	0.134269	0.1230065	0.0109883	0.1556656	0.0787418
t1 global slow	0.402605	0.153546	0.1491079	0.019786	0.1729317	0.0789926
t1 rolling slow	0.414886	0.150369	0.1400567	0.0111652	0.1726985	0.0848354
t2 global fast	0.032488	**0.010394**	**0.0094207**	0.0011641	**0.0117331**	0.0054092
t2 rolling fast	2.540828	0.685925	0.6164916	0.033027	0.8212265	0.4485892
t2 global medium	**0.031741**	0.010554	0.0097541	**0.0009457**	0.0118302	**0.0053232**
t2 rolling medium	0.298055	0.065875	0.0430071	0.0046874	0.0929791	0.0656141
t2 global slow	0.061656	0.014328	0.0123965	0.0013632	0.0174084	0.0097117
t2 rolling slow	0.059141	0.021181	0.0203814	0.0014179	0.0236289	0.0103383

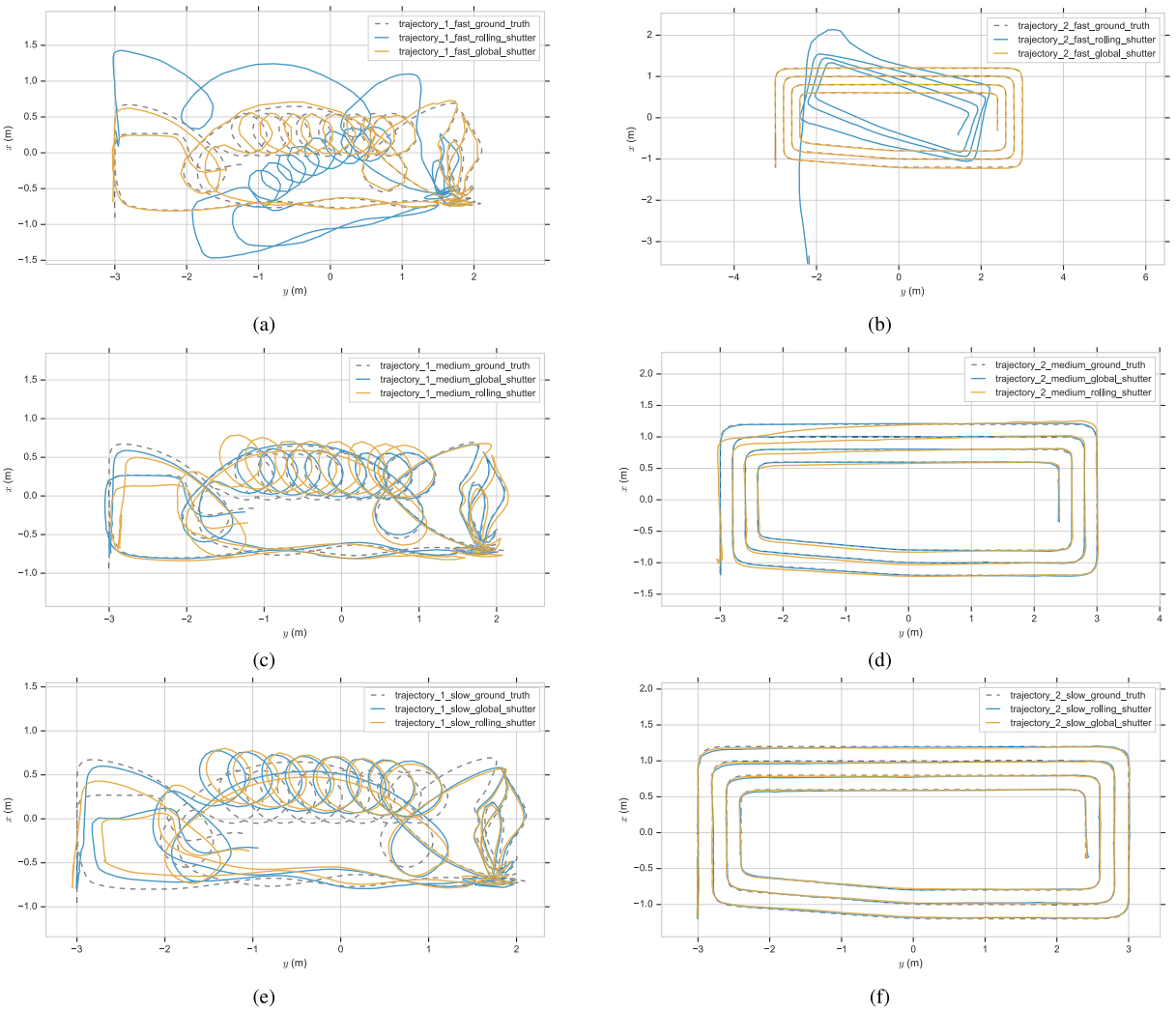

FIGURE 6. Trajectories of different speeds and camera modes estimated by VINS-Mono. (a) trajectory-1 with fast motion. (b) trajectory-2 with fast motion. (c) trajectory-1 with medium motion. (d) trajectory-2 with medium motion. (e) trajectory-2 with slow motion. (f) trajectory-1 with slow motion.

is coincident with the origin of the camera coordinates system. This is different from the general situation, in practice, the camera's coordinates origin and the IMU's coordinates origin have a translation in addition to the rotation.

The Local east, north, up (ENU) coordinates are used in the IMU coordinates. It is far more intuitive and practical than Earth-Centered, Earth-Fixed coordinates. In the local ENU coordinates, the east axis is labeled x, the north axis is labeled y, and the up axis is labeled z. The camera coordinates have a rotation of 90 degrees around the X-axis relative to the IMU coordinates in this dataset.

TABLE 3. Dataset description.

Folder/Filename	Description
./trajectory-n	The root folder of trajectory n.
−./speed/	Speed has three options: 'slow', 'medium' and 'fast'
−−./camera type/	Camera type can be 'gs' or 'rs' which represent the GS camera and the RS camera respectively.
− − −./noise type/	Noise type can be 'clean' or 'noise' which represents clean images and noisy images respectively.
− − −−./img/	Folder to store a sequence of images
− − − − −./frame_n.png	Image of frame n, 8-bit RGB format.
− − −−./data.csv	EuRoC format compatible timestamps of images.
− − −−./sensor.yaml	EuRoC format compatible camera parameters.
− − −−./times.txt	TUM format compatible timestamps of images.
−−./imu/	Folder for storing IMU related information.
− − −./data-nHz.csv	EuRoC format compatible data for nHz IMU data.
− − −./sensor.yaml	EuRoC format compatible IMU extrinsic parameters include noise density, etc.
−−./gt/	Ground truth Folder.
− − −./sensor.yaml	IMU extrinsic parameters ground truth.
− − −./groundtruth-nHz.csv	EuRoC format compatible ground truth at nHz rate.
− − −./groundtruth.txt	TUM format compatible ground truth at the image frame rate.
./readme.txt	More detailed descriptions about the files or folders listed above

C. DATA FORMAT

The dataset contains time-synchronized RS images and GS images at full resolution (640 × 480), and full frame rate (30 Hz). In an RS image, the time between two consecutive rows is 69.44 microseconds. The IMU data are recorded at various rates from 90Hz to 14400 Hz. The structure of the dataset is shown in Table 3.

1) IMAGES

All camera images are 8-bit RGB format. From a noise point of view, images include clean images and images with realistic noise. From the perspective of the sensors, images include RS images and the GS images. For each image sequence, a TUM compatible file *times.txt* and a EuRoC compatible file *data.csv* are created. The camera parameters are described in the *sensor.yaml* under the */img* folder which includes its extrinsic matrix, frame rate, resolution, and camera model.

2) IMUs

The */imu* folder includes two EuRoC compatible files, the *data.csv*, and the *sensor.yaml*. The *data.csv* records timestamps in nanoseconds. The units of the accelerometer and gyroscope are rad/s and m/s^2 respectively. Both accelerometer and gyroscope data have an 18-bit accuracy. The *sensor.yaml* in the */imu* folder records the extrinsic matrix to the body frame coordinates, the rate of data, the accelerometer noise density, the gyroscope noise density, the accelerometer random walk density, and the gyroscope random walk density.

3) GROUND TRUTH

The dataset provides ground truth with different frame rates from 90Hz to 14,400Hz. The *groundtruth* folder includes three types of files: *sensor − camera.yaml*, *sensor.yaml*, and different *groundtruth − nHz.csv*. The *sensor.yaml* records the extrinsic matrix of IMU. The *groundtruth.txt* is TUM compatible file that records the translation and quaternion in the world coordinates with timestamps. The *groundtruth − nHz.csv* is EuRoc compatible file that records the translation, quaternion, and velocity in the world coordinates, and gyroscope measurements, accelerometer measurements in the body-frame coordinates.

IV. CONCLUSION

This paper presents a new synthetic dataset mainly aiming at VSLAM or VI-SLAM related methods where the RS cameras and IMU sensors are widely used. For the same trajectory, sequences of three different motion speeds can be used to evaluate the influence of different levels of rolling shutter effect. The time-synchronized global shutter sequences can be used to compare with the RS sequences when they are used on a specific method. The sequences are estimated by the VI-SLAM framework VINS-Mono, and the results generally show that the faster an RS camera moves, the smaller the absolute trajectory error of a sequence.

Compared with other datasets obtained by a motion capture system, the ground truth of a trajectory is calculated without sampling error. Also, it has a variety of IMU rates, which is not available in other works. Finally, our dataset provides the same scenario, but different motion speeds. These characteristics make the researchers a better understanding of the VI-SLAM related methods.

This dataset can be extended into the other areas of computer vision and robotics. In the future, we will seek to expand the dataset with more sequences with motion blur and various image noise. The IMU is often used with the magnetometer, hence, adding the magnetometer data into the dataset is another research aspect.

ACKNOWLEDGMENT

The authors would like to thank Jaime Vives Piqueres who provided us with the open-source office scene.

REFERENCES

[1] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," 2016, *arXiv:1607.02555*. [Online]. Available: http://arxiv.org/abs/1607.02555

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[3] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1680–1687.
[4] 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. Res.*, vol. 35, no. 10, pp. 1157–1163, Jan. 2016.
[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.
[7] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 1524–1531.
[8] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
[9] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan north campus long-term vision and lidar dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.
[10] C. H. Tong, D. Gingras, K. Larose, T. D. Barfoot, and É. Dupuis, "The Canadian planetary emulation terrain 3D mapping dataset," *Int. J. Robot. Res.*, vol. 32, no. 4, pp. 389–395, Mar. 2013.
[11] K. Leung, D. Lühr, H. Houshiar, F. Inostroza, D. Borrmann, M. Adams, A. Nüchter, and J. R. del Solar, "Chilean underground mine dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 16–23, Jan. 2017.
[12] A. L. Majdik, C. Till, and D. Scaramuzza, "The Zurich urban micro aerial vehicle dataset," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 269–273, Apr. 2017.
[13] C. Kerl, J. Stuckler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2264–2272.
[14] D. Schubert, N. Demmel, L. von Stumberg, V. Usenko, and D. Cremers, "Rolling-shutter modelling for direct visual-inertial odometry," 2019, *arXiv:1911.01015*. [Online]. Available: http://arxiv.org/abs/1911.01015
[15] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2878–2885, Oct. 2018.
[16] M. Meilland, T. Drummond, and A. I. Comport, "A unified rolling shutter and motion blur model for 3D visual registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2016–2023.
[17] M. D. Grossberg and S. K. Nayar, "Modeling the space of camera response functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1272–1282, Oct. 2004.
[18] M. Konnik and J. Welsh, "High-level numerical simulations of noise in CCD and CMOS photosensors: Review and tutorial," 2014, *arXiv:1412.4031*. [Online]. Available: http://arxiv.org/abs/1412.4031
[19] A. El Gamal, B. A. Fowler, H. Min, and X. Liu, "Modeling and estimation of FPN components in CMOS image sensors," *Proc. SPIE*, vol. 3301, pp. 168–178, Apr. 1998.
[20] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 901–908.
[21] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, "Real-time camera tracking: When is high frame-rate best?" in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 222–235.
[22] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, nos. 15–16, pp. 1–35, 2006.
[23] W. J. Gordon and R. F. Riesenfeld, "B-spline curves and surfaces," in *Computer Aided Geometric Design*. Amsterdam, The Netherlands: Elsevier, 1974, pp. 95–126.
[24] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," *Int. J. Comput. Vis.*, vol. 113, no. 3, pp. 208–219, Feb. 2015.
[25] M. Grupp. (2017). *EVO: Python Package for the Evaluation of Odometry and SLAM*. [Online]. Available: https://github.com/MichaelGrupp/evo
[26] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

LIKE CAO received the B.S. degree in mechanical engineering from the School of Power and Mechanical Engineering, Wuhan University, Wuhan, China, in 2012, and the M.S. degree from the Department of Precision Instrument, Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in mechanical engineering with Wuhan University.

His research interests include visual SLAM, visual-inertial SLAM, and robotics.

JIE LING (Member, IEEE) received the B.S. and Ph.D. degrees in mechanical engineering from the School of Power and Mechanical Engineering, Wuhan University, Wuhan, China, in 2012 and 2018, respectively.

He was a joint Ph.D. Student with the Department of Automatic Control and Micro-Mechatronic Systems, FEMTO-ST Institute, France, in 2017. Since 2019, he has been a Joint Postdoctoral Researcher with the Department of Biomedical Engineering, National University of Singapore, Singapore. Since 2018, he has also been a Postdoctoral Researcher with the Department of Mechanical Engineering, Wuhan University. His research interests include mechanical design and precision motion control of nanopositioning stages and micromanipulation robots.

XIAOHUI XIAO (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Wuhan University, Wuhan, China, in 1991 and 1998, respectively, and the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, in 2005.

From 2006 to 2008, she was a Research Fellow with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, USA. She joined Wuhan University, in 1998, where she is currently a Full Professor with the Mechanical Engineering Department, School of Power and Mechanical Engineering. She has published a number of articles in the areas of mobile robots, dynamics and control, and sensors and signal procession. Her current research interests include mobile robotics, high-precision positioning control, and signal processing.