Rapid and Robust Monocular Visual-Inertial Initialization with Gravity Estimation via Vertical Edges

Jinyu Li, Hujun Bao, and Guofeng Zhang*

Abstract— **Monocular visual-inertial tracking without good initialization easily fails due to its non-linear nature. Rapid and accurate metric initialization is crucial. In this paper, we propose a novel monocular visual-inertial initialization method which can initialize the IMU states, camera poses, and scale in a rapid and robust way. To avoid mixing gravity and accelerometer bias, we propose to use the detected vertical edges to estimate a better gravity. This improves the observability to the underlying problem even without sufficient movement, so we can solve all the states crucial for a good initialization. We evaluate our approach on EuRoC dataset and compare with existing state-of-the-art methods. The experimental results demonstrate the effectiveness of the proposed method.**

I. Introduction

Motion tracking by the fusion of monocular visual and inertial measurements is a trending topic in robotics and computer vision community and has found tremendous opportunities in the market. Despite the recent developments in monocular visual-inertial tracking, initializing such a tracking system is not trivial. In a monocular setting, visual and inertial measurements are complement to each other: one gives 3D structure without scale, another provides scale information but can drift. These two must be initialized such that the scale and orientation are consistent with each other, otherwise the tracking will diverge. One way of initialization is to begin with a set of neutral state and wait for the algorithm to converge. However, many applications require a rapid and robust initialization. For example, in mobile augmented reality (AR), users generally expect that they can immediately watch the AR effect without waiting too much or initializing manually. The challenge comes from the coupled nature of IMU-related states, such as scale, gravity, velocity, and bias. It turns out that these states cannot be recovered altogether solely from IMU measurements [1].

We would then ask if the information outside of the IMU can be leveraged to address this problem. Visual measurements can provide complementary information. However, for the fusion of monocular visual and inertial information, it usually requires sufficient motion to fully excite all observable states. This means the convergence of initialization is slow. A second camera or a depth sensor will provide additional scale information, improving the overall observability of the system. However, monocular camera is still the most popular choice for mobile phones, drones, and many other devices. Hence the rapid and robust initialization for monocular visual-inertial tracking is quite useful for many applications.

As demonstrated in recent segment-based SLAM systems [2][3], edge information can help initialization. Specifically, in human-made scenes, vertical edges are good references for the gravity direction. Based on this observation, we propose a novel approach for visual-inertial initialization. By aligning the gravity direction with vertical edges in images, our approach can better separate the gravity from IMU measurements, leading to a rapid and robust initialization. Accurate initialization can be achieved as long as 8 frames with sufficient motion parallax, which is much smaller than existing methods. The success rate is also much higher than state-of-the-art methods. The source code of our implementation is available at https://github.com/zju3dv/vig-init.

II. Related Works

There has been a wide spectrum of monocular visual-inertial motion tracking systems. Traditional visual tracking methods use feature points, like [4], [5], [6]. In recent years, some direct-methods (e.g. [7]) are proposed, where the visual measurements are directly based on photometric error of tracked dense or semi-dense pixels. Some systems are proposed to track through a higher level of abstraction, for example [2] and [3] use edges for tracking. These systems work rather well in a human-made scene, where the scene might be textureless. In addition, edges can better regularize orientation estimation, hence can reduce drift errors.

While these well-known frameworks have achieved higher performance in accuracy, robustness or speed, few of them have well addressed the problem of system initialization, due to the challenges in dealing with the lack of observability.

From a theoretical aspect, [1], [8], [9] systematically analyzed the observability properties of visual-inertial tracking. A monocular visual-inertial tracking system has 4 unobservable dimensions, the 3D position and the yaw orientation globally, when the movement is sufficient. However, this movement requirement is usually too strict for initialization. As we will explain later, during the initialization, the accelerometer bias is hard to be distinguished from gravity, leading to initialization error.

There are works dedicated to recovering part of the initial conditions for the visual-inertial tracking system, i.e., [10]

*Corresponding author: zhangguofeng@cad.zju.edu.cn

The authors are affiliated with State Key Lab of CAD&CG, Zhejiang University, and ZJU-SenseTime Joint Lab of 3D Vision.

This work was partially supported by the National Key Research and Development Program of China (No. 2016YFB1001501), NSF of China (Nos. 61822310 and 61672457), and the Fundamental Research Funds for the Central Universities (No. 2018FZA5011).

and [11] proposed methods for gyroscope bias estimation, and also pointed out that the gravity and accelerometer bias are difficult to be perfectly distinguished if without rich motion. Recent works [12], [13], [14] all try to solve scale during initialization, all of which work in a decoupled way: initializing visual part first then performing an alignment. This alignment process can be motion-demanding, hence [12], [13] require a long-enough motion, and [14] falls back to a fail-and-reset strategy.

However, without utilizing more information from other than the IMU measurements, it is difficult to recover all initial conditions for the visual-inertial tracking system. Surprisingly, up to our knowledge, there is little work to further leverage the visual information.

III. Our Approach

Almost all VI-SLAM systems need an initialization stage to initialize the IMU states, camera poses and 3D landmarks. Figure 1 gives an overview of our initialization approach. In the first step, we employ a Structure-from-Motion (SfM) method to recover the camera trajectory and sparse 3D points. We also perform IMU preintegration with the input IMU measurements. Then we align the recovered camera trajectory with the preintegrated IMU measurements by solving the global scale and gravity. The estimated gravity is further refined with the detected vertical edges. Then with the refined gravity, the scale and biases can be further optimized by re-alignment and bundle adjustment (BA). In the following, we first briefly introduce the IMU preintegration and BA.

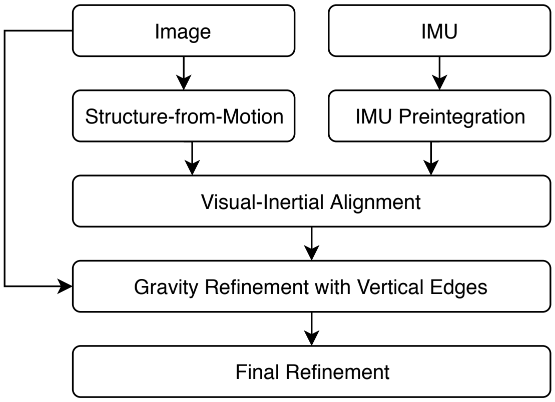


Image
IMU
Structure-from-Motion
IMU Preintegration
Visual-Inertial Alignment
Gravity Refinement with Vertical Edges
Final Refinement

Fig. 1. Pipeline of our initialization method.

A. IMU Preintegration

Forster *et al*. [15] gives a very good introduction to the IMU measurement model. An IMU usually contains a gyroscope and an accelerometer, which measure the rotation rate and acceleration of the device correspondingly. Formally, both gyroscope and accelerometer are modeled as measurements polluted by two error sources: a white noise and a random-walk bias:

$$\begin{aligned} \omega &= \hat{\omega} + \sigma_g + b_g, \\ a &= C^\top(q)(\hat{a} - g) + \sigma_a + b_a. \end{aligned} \tag{1}$$

Here, $C(q)$ denotes the rotation matrix from unit quaternion q, which is used to represent orientations of sensors. g is the gravity in world system. ω and $\hat{\omega}$ are gyroscope measurement and true rotation rate, respectively. a is the accelerometer measurement and $\hat{a}$ is the true acceleration in the world frame. The two measurements are expressed in IMU's local frame. σ_g and σ_a are white measurement noises, and b_g and b_a are time-varying biases. A reader may refer to [15] for details of the random-walk model of IMU biases, as well as their units. Practically, the covariance parameters for these four error sources can be calibrated with off-the-shelf tools like Kalibr [16]. The gravity acceleration might be slightly varied for different locations. But it can be determined given the current location, and can be regarded as constant during the short initialization period.

As we can see from (1), the accelerometer measurement depends on $C(q)$, creating difficulties when one tries to figure out the true acceleration $\hat{a}$ from a. For two consecutive frames of time i and j, we denote ω_k, a_k, $k = i...j-1$ be the IMU measurements between these two consecutive frames. We can simply perform the IMU integration from i to j to get the prediction of IMU state at time j, which has an intricate dependence on the IMU state at time i. To better represent the incremental update between two time-points, we employ the IMU preintegration technique introduced by [15]:

$$\begin{aligned} E[q_j] &= q_i \cdot \Delta q_{ij}(b_g) \cdot \exp\left(\frac{\partial \Delta q_{ij}}{\partial \delta b_g} \delta b_g\right), \\ E[v_j] &= v_i + g\Delta t_{ij} \\ &\quad + C(q_i)\left(\Delta v_{ij}(b_a, b_g) + \frac{\partial \Delta v_{ij}}{\partial \delta b_a}\delta b_a + \frac{\partial \Delta v_{ij}}{\partial \delta b_g}\delta b_g\right), \\ E[p_j] &= p_i + v_i\Delta t_{ij} + \frac{1}{2} g \Delta t_{ij}^2 \\ &\quad + C(q_i)\left(\Delta p_{ij}(b_a, b_g) + \frac{\partial \Delta p_{ij}}{\partial \delta b_a}\delta b_a + \frac{\partial \Delta p_{ij}}{\partial \delta b_g}\delta b_g\right), \end{aligned} \tag{2}$$

where $E[\cdot]$ represents the expectation, and q_i, v_i, p_i are the orientation, velocity and position of IMU at time i respectively, expressed in the world frame. $\Delta q_{ij}(\cdot)$, $\Delta v_{ij}(\cdot)$ and $\Delta p_{ij}(\cdot)$ are the preintegrated IMU measurements, expressed with respect to IMU's local frame. And the 5 Jacobians represent the first order bias updates when small perturbations are applied to b_a and b_g. We will omit the bias parameter of delta-states when it is clear from context. The complete discussion of preintegration is beyond the scope of this paper. Please refer to [17] and [15] for more details.

B. Bundle Adjustment

Bundle adjustment (BA) refers to the optimization of states, including poses, landmark locations and so on. BA has long been used in SfM and visual SLAM. A very good overview of BA can be found at [18]. Typical BA energy function in VI-SLAM can be formulated as follows:

$$\begin{aligned} \mathop{\arg\min}_{\{q_i, p_i\}, \{x_k\}} & \sum_i \sum_k \| u_{ik} - h(q_i, p_i, x_k) \|_\Omega^2 \\ & + \sum_i \mathcal{E}_{\mathrm{IMU}}(p_i, q_i, p_{i+1}, q_{i+1}) \end{aligned} \tag{3}$$

For SfM, only reprojection errors participate in BA. However, the metric scale cannot be recovered. The inertial measurements are independent of visual information and contain scale information, which can greatly benefit SLAM. So many VI-SLAM methods [2][4][5][6][19] are proposed to combine the reprojection error and IMU error terms to achieve robust tracking. To bound the computation cost, a sliding window BA is often used.

IV. Visual-Inertial Initialization

Our initialization method contains four steps. We first initialize camera poses by SfM without scale information. Then by combining IMU measurements, we estimate the global scale and gravity. We also use the detected vertical edges to better estimate gravity direction. The scale and biases can be further refined. A final bundle adjustment is employed to refine all the variables except scale and gravity.

A. Visual Initialization with SfM

For each incoming frame, we extract keypoints and use optical flow method to track them in the consecutive frames [20]. When sufficient motion parallax is detected, we begin to initialize the structure and camera poses. For robust initialization, we require the number of collected frames should be not smaller than a threshold $N_{\min}$. In order to achieve this objective, we keep a queue of images. In the beginning, the image queue waits to be filled. When there are at least $N_{\min}$ images, it begins to check the motion parallax between the latest image and each old ones until sufficient motion parallax is found. If no sufficient motion parallax is found, wait for the next incoming image and continue detection. We also set the maximum size $N_{\max}$ of the queue, so that the old images will eventually be popped out from the queue. In our experiments, we set $N_{\min} = 8$ and $N_{\max} = 22$.

If a frame pair (I_i, I_j) with sufficient parallax is found, we can compute their relative pose by five-point algorithm [21] and then triangulate the matched keypoints. Then with recovered 3D landmarks, we can estimate the camera poses between (I_i, I_j) through PnP algorithm [22]. Finally, we employ bundle adjustment to further refine the camera poses and 3D landmarks.

B. Visual-Inertial Alignment

With the recovered camera trajectory, we can solve a global scale and gravity by aligning it with the preintegrated IMU measurements. Our alignment method is similar to [14] and [13], which can be divided into two parts:

1) Gyroscope Bias Estimation: From (2), the estimation of b_g can be completely isolated from other states. We followed the common approach as in [14] and [13]. Given the pose of two images i and j, we first obtain their corresponding IMU orientations q_i and q_j. We also compute the preintegration between i and j, assuming $b_g^{(0)} = 0$ and $b_a^{(0)} = 0$. The optimal estimation of b_g can be solved by minimizing the following energy function:

$$\arg\min_{b_g} \| \log((q_i \cdot \Delta q_{ij}(b_g))^{-1} \cdot q_j) \|^2. \tag{4}$$

It is reasonable to assume that b_g is constant in a short time. We can accumulate all pairwise relations between N frames, and further linearize it by applying the lift-solve-retract approach [23]:

$$\begin{aligned} &\arg\min_{\delta b_g} \sum_{i=1}^{N-1} \left\| \log((q_i \cdot \Delta q_{i,i+1})^{-1} \cdot q_{i+1}) - \frac{\partial \Delta q_{i,i+1}}{\partial \delta b_g} \delta b_g \right\|^2, \\ &b_g^{(n+1)} \leftarrow b_g^{(n)} + \delta b_g. \end{aligned} \tag{5}$$

In practice, one iteration is enough for solving (5), which can be efficiently converted into solving a 3×3 linear equation.

2) Scale and Gravity Estimation: Upon getting gyroscope bias initialized, we update the preintegrations. We examine the other two equations in (2). Given the IMU states of two consecutive images i, j, we can have following alignment equation:

$$\begin{cases} & v_j & -v_i & -g\Delta t_{ij} & = C(q_i)\Delta v_{ij} \\ s(p_j - p_i) & & -v_i \Delta t_{ij} & -\frac{1}{2} g \Delta t_{ij}^2 & = C(q_i)\Delta p_{ij} \end{cases} \tag{6}$$

where s is the scale we wish to find. In [14], they gather N ($N \geq 4$) frames to build a $(6N-6) \times (4+3N)$ linear system of (6) for solving scale, gravity as well as velocity. We turn to use the method described in [13]. By introducing a third frame k and eliminate v_i and v_j, one gets following equations relating IMU poses of three frames:

$$\begin{aligned} \begin{bmatrix} S(i,j,k) & G(i,j,k) \end{bmatrix} \begin{pmatrix} s \\ g \end{pmatrix} &= D(i,j,k), \\ S(i,j,k) &= (p_k - p_j)\Delta t_{ij} - (p_j - p_i)\Delta t_{jk}, \\ G(i,j,k) &= -\frac{1}{2}\Delta t_{ij}\Delta t_{jk}(\Delta t_{ij} + \Delta t_{jk}), \\ D(i,j,k) &= C(q_j)\Delta p_{jk}\Delta t_{ij} - C(q_i)\Delta p_{ij}\Delta t_{jk}, \\ &+ C(q_i)\Delta v_{ij}\Delta t_{ij}\Delta t_{jk}. \end{aligned} \tag{7}$$

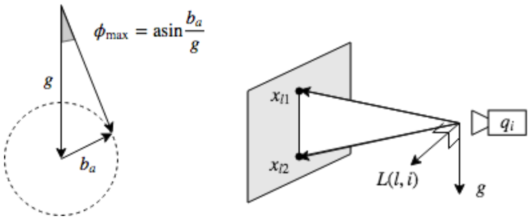
For an N-frame initialization window, we can simply take consecutive frames to construct (7). In this way, we can get $N-2$ equations. However, nearby frames may not contain sufficient motion. Instead, we enumerate all 3-frame groups and construct (7). Although its complexity is $O(N^3)$, for a small N, the resulting system can be easily solved. Also, by constructing a normal equation, only constant memory is required. By enumerating all groups, we can make sure that the maximum possible motion gets involved in our equations.

C. Gravity Refinement with Vertical Edges

We propose a novel algorithm to refine the gravity direction with the detected vertical edges. Many works have analyzed the coupling relation of true acceleration $\hat{a}$, gravity g and accelerometer bias b_a. Better true acceleration can result in better scale estimation. However, it is difficult to separate the three quantities perfectly. Especially, imperfect estimation of gravity direction will significantly influence the estimate of scale.

Since the performance of a tracking system is not sensitive to initial error in b_a [11], it is generally safe to set $b_a^{(0)} = 0$ during initialization. However, this operation will make b_a mixed with either gravity or body acceleration.

Gravity acceleration has a magnitude of approximately $9.81\mathrm{m/s}^2$. A consumer-level IMU, e.g. Bosch BMI-160, has bias around $\pm0.4\mathrm{m/s}^2$ from its datasheet[1]. The EuRoC dataset's maximum accelerometer-bias is $0.55\mathrm{m/s}^2$. If such bias got fully mixed into gravity, it will introduce a 3.24-degree error to gravity direction, as illustrated in Figure 2(a)



(a) (b)

Fig. 2. (a) When mixed in gravity, b_a can perturb gravity direction by a small angle, the maximum angle is given by $\phi_{\max}$. (b) Geometry relationship between the gravity direction and a vertical edge on an image.

In addition, such bias may get mixed with body acceleration and introduce error to preintegration. As we can see from the previous step, the error in preintegration will lead to inaccurate scale estimation.

Some methods [14], [13] proposed to fix the magnitude of gravity and refine the remaining states. Bias is still mixed in either gravity or body acceleration, or both. As a compromise, Qin *et al.* [14] requires that the motion should be large and long enough to make the system fully observable to estimate scale. This requirement is not user-friendly and maybe impractical in many applications. In [14], a quick fail-and-reset strategy is proposed to initialize the system regardless of the quality. If tracking fails, the system is reset and reinitialized. They found that a less-than-30% scale error would initialize the system successfully. So the initialization depends on the rate of success (i.e. the chances that the relative error of scale is smaller than 30%). If the rate of success is small, the total time of initialization may be long.

To shorten the initialization time, we must increase the success rate and use as less frames as possible. We found that if the gravity is known, the system will become observable, and accelerometer bias can be isolated to improve scale estimation. There is no easy way to directly get gravity component from IMU measurement. However, in many scenarios, images may contain edges which are parallel to the gravity direction. Human-made structure tends to have vertical boundaries, makes them a very good sign of gravity. So we can use these vertical edges to estimate gravity direction.

Given an input frame i, we first extract line segments by Line Segment Detector algorithm [24]. Each line segment l is parameterized with the coordinate of its two ends. Their 2D coordinates on the image projection plane are denoted as $\mathbf{x}_{l1}$, $\mathbf{x}_{l2}$. The orientation of image camera is denoted as q_i^{IMG}. We define

$$L(l,i) \equiv C(q_i^{\mathrm{IMG}}) \left(\begin{pmatrix} \mathbf{x}_{l1} \\ 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{x}_{l2} \\ 1 \end{pmatrix} \right), \tag{8}$$

which is a normal vector for the 3D plane passing through the camera center and the segment l, expressed in the world frame. The angle between the plane and gravity is formulated as

$$\sin(\theta) = \frac{L(l,i)}{\|L\|} \cdot \frac{g}{\|g\|}. \tag{9}$$

Since we already have a rough estimation of gravity, we can compute the angle according to (9). We only select the edges with $|\theta| < \theta_b$ to further refine the gravity. In our experiments, we set $\theta_b = 10°$.

We gather all satisfying vertical edges from N images and construct the following energy function:

$$\arg\min_{g} \left\| \begin{pmatrix} \vdots \\ \frac{\|x_{l1}-x_{l2}\|}{\|L(l,i)\|} L(l,i) \\ \vdots \end{pmatrix} g \right\|^2, \tag{10}$$
$$\text{subject to } \|g\| = g_n = 9.80665.$$

The length of line segment $\|x_{l1} - x_{l2}\|$ is utilized as a weight since the longer edge is more reliable. The above energy function can be easily solved using SVD. In practice, there is no need to explicitly construct the full coefficient matrix, but to construct the 3×3 normal equation matrix, leading to very efficient alignment.

D. Final Refinement

With the refined gravity, we fix g and re-solve (6) to refine scale s and velocities. We then re-scale the poses and landmarks according to the refined scale. To finalize the initialization, we perform a visual-inertial BA to refine all the variables of poses, biases and 3D landmarks while keeping the gravity and scale fixed. The refined states become more consistent with the re-aligned gravity. In our implementation, we only iterate this final BA for 5 iterations.

V. Experiments

We test our initialization algorithm with the EuRoC dataset [25], which contains 11 sequences taken from three different scenes. For each scene, there are sequences labeled "easy", "medium" and "difficult", which represent how violent the drone is moving, and how many other variations like illumination changes appeared in the dataset. The dataset has made good calibration and synchronization to the sensors. In addition, it provides high-quality ground-truth poses for quantitative evaluation.

A. Evaluation of Gravity Refinement

We tested our gravity estimation with vertical edges to show its effectiveness. Given a gravity vector, we can induce a direction field and overlay it on the image. This field should align with vertical edges in the image if the gravity estimation is accurate. Figure 3 shows an example of such gravity direction field overlayed on an image.

[1] https://www.bosch-sensortec.com/bst/products/all_products/bmi160

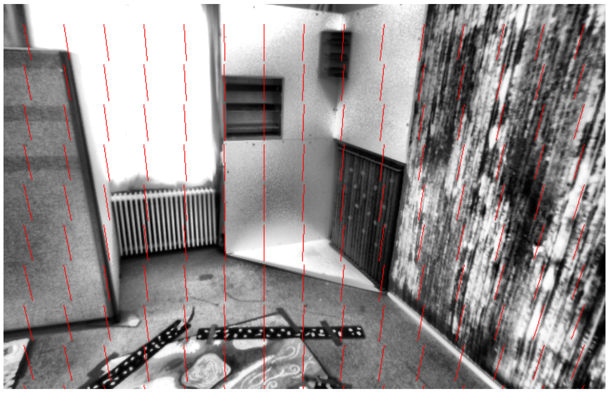


Fig. 3. A well-aligned gravity is visualized through a vector field overlayed on an image. Notice how the field lines align with the vertical edges in the image. This image is best viewed in color.

For each sequence, we continuously run our initialization algorithm. Upon finish, we can take the images and visualize the gravity. The gravity before and after alignment with vertical edges are visualized in pairs for comparison. Since the angle error in gravity can be subtle, many of the pairs may have little difference visually. Figure 4 lists some of the pairs with significant changes, and Table I lists the quantitative results for each sequence and the time spent on our gravity alignment process (without feature tracking and line extraction).

TABLE I

STATISTICS OF GRAVITY DIRECTION ESTIMATION

Sequence	Edges	ΔAngle(°)	Time(µs)
V01_01_easy	1600	2.25	234
V01_02_medium	1352	3.31	205
V01_03_difficult	1484	2.79	187
V02_01_easy	2276	1.18	248
V02_02_medium	1897	1.62	233
V02_03_difficult	1621	2.32	198
MH_01_easy	2281	1.02	311
MH_02_easy	1993	1.19	300
MH_03_medium	1810	0.98	291
MH_04_difficult	1607	1.44	246
MH_05_difficult	1561	1.31	240

Although a scene may contain many slant structures, our method still can reliably remove outlying edges by using the roughly estimated gravity in visual-inertial alignment step. As listed in Table I, our solving algorithm is quite fast (less than 0.5ms), which is negligible compared to much more expensive feature extraction and line extraction, which can take as much as 40ms per frame on our computer with an Intel Core i7 2.5GHz CPU and 8GB memory. The angle difference between the original gravity direction and the refined gravity direction is obvious, generally around $1^\circ \sim 3^\circ$.

B. Scale Estimation

In order to evaluate the accuracy of our estimated scale, we run our algorithm on sequences of EuRoC dataset and make comparisons with other state-of-the-art methods.

For each sequence, we run our initialization in a sliding-window manner. We set the number of initialization frames to $N = 8$, which is much smaller than that ($N = 15$) in [14]. We try initializing as much as possible. Upon initializing, a simple criterion is used to validate the result. We check the baseline d between the center of the first image and the center of the last image. Since the initialization is supposed to be metric, we require a "valid" initialization having $d \in [2\text{cm}, 100\text{cm}]$. The lower bound for this range is equivalently asking a valid initialization having at least 2cm movement, and the upper bound is for a safeguard. The number of valid results is recorded for each sequence.

Given valid initial poses, they are aligned with the ground-truth poses using Umeyama's method [26]. This alignment gives us the relative scale σ with respect to the ground-truth and should be 1 for perfect metric initialization. Thus, we define the relative scale error $\epsilon = |\sigma - 1|$. We then plot the success-error curve for every sequence. As shown in Figure 5. In this success-error plot, each point represents the total percent of valid initializations that has a relative scale error below a specific range.

We compared our algorithm with [13] and [14], as shown in Figure 5. The method proposed in [13] is not an instant initialization, which requires gathering a rather long period of images to start the metric alignment. In their paper, they take $10 \sim 15$ seconds to collect sufficient frames. In [14], the authors only tested the initialization 25 times. Since there is no official implementation available for [13], we re-implement their algorithms in our tracking framework. For [14], in order to make fair comparision, we made some modifications to their original implementation. All three algorithms will be initialized using the same set of frames for each test case, to avoid differences introduced by frame selection. As shown in Figure 5, our method significantly outperforms [13] and [14] in "vicon_room1". In "vicon_room2" scene, our method is also significant better than the other two methods on 'easy" and "difficult" sequences. In "machine_hall" scene, our method is slightly better on two "easy" sequences and is comparable with [14] on other sequences. In the "medium" and "difficult" sequences, the drone visits flat and dark areas where only a few vertical edges can be used. So the improvement is unobvious.

TABLE II

ANGLE (°) BETWEEN THE INITIALIZED AND THE GLOBAL GRAVITY

Sequence	[13]	[14]	Our Method
V1_01_easy	5.59	5.50	**4.01**
V1_02_medium	3.73	3.01	**2.47**
V1_03_difficult	8.29	6.67	**4.40**
V2_01_easy	4.94	4.10	**2.73**
V2_02_medium	4.35	2.70	**2.38**
V2_03_difficult	16.14	18.14	**13.84**
MH_01_easy	4.07	2.92	**2.65**
MH_02_easy	5.83	4.58	**4.34**
MH_03_medium	4.88	3.51	**2.88**
MH_04_difficult	8.33	4.60	**3.70**
MH_05_difficult	8.23	3.92	**3.52**

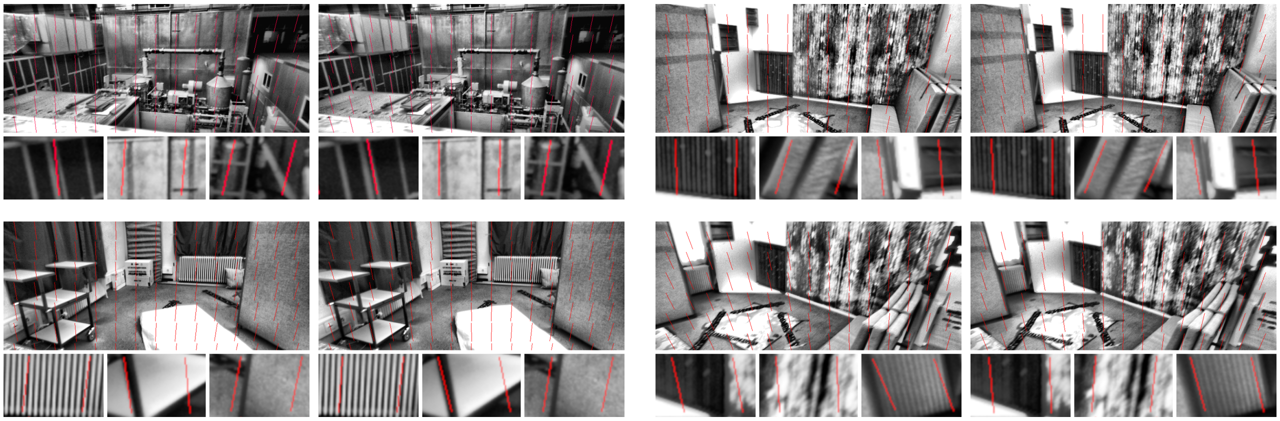


Fig. 4. Pairs of visualizations of gravity vector field before and after alignment with vertical edges. The left one is input and the right one is after alignment. Under each image, three pieces with 3x magnification are shown. The gravity field follows vertical edges better after alignment.

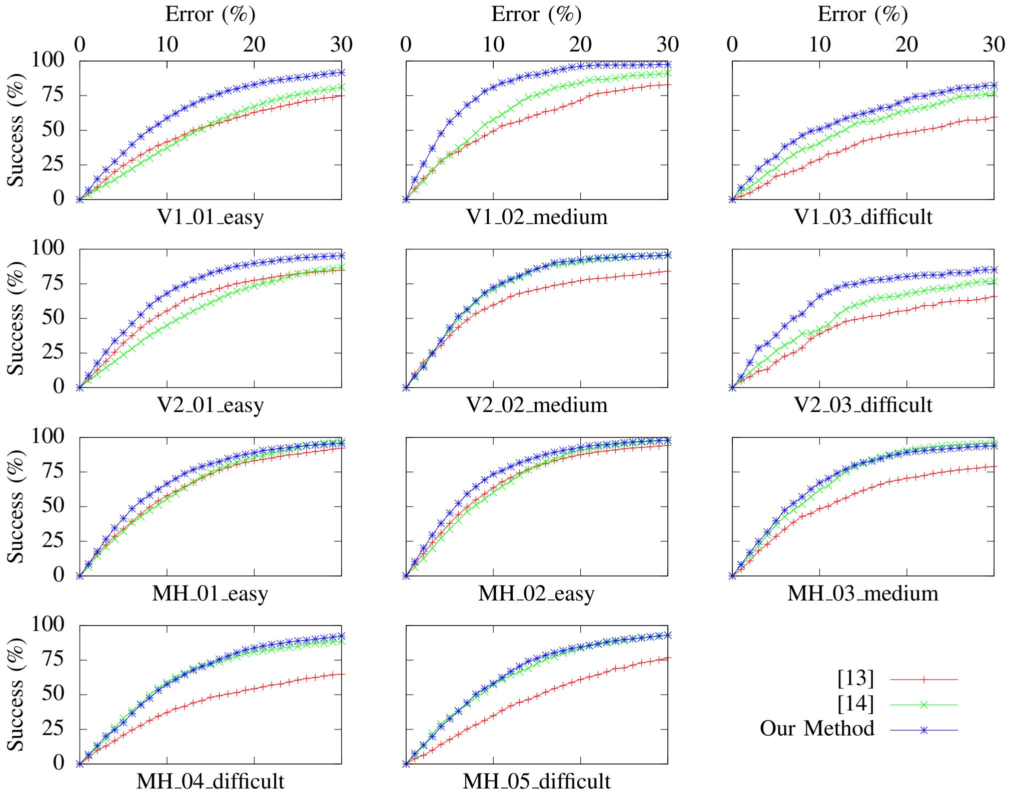


Fig. 5. Success-error comparison on EuRoC dataset with [13] and [14].

C. Improvement in Gravity Direction

In order to verify that the proposed alignment algorithm can achieve better gravity direction, we tried to compare the angle between initialized gravity and the global gravity. Unfortunately, the EuRoC dataset does not provide the ground-truth of gravity. Therefore, we run the full VINS-Mono system [6] on all the sequences. For each sequence, we aligned the recovered trajectory with the ground-truth trajectory. So we can register VINS-Mono's global gravity with the ground-truth. This global gravity is used as the substitution of the ground-truth. We followed the similar evaluation strategy in previous subsection: given an algorithm, for each successful initialization, we compute the angle between the estimated gravity and the global gravity, and we record the average angle for each sequence.

Table II are the estimated angles for three initialization algorithms on the EuRoC dataset. Our method outperforms other methods constantly. On sequences like V1_01_easy, V1_03_difficult and V2_01_easy, our method improved the gravity direction for more than 1 degree. As a result, the initialization quality was also improved, as shown in Figure 5. On sequences like V2_02_medium, our method only improves over [14] by a little, but performs much better than [13], which also explains the success-error curve for this sequence. On sequence V2_03_difficult, all the three algorithms could not get good estimation of the gravity, due to the rapid motion and fast rotation.

TABLE III

INITIALIZATION WITHOUT/WITH GRAVITY REFINEMENT EXTENSION

Sequence	VI-ORB-SLAM				VINS-Mono			
	10%		30%		10%		30%	
V1_01_easy	41.7	**50.9**	74.9	**82.8**	30.1	**46.6**	81.3	**86.6**
V1_02_medium	49.4	**53.5**	83.0	**87.6**	57.7	**63.6**	91.3	**92.3**
V1_03_difficult	29.0	**36.0**	59.5	**63.1**	40.7	**47.5**	**76.7**	72.6
V2_01_easy	55.4	**58.1**	84.8	**87.6**	45.0	**62.8**	87.0	**90.1**
V2_02_medium	59.7	**65.1**	84.2	**90.7**	**70.7**	67.9	**95.5**	93.1
V2_03_difficult	38.9	**45.3**	65.9	**70.4**	42.2	**50.0**	76.7	**76.9**
MH_01_easy	57.9	**65.5**	92.2	**95.2**	55.6	**66.8**	96.7	**97.5**
MH_02_easy	63.6	**74.5**	94.3	**97.2**	60.3	**77.0**	97.6	**98.1**
MH_03_medium	48.5	**59.6**	77.7	**82.7**	62.2	**67.7**	**95.8**	89.8
MH_04_difficult	37.2	**44.9**	64.9	**70.4**	**58.8**	55.2	**88.8**	80.1
MH_05_difficult	34.8	**45.2**	76.7	**84.4**	**57.4**	55.3	**93.1**	91.2
	Ori.	Ext.	Ori.	Ext.	Ori.	Ext.	Ori.	Ext.

D. Gravity Refinement as an Extension

Our gravity refinement with vertical edges is rather general and can directly benefit other initialization approaches. To show its effectiveness, we add our gravity refinement with vertical edges into the initialization methods in "VI-ORB-SLAM" [13] and "VINS-Mono" [14]. We implement VI-ORB-SLAM based on the original ORB-SLAM2[2]. For VINS-Mono, we directly used the original implementation[3] provided by the authors. During their initialization, we solve the gravity via alignment, and then fix this gravity in their initialization process. Table III shows the corresponding initialization success rate at 10% and 30% percent scale error, without (Ori.) or with (Ext.) gravity refinement extended. For most of the sequences, with our gravity refinement module, the success rate is significantly improved. In a few sequences, the success-error becomes slightly worse with our gravity refinement due to few valid vertical edges are detected. This again suggests that our initialization method is best suited for human-made scenearios with rich vertical edges. In real-world applications, the initialization process can be policy-based: use gravity-alignment if sufficient vertical edges are detected, and turn-off otherwise.

VI. CONCLUSION

In this paper, we propose a novel visual-inertial initialization approach by using vertical edges. The key idea is to use a better gravity information from visual cues instead of IMU to improve state estimation. By utilizing the information of vertical edges, our initialization method is not only very fast but also quite robust, which can satisfy the demand of many applications. The experimental results demonstrate the effectivenss of our initialization method.

REFERENCES

[1] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[2] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.

[3] S. He, X. Qin, Z. Zhang, and M. Jagersand, "Incremental 3D line segment extraction from semi-dense SLAM," in *International Conference on Pattern Recognition*, 2018.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

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

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

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[8] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[9] M. Li and A. I. Mourikis, "Improving the accuracy of EKF-based visual-inertial odometry," in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 828–835.

[10] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[11] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 18–25, 2017.

[12] J. Mustaniemi, J. Kannala, S. Särkkä, J. Matas, and J. Heikkilä, "Inertial-based scale estimation for structure from motion on mobile devices," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 4394–4401.

[13] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[14] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 4225–4232.

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

[16] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

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

[18] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International Workshop on Vision Algorithms*. Springer, 1999, pp. 298–372.

[19] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2016, pp. 1–10.

[20] J. Shi and Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[21] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.

[22] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge university press, 2004.

[23] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.

[24] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[25] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.

[26] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, April 1991.

[2] https://github.com/raulmur/ORB_SLAM2

[3] https://github.com/HKUST-Aerial-Robotics/VINS-Mono