ASRO-DIO: Active Subspace Random Optimization Based Depth Inertial Odometry

Jiazhao Zhang, Yijie Tang, He Wang, *Member, IEEE*, and Kai Xu, *Senior Member, IEEE*

***Abstract*—High-dimensional nonlinear state estimation is at the heart of inertial-aided navigation systems (INS). Traditional methods usually rely on good initialization and find difficulty in handling large interframe transformations due to fast camera motion. We opt to tackle these challenges by solving the depth inertial odometry (DIO) problem with random optimization. To address the exponentially increased amount of candidate states sampled for the high-dimensional state space, we propose a highly efficient variant of random optimization based on the idea of active subspace. Our method identifies the active dimensions, which contribute most significantly to the decrease of the cost function in each iteration, and samples candidate states only within the corresponding subspace. This allows us to efficiently explore the 18D state space of DIO and achieve good optimality by sampling and evaluating only thousands of candidate states. Experiments show that our method attains highly robust and accurate DIO under fast camera motions and low light conditions, without needing a slow-motion warm-up for initialization.**

***Index Terms*—Depth-inertial odometry (DIO), evolution strategy, simultaneous localization and mapping, state estimation.**

I. Introduction

In THE fields of robotics and vision, extensive research has been devoted to inertial-aided navigation systems (INS) [1], [2] for motion estimation in GPS-denied environments such as indoor rooms. In the literature, INS is typically realized with either filtering-based [1], [3], [4], [5], [6], [7], [8], [9] or optimization-based approaches [10], [11], [12], [13], [14], [15]. The basic principle underlying both approaches is the linearization of the nonlinear problem of motion estimation. In particular, the iterated Kalman filter is proven to be equivalent to the Gaussian–Newton algorithm [2], [16]. The success of such linearization usually hinges on a good initialization and requires that the change of states in each time-step is small so that the nonlinearity is not high. This makes the existing methods sensitive to initialization and error-prone when handling fast camera motion.

To mitigate these issues, a promising approach is random optimization. Random optimization estimates states by sampling a population of candidates and evaluating them with a cost function. It excels at finding more global optima in highly nonlinear optimization problems [17]. Recently, it has also been adopted to online RGB-D reconstruction under fast camera motion, showing good performance [18]. However, existing state estimators based on random optimization are mostly restricted to *low dimensional state space* (e.g., 6 DoFs of camera motion). To ensure a good exploration of state space, the amount of samples increases exponentially as the dimensionality grows [19], [20], [21]. This makes random optimization computationally prohibitive for INS, where a much higher dimensional state space (e.g., 18 DoFs of IMU states) is involved.

In this work, we present *active subspace random optimization* for solving the high-dimensional nonlinear state estimation of INS. Our key observation is that only a fraction of the dimensions have a significant impact on the cost function and the subset of active dimensions changes dynamically over iteration steps. Similar findings have also been reported and exploited in alternative optimization frameworks [21], [22], [20]. During the iterations of random optimization, we dynamically identify active dimensions based on the sampling efficiency of a dimension. Given a dimension, its sampling efficiency is measured by the ratio of the amount of state update over the range of search along that dimension. The amount of state update is probed with a set of presampled probing states.

With the active subspace, our method attains a good balance between the optimality and the computational cost of the optimization. As such, all states are estimated within the optimization framework; no specially designed initialization scheme is needed as in [15], [23], [24]. The main bottleneck of random optimization is the iterated sampling and the evaluation of candidate solutions (states). To accelerate the randomized search, we opt to presample states in the solution space, forming a presampled state template (PST), and then move and rescale the PST to explore and search in the solution space, similar to ROSEFusion [18]. Unlike ROSEFusion, where PST is constructed with uniform sampling, we take advantage of the prior knowledge of INS state space and work with different sampling spaces for different components (dimensions) of the states. In particular, orientation and gravity vector are sampled in $SO(3)$, velocity

Manuscript received 8 June 2022; accepted 25 August 2022. This work was supported in part by the National Nature Science Foundation of China under Grant 62132021 and in part by the National Key R&D Program of China under Grant 2018AAA0102200. This article was recommended for publication by Associate Editor J. Civera and Editor F. Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Kai Xu.)*

Jiazhao Zhang, Yijie Tang, and Kai Xu are with the Department of Computer Science, National University of Defense Technology, Changsha 410000, China (e-mail: zhngjizh@gmail.com; yjtang1024@gmail.com; kevin.kai.xu@gmail.com).

He Wang is with the Department of Computer Science, Peking University, Beijing 100871, China (e-mail: hewang@pku.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2022.3208503.

Digital Object Identifier 10.1109/TRO.2022.3208503

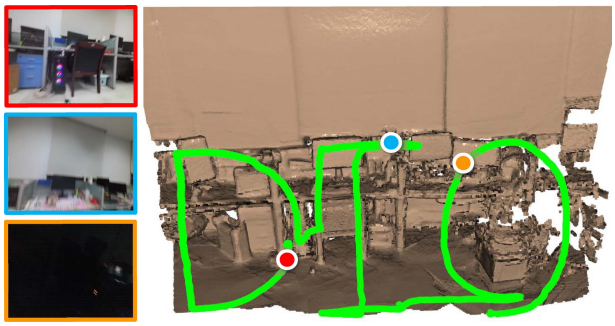

Fig. 1. Tracking and mapping result of our DIO for a sequence drawing letters "DIO." The sequence is captured with a hand-held depth camera under challenging conditions including quick start (no slow-motion warm up; the red point and the red RGB frame), fast camera motion (blue, zoom in to see the motion blur), low light (orange). The tracked camera trajectory is depicted with green curve and the scene is reconstructed with TSDF-based depth fusion.

and position with uniform distribution, and IMU measurement error from Gaussian distribution. Furthermore, we devise different updating (moving, rescaling, and weighted averaging) mechanisms for the three sampling spaces for better efficiency.

Although our optimization approach is general and adaptable to vision-inertial odometry (VIO), we focus on depth-inertial odometry (DIO) in this work (see Fig. 1). While VIO has a large body of prior works, DIO has not been extensively studied, partly due to its special challenges (e.g., lack of reliably trackable features). However, DIO has several advantages such as robustness to poor lighting condition and resilience to motion blur caused by fast motion, making it especially important to robotic applications. We adopt the truncated signed distance field (TSDF) based mapping framework [25], [26] which allows for depth-based fusion. More importantly, TSDF map facilitates a purely depth-based cost function via measuring the depth-TSDF conformance [27]. The cost function is parallel-friendly and admits a GPU implementation, enabling fast evaluation of candidate solutions in random optimization.

We have evaluated our method on several benchmarks. On the ETH3D benchmark [28], our method achieves the state-of-the-art accuracy under fast camera motions (*i.e.*, shaking cameras). We also contribute a robotic arm shaking (RAS) dataset of robot-operated fast-motion sequences with quality ground-truth trajectories. On RAS, our method shows high robustness and succeeds on all the sequences with low light conditions and high speed motions. Extensive quantitative and qualitative comparisons also demonstrate the significant advantage of our method over the state-of-the-arts, under extreme conditions and without a slow-motion warm up for initialization. *All source code and dataset will be released.*

Our main contributions include the following.

1) We present a practically robust tightly coupled solution to DIO based on random optimization.
2) We propose to utilize active subspace random optimization to tackle the high-dimensional state estimation problem in inertial-based navigation systems.
3) We propose a carefully designed PST through exploiting the prior knowledge of INS to ensure high sampling efficiency.
4) We have implemented a navigation system based on our technique. It realizes robust and accurate tracking under challenging conditions such as low light and fast camera motions, without needing a slow-motion warm up for initialization.

II. Related Work

a) Depth Inertial Navigation System: There is a large body of literature on INS. We decide to review only those which are highly related to our work. In [29], a tightly coupled EKF framework is proposed for pose estimation and IMU-RGBD extrinsic calibration. Since then, most works focus on how to incorporate depth information into VIO frameworks. Representative approaches are formulated either as filters [30], [31] or with second-order optimization [23], [24], [32], [33].

In the context of DIO system, however, there are only a few existing works, which leverage the ICP technique and develop loosely coupled [34], [35] or tightly coupled [32] navigation systems. Departing from these works, our work contributes the first random optimization framework for tightly coupled DIO, achieving highly robust and accurate tracking under various extreme conditions.

b) State Estimation Based on Sampling-Based Optimization: In the robotics and computer vision community, perhaps the most commonly used sampling-based algorithm is particle filter, and more specifically, Rao–Blackwellized particle filter (RBPF) [36]. There has been extensive research on particle filter based SLAM [37], [38], [39] and object pose tracking [40], [41], [42], [43]. Different from these works in which sampling-based algorithm is used for sequential state estimation across time steps, our method performs sampling-based random optimization to estimate the state *within a single time step*.

Another line of optimization approaches is evolutionary algorithms (EAs), including particle swarm optimization (PSO) [44], evolution strategies (ESs) [45], random optimization (RO) [46], etc. These methods involve heuristically designed sampling and updating strategies, often requiring extremely large computing resources. EAs are limited to low-dimensional and time-insensitive scenarios. Recently, Zhang et al. [18] proposed an online dense reconstruction method in which the core is 6-DoF camera tracking based on RO powered by PST. We substantially extend this work to deal with the high dimensional solution space of DIO through leveraging active subspace.

c) Active Subspace for High-Dimensional Optimization: High dimensional solution space induces very high computational and memory cost, and usually causes poor convergence. To tackle this difficulty, active subspace techniques and extensions [47] are becoming popular for dimensionality reduction on the fly during optimization. A classic method of obtaining active subspace is random projection [48], [49], which is simple but hard to be effective with a totally random selection. Recent works advance the active subspace idea [19], [21], [22] by evaluating the importance of different dimensions based on various heuristics or priors. Our method, sharing the same insights, identifies active subspace via measuring the sampling efficiency along different dimensions.

III. Method Overview

In this article, we introduce a novel method for DIO leveraging active subspace random optimization. Our system is based on a platform, on which we mount a depth sensor and a 6-axis inertial measurement unit (IMU). We assume that the transformation between the camera and the IMU is given by prior calibration. Our proposed method takes as inputs a live stream of depth observations along with IMU measurements and outputs the 6 DoF pose of the platform, namely 3-D orientation and 3-D position. We formulate DIO as an online optimization problem and construct a cost function based on depth-TSDF conformance [18] and IMU measurement residuals. Compared with ROSEFusion [18] that is depth-only, our method further leverages IMU measurement and thus incorporates the IMU related states into our state space, resulting in a high state space dimensionality. To achieve robust and real-time optimization for this high dimensional state estimation problem, we propose active subspace random optimization method. Note that our method does not rely on RGB images, which are shown in the article only for visualization.

Our methodology is structured as follow: We formulate the optimization problem of DIO in Section IV by first introducing the variables to be estimated (see Section IV-A and Section IV-B), also called candidate states in random optimization. To evaluate the candidate states, we leverage the depth-TSDF conformance (see Section IV-C1) and IMU measurement residual (see Section IV-C2) to construct our tightly coupled cost function (see Section IV-D). Based on the evaluated candidate states, we present the random optimization for state estimation (see Section V). After a brief review of the random optimization framework (see Section V-A), we then introduce how to extend the random optimization to high dimensional states estimation by the predefined state space (see Section V-B) and active subspace (see Section V-C).

IV. Optimization Formulation

A. Notations and Definitions

In this article, we deploy the following notations: superscripts $(\cdot)^W$, $(\cdot)^C$, and $(\cdot)^B$ denote variables in the world frame, camera frame, and IMU body frame, respectively; subscripts $(\cdot)_C$ and $(\cdot)_B$ denote quantities associated with the camera and the IMU body, respectively; $\hat{(\cdot)}$ denotes the values of measurements, in contrast with their true states $(\cdot)$; subscripts $()_t$ and $()_k$ represent quantities at the time step t of the high frame-rate IMU measurements and quantities at the kth camera frame, respectively.

Then, in the world frame, we can represent the position, velocity, and orientation of the IMU as $\mathbf{p}_B^W$, $\mathbf{v}_B^W$, $\mathbf{q}_B^W$, respectively. As the IMU measurements are done in its body frames, we can represent the measurements as $\hat{\boldsymbol{\omega}}_t^B$ from the gyroscope and $\hat{\mathbf{a}}_t^B$ from the accelerometer and their corresponding measurements errors as $\mathbf{E}_a^B$ and $\mathbf{E}_g^B$, respectively. We further denote gravity vector as $\mathbf{g}^W$.

In this work, we are only interested to estimate the states at the frame rate of the depth camera. At the kth camera frame, the state $\mathbf{x}_k$ to be estimated is shown as the following:

$$\mathbf{x}_k = \left[{\mathbf{p}_{B_k}^W}^T, {\mathbf{v}_{B_k}^W}^T, {\mathbf{q}_{B_k}^W}^T, {\mathbf{g}^W}^T, {\mathbf{E}_{a_k}^B}^T, {\mathbf{E}_{g_k}^B}^T \right]^T. \quad (1)$$

Note that, our method only samples the imaginary part of the unit quaternion to represent rotation. Our actual state space is therefore 18-DoF. All the states, including gravity vector $\mathbf{g}^W$, are estimated by the random optimization framework without the need for specifically designed initialization like [15], [23].

B. IMU Kinematic Model

Following the existing works [1], [15], we adopt the IMU model as shown follows:

$$\begin{aligned} \hat{\mathbf{a}}_t^B &= \mathbf{a}_t^B + \mathbf{q}_W^B \otimes \mathbf{g}^W + \mathbf{E}_{a_t}^B \\ \hat{\boldsymbol{\omega}}_t^B &= \boldsymbol{\omega}_t^B + \mathbf{E}_{g_t}^B \\ \mathbf{E}_{a_t}^B &= \mathbf{b}_{a_t}^B + \mathbf{n}_{a_t}^B \\ \mathbf{E}_{g_t}^B &= \mathbf{b}_{g_t}^B + \mathbf{n}_{g_t}^B \end{aligned} \quad (2)$$

where $\mathbf{q}_W^B = \mathbf{q}_B^{W-1}$ is the corresponding quaternion for the rotation from the world frame to the IMU body frame, $\otimes$ represents the Hamilton quaternion multiplication, the measurement errors $\mathbf{E}_t^B$ (can either be $\mathbf{E}_{a_t}^B$ or $\mathbf{E}_{g_t}^B$) is assumed to be comprised of an bias term $\mathbf{b}_t^B$ (*i.e.*, $\mathbf{b}_{a_t}^B$ or $\mathbf{b}_{g_t}^B$) and an additive noise term $\mathbf{n}_t^B$ (*i.e.*, $\mathbf{n}_{a_t}^B$ or $\mathbf{n}_{g_t}^B$).

Here, we assume the additive noise $\mathbf{n}_t^B$ are white Gaussian noise, *i.e.*, $\mathbf{n}_{a_t}^B \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{n}_a}^2)$, $\mathbf{n}_{g_t}^B \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{n}_g}^2)$, and the bias terms of gyroscope and accelerometer are modeled as random walk, *i.e.*, $\dot{\mathbf{b}}_{g_t}^B = \mathbf{b}_{g_t}^B - \mathbf{b}_{g_{t-1}}^B \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{b}_g}^2)$, $\dot{\mathbf{b}}_{a_t}^B = \mathbf{b}_{a_t}^B - \mathbf{b}_{a_{t-1}}^B \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{b}_a}^2)$. Assuming noises from different sources and different time steps are independent, we can derive that the measurement errors $\mathbf{E}_t^B$ are also random walks

$$\begin{aligned} \dot{\mathbf{E}}_t^B &= \mathbf{E}_t^B - \mathbf{E}_{t-1}^B \\ &= \mathbf{b}_{t-1}^B + \dot{\mathbf{b}}_{t-1}^B + \mathbf{n}_t^B - \mathbf{b}_{t-1}^B - \mathbf{n}_{t-1}^B \\ &= \dot{\mathbf{b}}_{t-1}^B + \mathbf{n}_t^B - \mathbf{n}_{t-1}^B \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{b}}^2 + 2\boldsymbol{\sigma}_{\mathbf{n}}^2\right) \end{aligned} \quad (3)$$

We further assume, at time step 0, the initial measurement errors are from Gaussian distributions, *i.e.*, $\mathbf{E}_0^B \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{E}}^2\right)$.

C. Cost Function Terms

In this section, we describe the cost function terms of the optimization problem for state estimation. To ensure the state estimation is robust to low lighting conditions or motion blur, we propose to incorporate the following depth-based cost terms in our cost function.

1) Depth-TSDF Conformance: Given the depth map I_k^d of frame k, and the so-far constructed TSDF $\psi : \mathbb{R}^3 \to \mathbb{R}$, the depth-TSDF conformance measures how well the 3-D point cloud back-projected from I_k^d aligns to the zero-crossing surface of ψ, similar to [18] and [27]. The back-projection is based on the camera pose. Our goal is to find the camera pose $[\mathbf{q}_{C_k}^W | \mathbf{p}_{C_k}^W]$ that best *fits* the depth map I_k^d into the TSDF, where $\mathbf{q}_{C_k}^W$ and $\mathbf{p}_{C_k}^W$

are the orientation and position of camera in world coordinate, respectively.

For a pixel (i, j) and its corresponding depth value $I^d(i, j)$, we can obtain the corresponding 3-D point $\mathbf{p}_{ij}^C$ in camera coordinate system by back-projection. Then, we can transform this 3-D point to the world frame using $\mathbf{p}_{ij}^W = \mathbf{q}_C^W \otimes \mathbf{p}_{ij}^C + \mathbf{p}_C^W$. For a newly captured frame, we fit only the overlapping area observed by both the current and the previous frames to avoid overestimation of fitness [18]. We compute the overlapping area O_k of the depth frame I_k^d as

$$O_k = \left\{(i,j) \mid \pi_{k-1}(\pi_k)^{-1}[(i,j)] \in I_{k-1}^d\right\} \tag{4}$$

where the π_k is the projection matrix of frame k. Note that, in (4), we use an approximate overlapping area between the consecutive frames (π_{k-1}, π_k). Under the online tracking, the transformation between consecutive time instants is relatively small (tens of centimetres or degrees) even under fast motion, meaning that overlapping area between consecutive frames is large. In our implementation, a rough π_k can be initialized with IMU propagation and then updated with camera pose after every iteration step during the optimization.

We then seek to find the camera pose $[\mathbf{q}_{C_k}^W | \mathbf{p}_{C_k}^W]$ such that the overlapping 3-D points lie as close as possible to the zero-crossing surface $\{\mathbf{x} | \psi(\mathbf{x}) = 0\}$ of TSDF. Following the same assumption in [27], the depth measurements contain Gaussian noise and that all pixels are independent and identically distributed. Therefore, the depth-TSDF conformance can be defined as

$$\begin{aligned}
\hat{\mathbf{r}}_{\mathcal{D}}(\mathbf{x}_k) &= \sum_{(i,j)\in O^k} \psi\left(\mathbf{q}_{C_k}^W \otimes \mathbf{p}_{ij}^C + \mathbf{p}_{C_k}^W\right)^2 \\
\mathbf{q}_{C_k}^W &= \mathbf{q}_{B_k}^W \otimes \mathbf{q}_B^C \\
\mathbf{p}_{C_k}^W &= \mathbf{q}_{B_k}^W \otimes \mathbf{p}_B^C + \mathbf{p}_{B_k}^W
\end{aligned} \tag{5}$$

where the transformation between the camera and the IMU $[\mathbf{q}_B^C | \mathbf{p}_B^C]$ is fixed and known from prior calibration. Finally, we can normalize the depth-TSDF conformance by the number of valid points in the overlapping area

$$\mathbf{r}_{\mathcal{D}}(\mathbf{x}_k) = \frac{\hat{\mathbf{r}}_{\mathcal{D}}(\mathbf{x}_k)}{|O_k|}. \tag{6}$$

The main advantage of the depth-TSDF conformance [see (5)] over existing methods is that it is correspondence-free, saving the need for descriptor matching. This also makes it robust to low lighting conditions or motion blur. Besides, the depth-TSDF conformance is parallel-friendly, which could be efficiently implemented with GPU for fast candidate states evaluation.

2) IMU Measurement Residual: Sharing the same insight with the primary INS works [12], [15], we perform IMU measurement residual minimization. At the frame k, we can obtain the IMU measurements $\hat{\mathbf{a}}_{t=t_{k-1}:t_k}^B$ and $\hat{\boldsymbol{\omega}}_{t=t_{k-1}:t_k}^B$ between two consecutive depth camera frames k and $k-1$, where t_{k-1} and t_k are the corresponding time steps of camera frame $k-1$ and k in the IMU rate. We can then propagate the IMU measurements to obtain the estimated rotation $\hat{\mathbf{q}}_{B_k}^W = Q(\mathbf{x}_{k-1}, \hat{\boldsymbol{\omega}}_{t=t_{k-1}:t_k}^B, \mathbf{E}_{a_k}^B)$ and position $\hat{\mathbf{p}}_{B_k}^W = P(\mathbf{x}_{k-1}, \hat{\mathbf{a}}_{t=t_{k-1}:t_k}^B, \hat{\boldsymbol{\omega}}_{t=t_{k-1}:t_k}^B, \mathbf{E}_{a_k}^B, \mathbf{E}_{g_k}^B)$,

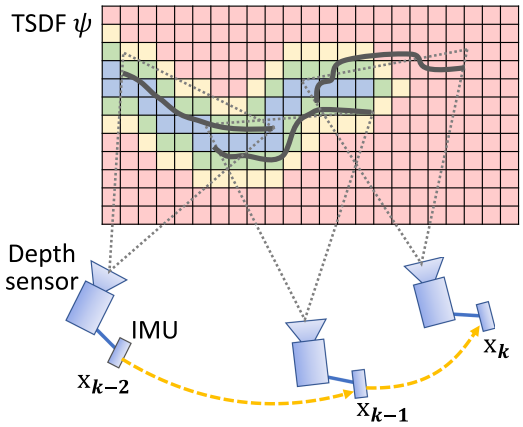

Fig. 2. Our cost function is composed of depth-TSDF conformance and the IMU measurement residual. Given a pose of the depth camera, a point cloud is back-projected from the depth map and embedded into the TSDF volume. The pose that causes the best alignment between the depth and the TSDF is the optimal solution. The transformation between the camera and the IMU is fixed.

where $P(\cdot)$ and $Q(\cdot)$ are the mid-point propagation function introduced in [15], and $\mathbf{E}_{ak}^B$ and $\mathbf{E}_{gk}^B$ are included in our state $\mathbf{x}_k$ and represent the average IMU measurement error from t_{k-1} to t_k. Assuming that the time interval between consecutive frames is nearly constant, it can be proved that both the two errors are approximative to random walk [see (3)] [50]. Therefore, we can define the IMU measurement residual as

$$\begin{aligned}
\mathbf{r}_{\mathcal{T}}(\mathbf{x}_k) &= \mathbf{p}_{B_k}^W - \hat{\mathbf{p}}_{B_k}^W \\
\mathbf{r}_{\mathcal{R}}(\mathbf{x}_k) &= \boldsymbol{\theta}\left(\mathbf{q}_{B_k}^{W\,-1} \otimes \hat{\mathbf{q}}_{B_k}^W\right)
\end{aligned} \tag{7}$$

where the $\boldsymbol{\theta}(\mathbf{q})$ extracts the rotation angle of a unit quaternion. For discrete-time implementation, we leverage the same mid-point algorithm as [15].

D. Formulation of Cost Function

The cost function (see Fig. 2) is composed of the depth-TSDF conformance (5) and the IMU measurement residual (7)

$$\begin{aligned}
\mathbf{x}_k^* = \arg\min_{\mathbf{x}_k} \{ & w_1 \mathbf{r}_{\mathcal{D}}(\mathbf{x}_k) + w_2 \mathbf{r}_{\mathcal{R}}(\mathbf{x}_{k-1}, \mathbf{x}_k) \\
& + w_3 ||\mathbf{r}_{\mathcal{T}}(\mathbf{x}_{k-1}, \mathbf{x}_k)||^2 \}
\end{aligned} \tag{8}$$

where w_1, w_2, and w_3 are the weights of each cost term. We empirically set the weights as $w_1 = 1$, $w_2 = 1$, and $w_3 = 0.1$, considering that the double-integration term $\mathbf{r}_{\mathcal{T}}$ has a relatively large error, especially at the initialization stage. Note that, the cost function does not require RGB inputs, making our method naturally resilient to the low light conditions and motion blur.

V. Active Subspace Random Optimization

To track the 6-DoF pose, we only need to solve the optimization problem in (8) at each camera frame. However, directly optimizing (8) brings three major challenges to state estimation algorithms. *First*, some states, like gravity vector and velocity, are unknown at initialization. Existing works require specially designed initialization scheme such as visual-inertial

alignment [15]. *Second*, gradient can be undefined when the depth map is out of the valid scope of TSDF under very large camera motion. *Third*, high dimensional state estimation is highly computationally consuming, especially for sampling-based algorithms. To mitigate the challenges, we propose active subspace random optimization by sampling candidate states only within active subspace. We first briefly review random optimization, then introduce PST and how to leverage them to replace random sampling, and finally describe our proposed active subspace random optimization method.

A. Random Optimization

Random optimization is a class of optimization methods, which does not require computing gradients of the objective function and thus is also known as derivative-free optimization. For a general optimization problem $\min_{\mathbf{x}\in\Omega} g(\mathbf{x})$, where $\mathbf{x}$ represents optimization variables or the states being optimized in the state space Ω and $g(\cdot)$ represents the objective function, the basic idea of random optimization is to iteratively sample around the current optimum $\mathbf{x}^*$ and move the optimum to the best found position in the state space. More specifically, in each step of the optimization, a set of candidate states $\{\mathbf{x}^{(i)}\}_{i=1:N}$ are sampled from a predefined distribution, usually Gaussian distributions, centered round the so-far reached optimum, i.e., $\mathcal{N}(x^*, \Sigma)$.

In this work, we adopt a fast and robust random optimization method introduced in [18], which we provide a brief recap below. The key idea is to presample a set of random states $\{\mathbf{x}^{(i)}_{PST}\}_{i=1:N}$ around the origin of the state space Ω, which we refer as PST, denoted by Ω^{PST}, and use them to replace true random sampling operations used in each iteration of random optimization, for saving time in performing samplings and accelerating optimization.

At each iteration step j, Zhang et al. [18] proposed to evolve Ω^{PST} in the following way: it rescales each presampled states $\mathbf{x}_{\text{PST}}$ from Ω^{PST} anisotropically using a scaling factor $\mathbf{r}_j \in \mathbb{R}^D$ ($D = 6$ for [18] but in our work $D = 18$) and then add each of them as a shift to the current optimum $\mathbf{x}^*_{j-1}$ of step $j-1$, yielding a new set of candidate states Ω_j in replacement of true random sampled states around $\mathbf{x}^*_{j-1}$

$$\Omega_j = \left\{ \mathbf{x}^{(i)}_j \,|\, \mathbf{x}^{(i)}_j = \mathbf{r}_{j-1} \odot \mathbf{x}^{(i)}_{PST} + \mathbf{x}^*_{j-1} \right\} \tag{9}$$

where $\odot$ represents element-wise multiplication. We can then evaluate all the candidate states $\{\mathbf{x}^{(i)}_j \mid \mathbf{x}^{(i)}_j \in \Omega_j\}$ using the objective function $g(\cdot)$, followed by elitist selection that keeps only the ones that have lower costs than the previous optimum and forms Ω^e_j. Finally, we perform a weighted average among the selected candidate states, where the weight is proportional to the cost margin. This procedure can be formally described as follows:

$$\begin{aligned}
\Omega^e_j &= \left\{ \mathbf{x}^{(i)}_j \in \Omega_j \,|\, g\left(\mathbf{x}^{(i)}_j\right) < g\left(\mathbf{x}^*_{j-1}\right) \right\} \\
\mathbf{w}^{(i)} &= g\left(\mathbf{x}^*_{j-1}\right) - g\left(\mathbf{x}^{(i)}_j\right) \\
\bar{\mathbf{w}}^{(i)} &= \mathbf{w}^{(i)} \Big/ \sum_{\mathbf{x}^{(i)}_j \in \Omega^e_j} \mathbf{w}^{(i)} \\
\mathbf{x}^*_j &= \sum_{\mathbf{x}^{(i)}_j \in \Omega^e_j} \bar{\mathbf{w}}^{(i)} \mathbf{x}^{(i)}_j .
\end{aligned} \tag{10}$$

Once the optimum is updated, we can easily adjust the scaling factor based on the dimensional deviation ($\mathbf{v} = \mathbf{x}^*_j - \mathbf{x}^*_{j-1}$) and objective/cost function $g(\cdot)$

$$\hat{\mathbf{r}}_j = g\left(\mathbf{x}^*_j\right) \frac{\mathbf{v}}{||\mathbf{v}||}. \tag{11}$$

Here, $\frac{\mathbf{v}}{||\mathbf{v}||}$ drives the candidate states toward the so far best state, which can be scaled by the current best value of objective function $g(\mathbf{x}^*_j)$. This means that the search range is adaptive to the performance of optimization, which helps the optimization converge more stably. In [18], $\mathbf{r}_j = \hat{\mathbf{r}}_j$. Our method proposes further adjustment for computing $\mathbf{r}_j$ to control the active subspace (see Section V-C)

By leveraging this variant of random optimization, Zhang et al. [18] showed impressive performance for low dimensional state estimation problem, *e.g.,* 6-DoF camera tracking. However, we identify two issues of [18] when applying it to our problem: 1) to generate Ω_{PST}, [18] treats all state variables as Euclidean variables and always perform uniform sampling, which is problematic for rotations and other variables introduced in our state; 2) when the dimensionality of the state space increases, the amount of sampled candidate states to cover the state space needs to increase exponentially, making this method intractable for high dimensional state estimation problem, such as in INS and this work. We will tackle these two issues separately in Section V-B and Section V-C.

B. Sampling and Evolving Ω^{PST} for DIO

For depth inertial odometry, our state $\mathbf{x} = \left[\mathbf{p}_B^{W\,T}, \mathbf{v}_B^{W\,T}, \mathbf{q}_B^{W\,T}, \mathbf{g}^{W\,T}, \mathbf{E}_a^{B\,T}, \mathbf{E}_g^{B\,T}\right]^T$ contains both Euclidean and SO(3) variables that follow a variety of distributions. To presample them and generate our Ω^{PST}, we want to make reasonable assumptions about the underlying distributions for each state variables, unlike [18] always uses uniform distribution. Our general guideline is that, since states in Ω_{PST} will be used as shift to the current state, their distributions need to resemble the distribution of the change in each state variable between two camera frames.

Here, we can categorize the state variables into following three types.

1) Unit Quaternions: The orientation $\mathbf{q}_B^W$ is a quaternion representing a rotation in $SO(3)$. Gravity vector $\mathbf{g}^W$ can also be seen as a rotation multiplies $[0, 0, g]^T (g = 9.81)$. Considering arbitrary and rapid change in camera orientation, we thus assume these quaternions follow uniform distribution in $SO(3)$. We then adopt the uniform rotation sampling algorithm proposed by [51]. And we use the positive-dot-product to check that all the quaternions are on the same hemisphere, which indicates the quaternions are continuously distributed. Note that, for these unit

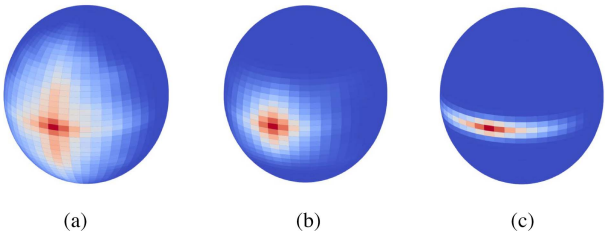

Fig. 3. Visualization of the rescaling of the state template (sphere) corresponding to rotations (represented with quaternions). The color map on the sphere depicts the Gauss map of the rotations. Initially, the rotations are perturbed with a random scaling (a). The dispersion of the rotations then converges over iteration steps (b). (c) shows anisotropic rescaling. (a) Initially scaled rotation state template. (b) After uniform scaling. (c) After nonuniform scaling.

quaternions, we only use their imaginary part $[q_x, q_y, q_z]$ as state variables.

2) Position and Velocity: Due to the same reason, we assume the change in velocity $\mathbf{v}_B^W$ and position $\mathbf{p}_B^W$ may follow a uniform distribution, we thus sample the corresponding variables from uniform distribution $\mathcal{U}([-1, 1])$.

3) IMU Measurement Errors: According to (3), the change in IMU measurement errors $\mathbf{E}_a^B, \mathbf{E}_g^B$ follow Gaussian distribution, so we presample them from $\mathcal{N}(0, \sigma_{\mathbf{E}_a} = 10^{-3})$ and $\mathcal{N}(0, \sigma_{\mathbf{E}_g} = 10^{-4})$, respectively. To ensure unbiased sampling, we use 3-D Poisson disk sampling [52].

As $\mathbf{v}_B^W, \mathbf{p}_B^W, \mathbf{E}_a^B, \mathbf{E}_g^B$ are all Euclidean variables, we can use the previous way in [18] to evolve their states (9) and update state $\mathbf{x}$ (10). Note that both uniform distribution and Gaussian distribution are closed under scaling and translation operations.

However, it is highly nontrivial to evolve and update unit quaternion variables, *i.e.*, $\mathbf{q}_B^W, \mathbf{g}^W$.

To scale a unit quaternion $\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$, we devise a scale function $\phi_{\mathbf{q}}(\mathbf{q}, \mathbf{r}_q) : \mathcal{S}^3 \times \mathbb{R}^3 \to \mathcal{S}^3$, where $\mathcal{S}^3$ is the unit hypersphere that quaternions lie in. The imaginary part of $\mathbf{q}' = \phi_{\mathbf{q},\mathbf{r}_q}$ output is simply the element-wise multiplication of the imaginary part of the quaternion and $\mathbf{r}_q$, and the real part is then computed using $\mathbf{q}_w = \sqrt{1 - \mathbf{q}_x'^2 - \mathbf{q}_y'^2 - \mathbf{q}_z'^2}$ Finally, we can evolve the quaternion variables in the following way.

$$\mathbf{q}_j^{(i)} = \phi_{\mathbf{q}}\left(\mathbf{q}_{\text{PST}}^{(i)}, \mathbf{r}_{jq}\right) \otimes \mathbf{q}_{j-1}^* \tag{12}$$

where $\mathbf{q}_{\text{PST}}^{(i)}$, $\mathbf{q}_{j-1}^*$, $\mathbf{r}_{jq}$ are the corresponding portions in $\mathbf{x}_{\text{PST}}^{(i)}$, $\mathbf{x}_{j-1}^*$, $\mathbf{r}_j$. In fact, $\mathbf{r}_q$ always stay between 0 and 1, leading to a decrease in the angle of rotation and attracting the rotation axis of $\mathbf{q}$ to $||\mathbf{r}_q||$ (see Fig. 3). Finally, to update a quaternion variable $\mathbf{q}_j$ in $\mathbf{x}_j^*$, we need to perform weighted averaging over many quaternions, which we propose to do as the following:

$$\hat{\mathbf{q}_j} = \sum_{\mathbf{x}_j^{(i)} \in \Omega_j^e} \bar{\mathbf{w}}^{(i)} \mathbf{q}_j^{(i)}$$

$$\mathbf{q}_j^* = \frac{\hat{\mathbf{q}_j}}{||\hat{\mathbf{q}_j}||}. \tag{13}$$

There are other options for quaternions averaging, like Eigendecomposition [53] or high dimensional rotation averaging [54]. However, for a relatively small rotation (tens of degrees) under the online tracking, the directly averaging (13) shows comparable performance with the fastest calculation (2.35x faster than [53] and 8.17x faster than [54]).

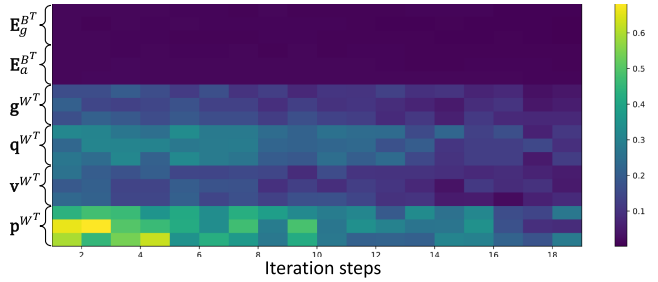

Fig. 4. Change of the sampling efficiency [see (14)] of the 18 dimensions of the solution space over increasing iterations.

C. Active Subspace

To tackle the large dimensionality (18 DoFs) with random optimization, we propose the method of active subspace. More specially, we evaluate which dimensions in the state space are most crucial for the optimization and only optimize them.

We apply the sampling efficiency from sampling theory [55] and extend it to the dimensional sampling efficiency. The idea is that within the given sampling range/size, the favourable samples should lie as further as possible to the sampling center, which means these samples introduces more posterior knowledge to the optimization. Given finite candidate sampling states $(\mathbf{x}_d^{(i)})_{d=1:D}$ over given range $\mathbf{r}_d$, the dimensional sampling efficiency $\mathbf{e}_d$ of each dimensionality can be formulated as follows:

$$\mathbf{e}_d = \frac{|\mathbf{x}_d^* - \bar{\mathbf{x}}_d|}{\mathbf{r}_d} \tag{14}$$

where the $\bar{\mathbf{x}}_d$ is the sampling center of each dimensionality (local optimum at last step) and $\mathbf{x}_d^*$ is new optimum calculated by (10).

Based on $\mathbf{e}_d$, our method selects the top-k (in implementation, we set $k = 6$) dimensions with highest $\mathbf{e}_d$ as active subspace Ω^a. The active subspace represents the most crucial subspace that can be used to replace the whole state space in random optimization. But the dimensions in active subspace may not stay *high effiency*, especially when close to a local optimal. Therefore, we still maintain a small search size for inactive dimensions and re-evaluate the $\mathbf{e}_d$ in each step for updating the active subspace. Here, we use the $\mathbf{r}_d$ to control the active subspace and inactive subspace

$$\mathbf{r}_d = \begin{cases} \hat{\mathbf{r}_d} + \epsilon, & \mathbf{x}_d \in \Omega^a \\ {\mathbf{e}_d}^2 \hat{\mathbf{r}_d} + \epsilon, & \mathbf{x}_d \notin \Omega^a \end{cases}. \tag{15}$$

The square term of the $\mathbf{e}_d$ can significantly shorten the search size of inactive subspace and the $\hat{\mathbf{r}}$ is calculated from (11). We use a small number ϵ (10^{-3}) as a lower bound of search size to avoid degenerating dimensionality. The lower bound is far smaller than camera motions, which suppresses the sampling of inactive subspace while promoting that of activate one. This helps avoiding local optima.

Fig. 4 visualizes the change of the sampling efficiency of different dimensions over increasing iteration steps. From the visualization, we can see that the dimensions corresponding to position contributes the most to the minimization of the cost function, while those corresponding to IMU measurement errors contributes the least. This means that the position part of the candidate states are initially far from the optimum. As the iteration proceeds, the sampling efficiency of all dimensions reduces to a low level, indicating convergence of optimization. In our implementation, we set the number of sampled candidate states to 3072 at each iteration for a good trade-off between accuracy and speed.

VI. Experiments

The experimental evaluation is designed and conducted to validate our key claims: first, the proposed active subspace random optimization can achieve good optimality for the high dimensional optimization problem of DIO; second, our DIO is robust to low light conditions and fast camera motions without the need of warming up initialization.

In the video of our supplementary material, we show live demo of several challenging sequences. Especially on the camera shaking sequences, our method is able to achieve perhaps the fastest camera motion tracking ever seen for odometry in indoor environments.

A. Experimental Setup

We evaluate the methods on three indoor scene benchmark, including two public available benchmarks and one new benchmark, RAS benchmark, contributed by this work. All benchmarks contain RGB, depth, and inertial measurements (though our method does not use RGB images).

Existing Benchmarks:

1) The ETH3D benchmark [28] is captured by synchronized global shutter cameras together with an Asus Xtion Live Pro. We are especially interested in three challenging sequences, namely *camera_shake_1/2/3*, which contain severely fast motions without slow-motion warm up. For a comprehensive evaluation, we also test on 12 less aggressive sequences, named in four categories including *sofa*, *table*, *einstein,* and *mannequin_face*. They are explicitly marked as slow motion sequences in ETH3D.
2) The VCU_RVI benchmark [56] contains a set of sequences captured by structure core under different environment conditions. We take three most challenging sequences, namely, lab-motion, lab-light, and bumper, which are captured under fast camera motions, low light conditions and robot-scanning, respectively.

RAS Benchmark: To thoroughly evaluate different odometry methods in the presences of fast motions, we curate a new benchmark, namely RAS. The RAS benchmark contains 24 sequences captured by Azure Kinect DK. We intentionally use a different depth sensor to the ones in the previous two datasets, which may help examine the methods against different depth sensor qualities. During capture, we fix the camera on a robot arm AUBO-i5 and move the arm following one pregenerated path, which was pregenerated by human demonstration [57]. Unlike the existing datasets, we keep the path constant as a control variable but vary the lighting conditions (normal/low light conditions) and motion speeds (fast/slow with the maximum linear speeds above or below $2\,\mathrm{m/s}$, respectively). The combination of two light conditions and two motion speeds yields four different settings, for each of which we capture six sequences to cover randomness in depth and IMU measurements. We obtain the ground truth trajectories using hand-eye calibration method [58].

Evaluation Metrics: Following the existing works [24], [33], [56], we adopt the following evaluation metrics. 1) Absolute Trajectory Error (ATE): It measures the root mean squared distance between the ground truth and estimated trajectories. If a dataset contains many sequences, we use mean absolute trajectory error (mATE) to compute the mean ATE among all the successfully tracked trajectories. 2) Success Rate (SR): It is defined as the ratio of successfully tracked sequences to total sequences and thus indicates the robustness of navigation systems under challenging conditions. In both two metrics, a tracking is successful if it satisfies both the accuracy condition (ATE$<5\,\mathrm{m}$) and the completeness condition for at least half of the sequence. 3) Relative Pose Error (RPE): It evaluates the relative pose differences between the estimated and the ground-truth motion. This metric is well-suited for evaluating local trajectory accuracy.

B. Quantitative Comparisons

1) Sequence camera_shake_1/2/3 *in ETH3D:* The three *camera_shake* sequences mainly contain rapid shaking motions and exhibit increasing moving speeds from *camera_shake_1* to *camera_shake_3*. The shaking motions could cause significant difficulties in the initialization of inertial-aided methods. We compare our method with previous state-of-the-art (SOTA) RGB-D SLAM and inertial-aided methods based on traditional optimization methods (e.g., Gaussian–Newton and Levenberg–Marquardt). Besides, we also consider removing the dense photometric term of the tightly coupled RGB-D-Inertial SLAM [32] to come up with a tightly coupled DIO baseline. However, as the authors of [32] did not release their source code and the associated data for evaluation, we tried our best to faithfully implement their method and turn it into a depth-inertial version, which we name as D-Inertial SLAM*. As shown in Table I, our method achieves successful tracking with the highest tracking accuracy on all three sequences.

Note that, all the other inertial-aided methods fail in all the three sequences. The reason is that the warm-up step of the existing inertial-aided methods are quite vulnerable to fast motion under which the input visual signals have been severely degraded by motion blur and the elevated nonlinearity in the optimization problem caused by large interframe rotations hampers their optimization. On the contrary, our method does not rely on RGB input and our random optimization demonstrates more robustness under high nonlinearity than second-order optimization methods.

Compared with ROSEFusion [18], which is the previous SOTA method on the three sequences and uses only depth, our method further incorporates IMU measurements into the optimization, taking the best of both worlds and thus leading

TABLE I
RESULTS OF ATE ON *CAMERA_SHAKE_1/2/3* SEQUENCES IN ETH3D

	Inertial	Depth	RGB	CS_1	CS_2	CS_3
BAD SLAM [28]		✓	✓	–	–	–
DVO-SLAM [59]		✓	✓	9.40	–	–
ORB-SLAM2 [60]		✓	✓	–	6.89	–
ElasticFusion [61]		✓	✓	8.44	–	–
BundleFusion [62]		✓	✓	5.17	3.49	–
DROID-SLAM [63]		✓	✓	1.20	1.92	2.69
ROSEFusion [18]		✓		0.62	1.35	4.67
VINS-Mono [15]	✓		✓	–	–	–
VINS-RGBD [24]	✓	✓	✓	–	–	–
DUI-VIO [23]	✓	✓	✓	–	–	–
D-Inertial SLAM* [32]	✓	✓		–	–	–
Ours (ASRO-DIO)	✓	✓		**0.59**	**0.98**	**2.37**

Here CS 1/2/3 indicates which sequence in *Camera shake*, ATE is measured in cm (the lower, the better), and – in the table indicates failure in tracking. D-inertial SLAM* indicates our own implementation of [32] as author's code is unavailable.

TABLE II
MEAN ATE (CM) ON 4 LESS AGGRESSIVE CATEGORIES INCLUDING *SOFA*, *TABLE*, *EINSTEIN* AND *MANNEQUIN_FACE* OF ETH3D

	sofa	table	einstein	man._face
BAD SLAM	**0.21**	**0.35**	**0.55**	**0.63**
DROID-SLAM	0.61	0.90	**0.54**	**0.39**
ROSEFusion	1.39	2.57	1.19	2.90
DUI-VIO	7.89	3.72	1.84	2.43
D-Inertial SLAM*	-	5.77	6.13	5.63
Ours (ASRO-DIO)	**0.57**	**0.83**	0.91	1.59

The best and the second best results for each category are highlighted in blue and green colors, respectively.

TABLE III
COMPARING THE SR AND ACCURACY (MEAN ATE (M)) OF FOUR METHODS ON THE RAS BENCHMARK

	VINS-Mono		VINS-RGBD		DUI-VIO		Ours (ASRO-DIO)	
	SR	ATE	SR	ATE	SR	ATE	SR	ATE
SL	83.3%	0.496	100%	0.073	**100%**	0.139	**100%**	**0.044**
FL	50.0%	3.28	83.3%	0.149	**100%**	0.411	**100%**	**0.052**
SD	0.00%		0.00%		0.00%		**100%**	**0.042**
FD	0.00%		0.00%		0.00%		**100%**	**0.053**
All	33.3%	0.154	45.83%	0.108	50%	0.275	**100%**	**0.048**

TABLE IV
COMPARISON THE SR AND ACCURACY (MEAN ATE (M)) ON THE THREE CATEGORIES OF THE VCU_RVI DATASET: BUMPER, MOTION, AND LIGHT

		Bumber	Motion	Light	All
VINS-MONO	SR	80.0%	75.0%	75.0%	76.4%
	ATE	2.472	0.388	**0.824**	1.18
VINS-RGBD	SR	90.0%	33.3%	66.7%	61.7%
	ATE	0.514	0.247	1.301	0.763
DUI-VIO	SR	40.0%	91.7%	83.3%	76.4%
	ATE	0.336	**0.154**	0.921	**0.477**
DUI-VIO-NoBA	SR	40.0%	66.7%	75.0%	61.7%
	ATE	0.782	1.344	1.195	1.173
Ours (ASRO-DIO)	SR	**100%**	**100%**	**100%**	**100%**
	ATE	**0.154**	1.420	0.866	0.852

to significant performance improvements, especially on the sequence *camera_shake_3* with the fastest camera motion.

2) Slow Camera Motion Sequences in ETH3D: We have tested our method on 12 less aggressive sequences of ETH3D in Table II. They are explicitly marked as slow-motion sequences with no warm-up initialization. Here, we only show the average ATE of each category. The detailed results of the 12 sequences are provided in the supplemental material. Compared with the top two methods (BAD SLAM [28] and DROID-SLAM [63]), our method attains comparable performance. Note that our method does not include any postprocessing of global pose optimization or loop closure, which were employed by the two methods. Our method performs better than the inertial-based methods (DUI-VIO [23] and D-Inertial SLAM* [32]), clearly demonstrating its effectiveness in incorporating inertial information without needing a warm-up initialization.

3) RAS: We evaluate our method against the SOTA inertial-aided methods, including VINS-Mono, VINS-RGBD, and DUI-VIO, on the 24 sequences of the RAS dataset under various combinations of acquisition conditions: slow motions (S), fast motions (F), normal lighting (L), and dark (low light) (D). "All" means the average results of all sequences. As shown in Table III, our method demonstrates strong robustness to low light conditions and fast motions by successfully tracking all the sequences while always maintaining the highest accuracy. Under low light conditions, our method is the only working method with perfect tracking, while all the other methods fail completely. This is because our method does not rely on RGB input by utilizing the depth-TSDF conformance as cost function, thus being naturally independent on lighting conditions. This verifies the necessity of choosing depth-only inputs under challenging lighting conditions.

Under adequate lighting, we can see the switching from slow to fast motion leads to severe performance degradation for the alternative methods: VINS-Mono and VINS-RGBD both fail more frequently and their ATEs increase by 661.3% and 104.1%, respectively; the ATE of DUI-VIO also increases 195.7%. In contrast, our method incurs only a slightly decrease in accuracy and always maintains successful tracking (low ATEs). For the persequence results, please refer to the supplementary material.

4) VCU_RVI Sequences (Lab-Light, Lab-Motion, and Bumper): To avoid the bias induced by the quality of initialization, we skip the warm-up part and start from three uniformly sampled timestamps. The average results are reported. "All" means average results of all the three sequences. Note that ATE (m) is measured only for successfully tracked sequences. The results are demonstrated in Table IV.

Our method shows robustness over the three sequences, especially on the *bumper* sequences that contain severe camera shaking when the wheeled robot moves on a bumpy ground plane. However, on the sequences of *lab-light* and *lab-motion* whose trajectories contain multiple loops, the accuracy of our method is lower than the alternative methods. The reason is due to the lack of global pose optimization or loop closure in our method and these postprocessing can be the following future works. To reveal the effect of these postprocessing, we also compare to a downgraded version of DUI-VIO which turns off the loop closure and bundle adjustment, named as DUI-VIO-NoBA. This is achieved by setting a small number of keyframes ($\mathbf{n} = 3$). Our method works significant better than DUI-VIO-NoBA in both

TABLE V
COMPARISON OF SR AND ACCURACY (MEAN RPE (M)) ON SHORT SEQUENCES SAMPLED FROM THREE VCU_RVI SEQUENCES

	Num	DUI-VIO		Ours (ASRO-DIO)	
		SR	RPE	SR	RPE
Bumper	15	93.3%	0.073	**100%**	**0.020**
Motion	18	77.8%	0.305	**100%**	**0.056**
Light	18	55.6%	0.068	**100%**	**0.048**
All	51	74.5%	0.153	**100%**	**0.037**

"Num" is the number of fragments for each category. The average results are reported on all fragments. "All" means average results over all three categories. RPE is measured only for successfully tracked sequences.

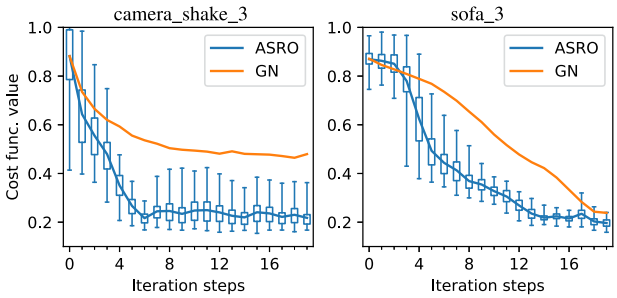

Fig. 5. Plots of cost value at different iteration steps over all frames for our ASRO and GN. For our method, we also show the range of cost value of all sampled states. The results are reported for both camera_shake_3 (fast motions) and sofa_3 (slow motions).

robustness and accuracy. Furthermore, with two keyframes, the DUI-VIO fails on all the sequences.

5) Comparison of Fast Initialization: To test the ability of fast initialization, we evaluate our method and DUI-VIO on uniformly sampled short sequence of 10 s, to see if they can quickly start up. For each short sequence, we skip the warm-up part and start from a uniformly sampled timestamp. The results are reported in Table V. Note that RPE (m) is measured only for successfully tracked sequences. Here, DUI-VIO adopts a visual-inertial alignment algorithm for system initialization, which is widely used in existing optimization methods [15], [33]. Our method, on the other hand, does not require an explicit initialization step. The results in Table V show that our method achieves better performance on the short sequences with 100% SR and thus faster initialization. More details can be found in the supplementary material.

6) Comparison With Second-Order Optimization Method: To verify the superiority of our method (ASRO) over the second-order optimization methods, we conduct a comparison based on the same cost function proposed in this work (8). To do so, we build a Gauss–Newton (GN) based method where we use second-order derivatives of depth-to-TSDF based objective, similar to [27]. The two methods are tested on two sequences, *i.e.*, camera_shake_3 (fast motion) and sofa_3 (slow motion) of the ETH3D benchmark. We compare the performance of optimization by evaluating the converging cost (8) for each frame with the same initial (center) state. To make a fair comparison, we let both methods iterate sufficiently to converge.

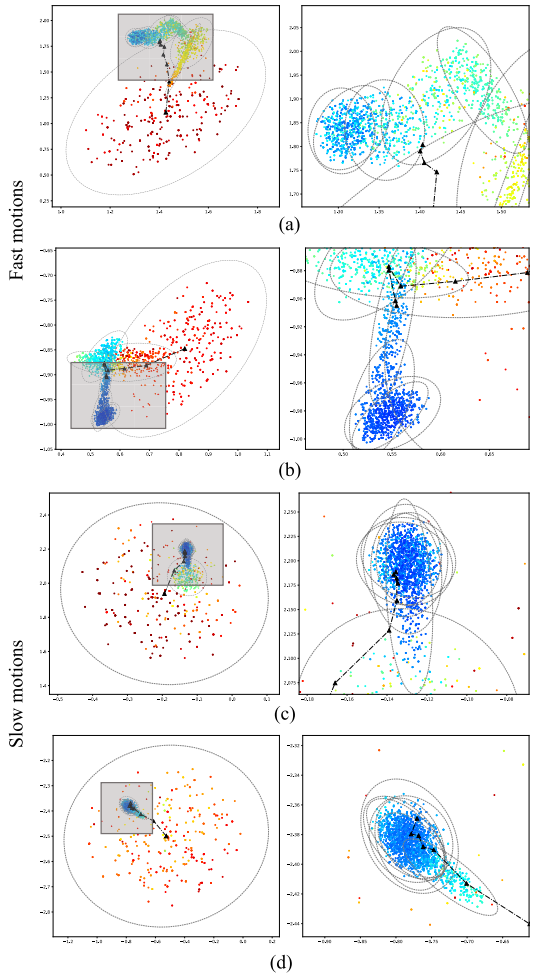

Fig. 6. 2-D visualization of optimization process of our ASRO and GN for a given frame. For our method, we plot the cost value (color-coded) of sampled states at different iteration steps of optimization. The samples of each iteration are circled with an grey ellipse. For GN, we plot the states being optimized with black triangles (connected sequentially with black line segments). The right plot in each row is a zoom-in view of the gray box in the left plot.

As shown in Fig. 5 (left, fast motion), GN quickly converges to an average cost of 0.47, which is significantly higher than 0.217 of ours. The reason is that the least square optimization finds difficulty in handling highly nonlinear situations, e.g., large rotation, as found by many previous works [18], [64]. Another reason is that, with the depth-TSDF based objective, only 68.3% of the points of a depth frame lie in the valid region of TSDF (within 20 cm from the surface) to ensure a valid gradient evaluation. The limited gradient information makes GN easily get stuck in local optima. In Fig. 5 (right, slow motion), the cost values of GN are mostly larger than the medium half of ASRO (see the slim boxes). This demonstrates that our method outperforms the GN with a significantly faster convergence on two separate sequences.

In the plots of Fig. 6, we provide a breakdown study of the performance of ASRO versus GN. The plots visualize the progressive evolution of state estimation. The states are embedded in 2-D using sparse random projection [65]. Starting with the same initial state, GN quickly gets stuck in a local optimum

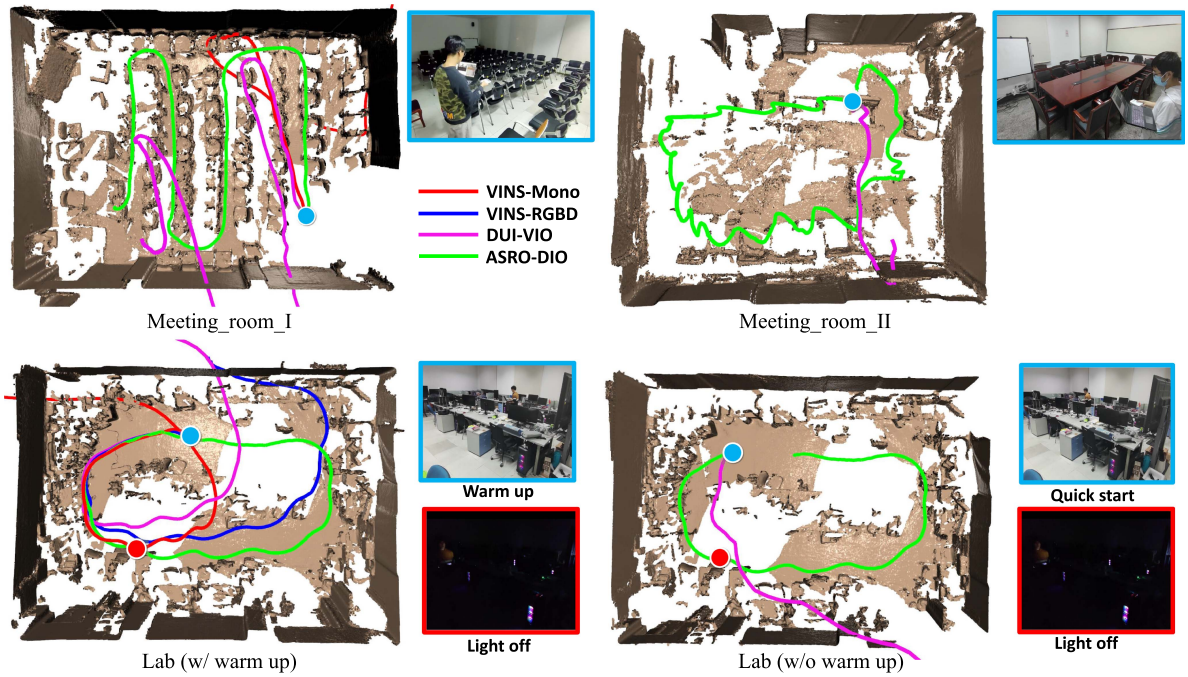

Fig. 7. Visual results of tracking and mapping on four challenging sequences. For each sequence, we compare the tracked trajectories of our method and three alternatives. The trajectory is not drawn when the method failed on a sequence completely. The 3-D reconstruction is obtained with our method, via meshing the zero-crossing surface of the TSDF.

when initialized far from the global optimum. In contrast, ASRO realizes a good trade-off between exploration and exploitation and converges to much better local minima for both slow- and fast-motion sequences.

C. Qualitative Results

1) Real Captured Sequences Under Challenging Conditions: To demonstrate the superiority of our method under low light conditions and fast camera motions without warm up initialization, we further collect the following extremely challenging trajectories using man-hold Azure Kinect DK.

a) *Lab:* One sequence was captured under fast camera motions but with slow motions at the beginning (blue point) for warm up initialization. At the half of the scanning (red point), the light was turned off [see Fig. 7 Lab (w/ warm up)]. We also get rid of the slow motions from the beginning and start the methods directly under the fast motions [see Fig. 7 Lab (w/o warm up)].

b) *Meeting_room:* two visual comparisons on challenging sequences captured under fast camera motions (see Fig. 7 *Meeting_room_I* and *Meeting_room_II*).

Here, we provide qualitative results (please refer to the video in supplementary material for more information). As shown in Fig. 7 Lab [w/ warm up)], DUI-VIO, VINS-RGBD, and VINS-Mono all fail after the light is turned OFF, due to their failure to obtain informative visual features under low light conditions. Meanwhile, in Fig. 7 Lab (w/o warm up), they also suffer from erroneous initialization under the fast motions. And even after the initialization, the output trajectory of DUI-VIO still suffer from large drift under fast motions. In contrast, our method performs well under all the challenging conditions.

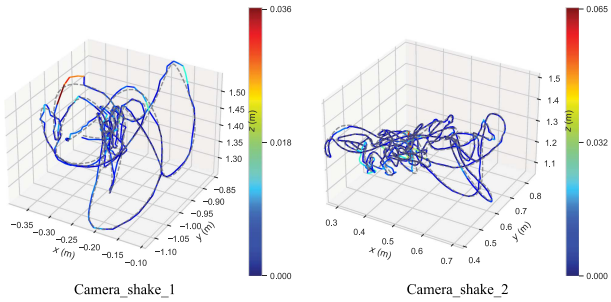

Fig. 8. 3-D plots of trajectories of our method on camera_shake_1/2 [28]. The ground-truth is indicated as the gray dash line. And the colors on our trajectories encode the relative error against ground-truth (the closer to blue, the smaller).

2) ETH3D Sequences: For a more comprehensive study, we further provide the results of our method under both the fast-motion and slow-motion sequences on ETH3D benchmark. For fast-motion sequences, we select the camera_shake_1 and camera_shake_2 for evaluation. Specifically, We plot the 3-D visual trajectories of our method against the ground-truth trajectories in Fig 8 . Here, our method obtains robust tracking results while the existing inertial-based methods failed (see Table I). And a related breakdown study is conducted in Fig. 10, where we plot the position and orientation errors along the time. From Fig. 10, the well-aligned curves demonstrate that our method can perform robust tracking under vibratory shaking motions. For slow-motion sequences, we compare the trajectories of our approach with DUI-VIO. This experiment shares the same slow-motion sequences in Table II. The results are shown in Fig. 9. Our approach demonstrates more accurate tracking performance against DUI-VIO. Moreover, our trajectories are more complete because of the fast and robust initialization at the beginning of

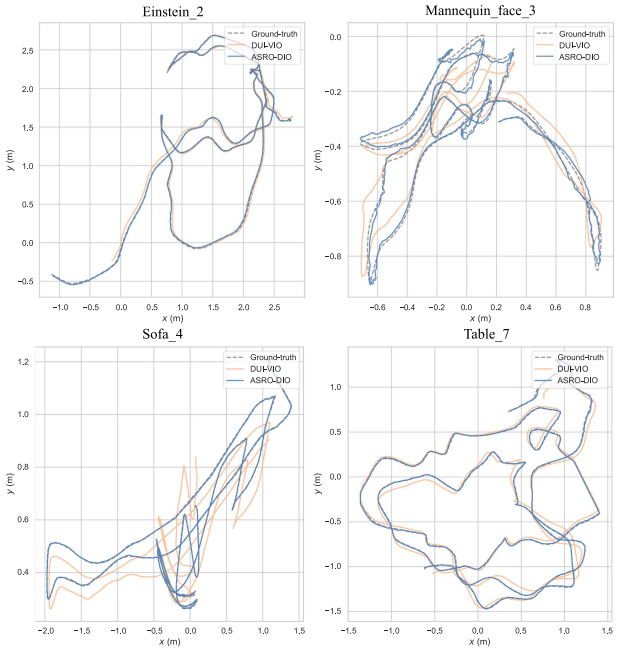

Fig. 9. 2-D plots of trajectories of our method on slow-motion sequences of ETH3D [28]. The ground-truth is indicated as the gray dash line. And the colors of our approach and DUI-VIO are encoded in orange and blue, respectively.

TABLE VI
ABLATION STUDY OF VARIOUS DESIGN CHOICES, EXPERIMENT SETTINGS AND ALGORITHMIC COMPONENTS

Method	DoFs	Gyr	Acc	AS	PS	Mean ATE
M1	$6DoFs$					5.27cm
M2	$9DoFs$	✓				4.94cm
M3	$18DoFs$	✓	✓			5.99cm
M4	$18DoFs$	✓	✓	✓		4.89cm
M5	$18DoFs$	✓	✓	✓	✓	**4.32**cm

each given sequence. More results on ETH3D can be found in the supplemental material.

D. Ablation Studies on the RAS Benchmark

In this section, we conduct ablation studies on the sequences of RAS to validate the necessity of the two key designs of our method, *i.e.*, active subspace and predefined sampling space (see Section V-B). In Table VI, we add the designs to the basic random optimization framework (see Section V-A), including the use of gyroscope readings (Gyr) and accelerometer readings (Acc), and the utilization of active subspace (AS) and predefined space (PS), tested for different DoFs of search space. All methods share the same parameter setting, including the number of sampling states (3072), the initial scaling factor (following ROSEFusion [18]) and the maximum iteration times (20). M1, M2, and M3 share the same random optimization framework of ROSEFusion performing uniform sampling for all dimensions. Comparing M1 with M2, we notice that simply leveraging gyroscope measurements and thus incorporating $\mathbf{E}_g^B$ into the state space, forming a 9-DoF optimization, could lead to higher accuracy. Here, the gyroscope provides a rough prior of the change in rotation between consecutive frames, which can benefit the optimization at the cost of a slight increase of the dimensionality of state space. However, when we further

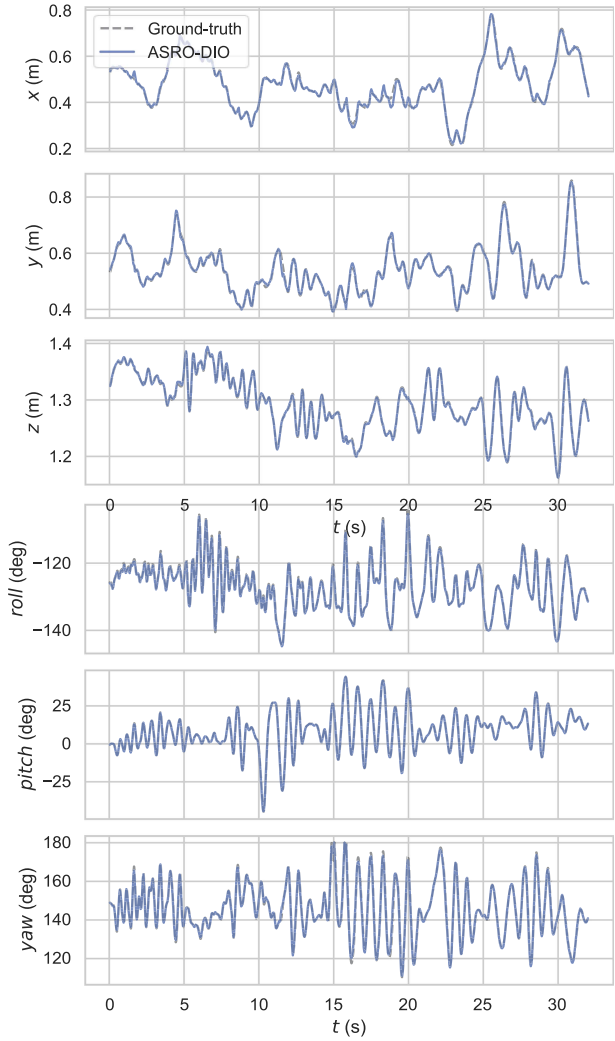

Fig. 10. Plots of position and orientation errors of our method versus ground-truth on camera_shake_2.

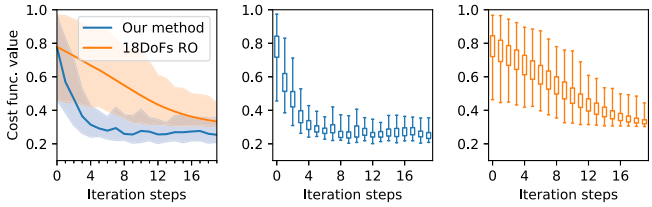

Fig. 11. Plots of the average and the range of cost function values at different iteration steps for our method (blue) and random optimization in 18-DoF state space (orange).

introduce accelerometer measurements, as in M3, the state space becomes 18D, immediately causing a significant performance drop, with an accuracy even worse than M1. By introducing active subspace in M4, the accuracy is greatly improved and is further boosted by utilizing predefined state space (M5).

In Fig. 11, we plot the cost function values (IV-D) of all frames and all sequences at different iteration steps with M3 and M5. For each method, we plot both the average and the range of cost values of all states (left) and the individual boxplot (middle and right). In accord with the result in Fig. 11, we notice that our method achieves fast convergence with higher confidence. And

compared to the smooth convergence of M3, our method shows the *rugged* convergence which could be caused by the active subspace update.

E. Runtime

We have implemented our core algorithm based on C++ and CUDA. Both the main optimization pipeline and the volumetric fusion run on a workstation with an Intel Core™ i7-5930 K CPU @ 3.50 GHz × 12 with 32 G RAM and an Nvidia GeForce RTX 2080 SUPER GPU with 8 G memory. To enable flexible scanning, we implemented a front-end program running on a laptop for RGB-D-inertial measurements capturing, compressing, and streaming to the workstation via WiFi. Our pipeline runs with a framerate of 30 Hz for all shown test sequences. The readers are welcome to watch our accompanying video.

VII. Conclusion

We have presented an efficient and robust DIO method based on random optimization with active subspace. Through identifying active subspace and sampling candidate states only within it, our method is able to explore the 18D state space of DIO efficiently and achieve good optimality under extremely challenging conditions. Although, we adopt a depth-TSDF-based cost function based on which fitness evaluation is parallel-friendly, our method is general and adaptable to other settings. In the future, we would like to investigate the integration of visual features especially when the camera motion slows down, to ensure high tracking accuracy at all time, thus enjoying the strength of both worlds. We would also like to exploit random optimization in back-end optimizations such as bundle adjustment, pose graph optimization. Finally, our method could be further improved by more efficient sampling strategies like CMA-ES [66].

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 9572–9582.
[2] D. Scaramuzza and Z. Zhang, "Visual-inertial odometry of aerial robots," 2019, *arXiv:1906.03289*.
[3] J. Kim and S. Sukkarieh, "Real-time implementation of airborne inertial-SLAM," *Robot. Auton. Syst.*, vol. 55, no. 1, pp. 62–71, 2007.
[4] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565–3572.
[5] M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne SLAM," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 261–280, Jan. 2008.
[6] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 6319–6326.
[7] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 264–280, Apr. 2009.
[8] M. Brossard, S. Bonnabel, and A. Barrau, "Invariant Kalman filtering for visual inertial SLAM," in *Proc. 21st Int. Conf. Inf. Fusion*, 2018, pp. 2021–2028.
[9] S. Ebcin and M. Veth, "Tightly-coupled image-aided inertial navigation using the unscented Kalman filter," in *Proc. 20th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2007, pp. 1851–1860.
[10] D. Strelow and S. Singh, "Motion estimation from image and inertial measurements," *Int. J. Robot. Res.*, vol. 23, no. 12, pp. 1157–1195, 2004.
[11] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. Auton. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.
[12] 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, Feb. 2017.
[13] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual–inertial navigation," *Int. J. Robot. Res.*, vol. 38, no. 5, pp. 563–586, 2019.
[14] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[15] 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.
[16] B. M. Bell and F. W. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.
[17] Y. Akimoto, A. Auger, and N. Hansen, "CMA-ES and advanced adaptation mechanisms," in *Proc. Genet. Evol. Computation Conf. Companion*, 2016, pp. 533–562.
[18] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Rosefusion: Random optimization for online dense reconstruction under fast camera motion," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–17, 2021.
[19] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas, "Bayesian optimization in high dimensions via random embeddings," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1778–1784.
[20] R. M. Gower, D. Kovalev, F. Lieder, and P. Richtárik, "RSN: Randomized subspace newton," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 616–625.
[21] J. Lacotte, M. Pilanci, and M. Pavone, "High-dimensional optimization in adaptive random subspaces," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10847–10857.
[22] K. M. Choromanski, A. Pacchiano, J. Parker-Holder, Y. Tang, and V. Sindhwani, "From complexity to simplicity: Adaptive ES-active subspaces for blackbox optimization," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 10299–10309, 2019.
[23] H. Zhang and C. Ye, "DUI-VIO: Depth uncertainty incorporated visual inertial odometry based on an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5002–5008.
[24] Z. Shan, R. Li, and S. Schwertfeger, "RGBD-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, 2019, Art. no. 2251.
[25] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 303–312.
[26] R. A. Newcombe et al., "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
[27] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time camera tracking and 3D reconstruction using signed distance functions," in *Proc. Robotics, Sci. Syst.*, Berlin, Germany, Jun. 2013, doi: 10.15607/RSS.2013.IX.035.
[28] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 134–144.
[29] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 2935–2942.
[30] N. Brunetto, S. Salti, N. Fioraio, T. Cavallari, and L. Stefano, "Fusion of inertial and visual measurements for RGB-D SLAM on mobile devices," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 1–9.
[31] Z. Zhu et al., "Real-time indoor scene reconstruction with RGBD and inertial input," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2019, pp. 7–12.
[32] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense RGB-D-inertial slam with map deformations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6741–6748.
[33] A. Tyagi, Y. Liang, S. Wang, and D. Bai, "DVIO: Depth-aided visual inertial odometry for rgbd sensors," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2021, pp. 193–201.
[34] M. Nießner, A. Dai, and M. Fisher, "Combining inertial navigation and ICP for real-time 3D surface reconstruction," in *Proc. Eurographics* (Short Papers), 2014, pp. 13–16.

[35] M. Camurri, S. Bazeille, D. G. Caldwell, and C. Semini, "Real-time depth and inertial fusion for local SLAM on dynamic legged robots," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2015, pp. 259–264.
[36] C. Andrieu and A. Doucet, "Particle filtering for partially observed gaussian state space models," *J. Roy. Stat. Society: Ser. B. (Statistical Methodol.)*, vol. 64, no. 4, pp. 827–836, 2002.
[37] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 498–519, 2012.
[38] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá, "Multi-robot visual slam using a Rao-Blackwellized particle filter," *Robot. Auton. Syst.*, vol. 58, no. 1, pp. 68–80, 2010.
[39] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
[40] E. Arnaud and E. Mémin, "An efficient Rao-Blackwellized particle filter for object tracking," in *Proc. IEEE Int. Conf. Image Process.*, 2005, vol. 2, pp. II–426.
[41] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao–Blackwellized particle filter for 6-D object pose tracking," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1328–1342, Oct. 2021.
[42] M. Nieto, A. Cortés, O. Otaegui, J. Arróspide, and L. Salgado, "Real-time lane tracking using Rao-Blackwellized particle filter," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 179–191, 2016.
[43] G. Huang, "Particle filtering with analytically guided sampling," *Adv. Robot.*, vol. 31, no. 17, pp. 932–945, 2017.
[44] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," *Evol. Computation*, vol. 25, no. 1, pp. 1–54, 2017.
[45] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.
[46] J. Matyas et al., "Random optimization," *Automat. Remote Control*, vol. 26, no. 2, pp. 246–253, 1965.
[47] P. G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. Philadelphia, PA, USA: SIAM, 2015.
[48] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 10, pp. 281–305, 2012.
[49] S. S. Vempala, *The Random Projection Method*, vol. 65, Providence, RI, USA: American Mathematical Society, 2005.
[50] F. Spitzer, *Principles of Random Walk*. Berlin, Germany: Springer, 2001, vol. 34.
[51] K. Shoemake, "Uniform Random Rotations," in *Graphics Gems III (IBM Version)*. Amsterdam, Netherlands: Elsevier, 1992, pp. 124–132.
[52] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions," in *ACM SIGGRAPH Sketches*, San Diego, CA, USA, 2007, p. 22–es.
[53] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *J. Guidance, Control, Dyn.*, vol. 30, no. 4, pp. 1193–1197, 2007.
[54] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5745–5753.
[55] D. K. Nassiuma, *Survey Sampling: Theory and Methods*. Nairobi, Kenya: Nairobi Univ. Press, 2001.
[56] H. Zhang, L. Jin, and C. Ye, "The VCU-RVI benchmark: Evaluating visual inertial odometry for indoor navigation applications with an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6209–6214.
[57] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Berlin, Heidelberg, Germany: Springer, 2008, pp. 1371–1394.
[58] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.
[59] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 2100–2106.
[60] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[61] T. Whelan, S. Leutenegger, R. Salas Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proc. Robot.: Sci. Syst.*, Rome, Italy, Jul. 2015, doi: 10.15607/RSS.2015.XI.001.
[62] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, Aug. 2017, Art. no. 76a.
[63] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Proc. Adv. Neural Inf. Process. Syst.*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, Eds., 2021, vol. 34, pp. 16558–16569. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/89fcd07f20b6785b92134bd6c1d0fa42-Paper.pdf
[64] J. Huang et al., "Subspace gradient domain mesh deformation," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 1126–1134.
[65] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. data mining*, 2006, pp. 287–296.
[66] N. Hansen, "The CMA evolution strategy: A tutorial," 2016, *arXiv:1604.00772*.

Jiazhao Zhang received the B.Eng. degree in software engineering from Shandong University, Jinan, China, in 2019, and the M.S. degree in computer science from National University of Defense technology, Changsha, China in 2021. He is currently working toward the Ph.D. degree in computer science with Peking University, Beijing, China.

His current research interests include state estimation, robust odometry systems and 3-D vision.

Yijie Tang received the B.Eng. degree in software engineering from Wuhan University, Wuhan, China, in 2019. He is currently working toward the M.S. degree in computer science with National University of Defense Technology, Changsha, China.

His research interests include online 3-D reconstruction, simultaneous localization, and mapping.

He Wang (Member, IEEE) received the bachelor's degree in microelectronics and nanoelectronics from Tsinghua University, Beijing, China, in 2014, the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2021, under the advisory of Prof. Leonidas J. Guibas.

He is a tenure-track Assistant Professor with the Center on Frontiers of Computing Studies, Peking University, where he founded and leads Embodied Perception and InteraCtion Lab. His research interests span 3-D vision, robotics, and machine learning, with a special focus on embodied AI. He has authored or coauthored more than 20 papers on top vision and learning conferences (CVPR/ICCV/ECCV/NeurIPS) with 8 of his works receiving CVPR/ICCV orals and one work receiving Eurographics 2019 best paper honorable mention.

Prof. Wang was an Area Chair in CVPR 2022 and WACV 2022.

Kai Xu (Senior Member, IEEE) received the Ph.D. degree in computer science from National University of Defense Technology, Changsha, China, in 2011.

He is currently a Professor with the School of Computer, National University of Defense Technology. He is an Adjunct Professor with Simon Fraser University, Burnaby, BC, Canada. During 2017–2018, he was a Visiting Research Scientist with Princeton University, Princeton, NJ, USA. He has authored or coauthored more than 100 research papers, including 27 SIGGRAPH/TOG papers. His research interests include geometric modeling and shape analysis, especially on data-driven approaches to the problems in those directions, as well as 3-D vision and robotic applications.

Dr. Xu serves on the editorial board of *ACM Transactions on Graphics, Computer Graphics Forum, Computers & Graphics, and The Visual Computer*. He also served as program co-chair and PC member for several prestigious conferences. He has co-organized several SIGGRAPH courses, Eurographics STAR tutorials and CVPR workshops