

Improved Visual-Inertial Localization for Low-cost Rescue Robots

Xiaoling Long^{*,*} Qingwen Xu^{*,**,*} Yijun Yuan^{*}
Zhenpeng He^{*} Sören Schwertfeger^{*}

^{*} School of Information Science and Technology, ShanghaiTech, China
(e-mail: {longxl, xuqw, yuanwj, hezhp,
soerensch}@shanghaitech.edu.cn).

^{**} Chinese Academy of Sciences, Shanghai Institute of Microsyst &
Information Technology, ShanghaiTech, China; University of Chinese
Academy of Sciences, Beijing 100049, China

Abstract:

This paper improves visual-inertial systems to boost the localization accuracy for low-cost rescue robots. When robots traverse on rugged terrain, the performance of pose estimation suffers from big noise on the measurements of the inertial sensors due to ground contact forces, especially for low-cost sensors. Therefore, we propose *Threshold*-based and *Dynamic Time Warping*-based methods to detect abnormal measurements and mitigate such faults. The two methods are embedded into the popular VINS-Mono system to evaluate their performance. Experiments are performed on simulation and real robot data, which show that both methods increase the pose estimation accuracy. Moreover, the *Threshold*-based method performs better when the noise is small and the *Dynamic Time Warping*-based one shows greater potential on large noise.

Keywords: Robotics, Fault Detection, State Estimation, Inertial Localization, Rescue Robot

1. INTRODUCTION

Visual-inertial systems have achieved great success during the last decades, both on methods and applications. Huang (2019) reviewed all kinds of visual-inertial approaches for state estimation, with vision from direct (Usenko et al., 2016) to feature-based (Mur-Artal and Tardós, 2017) methods and fusion methods from filtering (Mourikis and Roumeliotis, 2007; Li and Mourikis, 2012) to optimization (Indelman et al., 2013; Qin et al., 2018). Also, Kuang et al. (2019); Xu et al. (2019) applied spectral methods for visual odometry. In addition, there are also multiple different applications: augmentation reality for mobile device like Google Tango¹; visual-inertial odometry on unmanned air vehicles (Ling et al., 2016; Sun et al., 2018; Do et al., 2019); visual-inertial systems on wheeled robots (Wu et al., 2017) and tracked robots (Shan et al., 2019).

Assuming that the camera can provide reliable visual information, the inertial measurement unit (IMU) largely determines the performance of visual-inertial localization. The IMU is used to obtain the acceleration and angular velocity of the system, which can be integrated to calculate rough poses. However, it can also be the drift source of inertial-based localization. Often visual-inertial systems are applied to flying robots or robots using wheels on smooth terrain and thus the reported performance of such system is often good. But, as mentioned in (Shan et al., 2019), the accelerometer equipped on a rescue robot is susceptible to a lot of vibration and ground contact forces

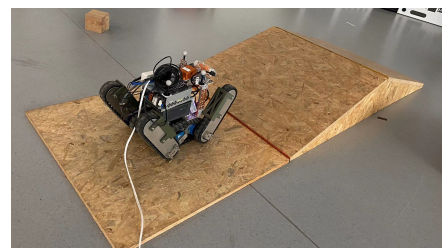


Fig. 1. MARS Rescue Robot used for the experiments.

when the robot traverses on rugged terrain, which usually causes unreliable localization. For example, robots might fall down a step, so impulses appear on acceleration measurements. After the double integration on acceleration to get the pose, the noise enlarges substantially. Thus the inertial-based localization fails.

In this work, we attempt to address such fault problems with the application on rescue robots. In other words, we focus on building robust and reliable visual-inertial systems for robots operating on rough terrain. The reason why we choose rescue robots as our main research object is, that such fault conditions of IMU is more likely to happen here than other robots which move on 2D planes, due to the challenging terrains.

There have been some attempts already to try to deal with the instabilities of the IMU resulting from different causes, such as large noise, disturbance and disconnection. Avram et al. (2015) detected the IMU bias fault based on the known roll and pitch for a UAV. Quan et al. (2019) fuses vision, wheel encoders and gyroscope tightly

^{*} The first two authors contributed equally.

¹ <https://www.google.com/atap/projecttango>.

to increase the accuracy and robustness of localization. Nguyen et al. (2009) proposed two model-based fault detection and isolation methods to maintain the robustness of a quadrotor, including parameter estimation and set membership estimation. In addition, multi-IMUs are also used to detect and isolate anomalies happened from each sensor. For example, D'Amato et al. (2017) proposed a filtering-based approach to detect faults on one of its two IMUs; Eckenhoff et al. (2019) utilized even more IMUs to address IMU failures.

The paper proposes *Threshold*-based and *Dynamic Time Warping* (DTW (Berndt and Clifford, 1994))-based IMU preprocessing methods, which make inertial based localization more reliable and robust for low-cost robots. We summarize our contributions as:

- (1) Detect IMU abnormal measurement values by both *Threshold*-based and DTW-based methods. Further replace the measurement with reasonable data.
- (2) Apply VINS-Mono on the processed data to illustrate the effectiveness of the proposed methods.
- (3) Make a rescue robot have more reliable self-localization to achieve better performance.

The rest of this paper is organized as follows: Sec. 2 analyzes and states the problem of IMU measurements. Then we propose the *Threshold*-based and DTW-based methods to address this problem in Sec. 3. Afterwards, experiments are performed on both simulation data and real robot data to evaluate the proposed methods in Sec. 4. Finally, we conclude our work in Sec. 5.

2. PROBLEM ANALYSIS

In this work, we diagnose the issue that the localization of our rescue robot equipped with the low-cost RealSense (see Fig. 1) often fails on rugged terrains when using the popular Visual Intertial System VINS-Mono (Qin et al. (2018)). For that purpose, we firstly review the IMU model used in this system. Then the real IMU data collected from the rescue robot is used to analyze the localization failure problem.

Table 1. Notation

\mathbf{a}^w	The acceleration at <i>world</i> frame
\mathbf{a}^b	The acceleration at <i>body</i> frame
$\tilde{\mathbf{a}}^b$	The measurement of acceleration at <i>body</i> frame
\mathbf{v}^w	The velocity at <i>world</i> frame
\mathbf{w}^b	The angular velocity at <i>body</i> frame
$\tilde{\mathbf{w}}^b$	The measurement of angular velocity at <i>body</i> frame
\mathbf{n}	The white noise, $n \sim \mathcal{N}(0, \sigma^2)$
\mathbf{n}^g	The white noise at <i>gyroscopes</i> frame
\mathbf{b}^a	The measurement bias at <i>accelerometer</i> frame
\mathbf{b}^g	The measurement bias at <i>gyroscope</i> frame
\mathbf{g}^w	The gravity at <i>world</i> frame
\mathbf{q}_w^b	Rotation from <i>world</i> to <i>body</i> frame in quaternion
$\mathbf{p}_i, \mathbf{q}_i$	The robot pose at time i
$\mathbf{p}_i^w, \mathbf{q}_i^w$	The robot pose at time i at <i>world</i> frame
d	The dimensionality of a single IMU measurement
\mathbf{P}, \mathbf{Q}	Two time series of IMU measurements
\mathbf{P}	One of the template patterns to be matched to
\mathbf{Q}	The actual IMU measurements
N, M	The length of \mathbf{P}, \mathbf{Q}
k	The number of template patterns

2.1 IMU Model

Considering the white noise and the bias from the random walk and ignoring the effect of scale, we can get the IMU measurement model as follows,

$$\tilde{\mathbf{w}}^b = \mathbf{w}^b + \mathbf{b}^g + \mathbf{n}^g \quad (1)$$

$$\tilde{\mathbf{a}}^b = \mathbf{q}_w^b (\mathbf{a}^w + \mathbf{g}^w) + \mathbf{b}^a + \mathbf{n}^a \quad (2)$$

Since we have the derivative of position, velocity and quaternion at time t ,

$$\dot{\mathbf{p}}^w(t) = \mathbf{v}^w(t) \quad (3a)$$

$$\dot{\mathbf{v}}^w(t) = \mathbf{a}^w(t) \quad (3b)$$

$$\dot{\mathbf{q}}_w^b(t) = \mathbf{q}_w^b(t) \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \mathbf{w}^b(t) \end{bmatrix} \quad (3c)$$

the pose of robot at time t_j is obtained by integration during time interval $[t_i, t_j]$:

$$\mathbf{p}^w(t_j) = \mathbf{p}^w(t_i) + \int_{t_i}^{t_j} \mathbf{v}^w(t) \delta t + \int \int_{t \in [t_i, t_j]} (\mathbf{q}_w^b(t) \mathbf{a}^b(t) - \mathbf{g}^w) \delta t^2 \quad (4a)$$

$$\mathbf{v}^w(t_j) = \mathbf{v}^w(t_i) + \int_{t_i}^{t_j} (\mathbf{q}_w^b(t) \mathbf{a}^b(t) - \mathbf{g}^w) \delta t \quad (4b)$$

$$\mathbf{q}_w^b(t_j) = \int_{t_i}^{t_j} \mathbf{q}_w^b(t) \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \mathbf{w}^b(t) \end{bmatrix} \delta t. \quad (4c)$$

Similarly, we also have the discrete version by replacing t_i, t_j with $k, k+1$ in proper way,

$$\mathbf{p}^w[k+1] = \mathbf{p}^w[k] + \mathbf{v}^w[k] \Delta t + \frac{1}{2} \mathbf{a}^w \Delta t^2 \quad (5a)$$

$$\mathbf{v}^w[k+1] = \mathbf{v}^w[k] + \mathbf{a}^w \Delta t \quad (5b)$$

$$\mathbf{q}_w^b[k+1] = \mathbf{q}_w^b[k] \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \mathbf{w}^b \Delta t \end{bmatrix} \quad (5c)$$

where

$$\mathbf{a}^w = \mathbf{q}_w^b[k] (\mathbf{a}^b[k] - \mathbf{b}^a[k]) - \mathbf{g}^w \quad (6a)$$

$$\mathbf{w}^b = \mathbf{w}^b[k] - \mathbf{b}^g[k] \quad (6b)$$

or Runge-Kutta approximation, which takes the average between current state and last state

$$\mathbf{a}^w = \frac{1}{2} [\mathbf{q}_w^b[k] (\mathbf{a}^b[k] - \mathbf{b}^a[k]) - \mathbf{g}^w + \mathbf{q}_w^b[k+1] (\mathbf{a}^b[k+1] - \mathbf{b}^a[k+1]) - \mathbf{g}^w] \quad (7a)$$

$$\mathbf{w}^b = \frac{1}{2} [\mathbf{w}^b[k] - \mathbf{b}^g[k] + \mathbf{w}^b[k+1] - \mathbf{b}^g[k+1]] \quad (7b)$$

2.2 Real Data Analysis

When the robot is static, its acceleration should also be stable at local gravity. When the tracked robot moves on a plane, the acceleration begins to change within an acceptable interval. However, the impulses of the acceleration may occur if the robot suddenly changes its state, such as falling down from a step, which might result in large drift in localization due to the double integration of the acceleration. Fig. 2 displays the real IMU data during runtime, which shows that the RealSense IMU has noise with larger covariance and more unexpected impulses than the Xsens IMU data. When the robot moves suddenly, such as falling down, peaks representing big acceleration occur in the IMU measurements (see Fig. 2(a)), which are the normal measurements. However, the IMU measurements with larger noise from RealSense may contain the unexpected

impulse (see Fig. 2(b)), which are the fault we want to mitigate.

In a visual-inertial system, we consider two possible effects of influence of the IMU on the pose estimation:

1) When the robot is abruptly accelerating, for example due to bumping or falling, the IMU measurements may not correctly reflect the changes according to the Nyquist sampling theory, which leads to errors on velocity and pose. For example, the robot starts to move and stop at certain point. The start and the end velocity should be zero, since the integral of acceleration through time is zero. However, the estimated value might be non-zero during run-time. Then the errors in acceleration accumulate in the velocity due to the integration, resulting in unpredictable estimation.

2) We find that instantaneous accelerations are highly related to the robot instantaneous rotations. Based on the formulation, if the instantaneous rotation is estimated with small errors, the acceleration orientation will be wrong and that error will accumulate in the velocity and position.

It should be noticed that the performance of the visual estimation is not discussed in this part. On one hand, if the vision-based pose estimation is accurate, it will also be influenced by the noise of IMU as constraints. On the other hand, if the visual estimation fails, the IMU data will be dominant in the visual-inertial system.

3. METHODOLOGY

For the inertial-based localization, the acceleration measurements will produce significant impulses if the rescue robot falls down or moves rapidly. Some of the impulses lead the robot localization to fail, especially also because it typically goes together with a large visual viewpoint change and blurred images, both of which tend to let the visual odometry fail. To make the inertial-based system more robust and accurate, we divide the approaches into two steps: fault detection and mitigation.

3.1 Fault Detection

As mentioned in Sec. 2.2, we want to detect the unexpected impulse of IMU measurements. Following are *Threshold*-based and DTW-based methods to detect these unexpected IMU impulses.

1) *Threshold*: One straightforward method is the *Threshold*-based approach, which identifies any measurement beyond the specified threshold. However, the threshold tuning is tricky: it should not affect any normal measurement and be able to identify as many as possible abnormal measurements. For example, the acceleration value when the robot begins to move suddenly might be similar to that when a glitch occurs.

2) *Dynamic Time Warping (Berndt and Clifford, 1994)*: The DTW algorithm is able to find the optimal match between two time series. Furthermore, it could be used as one kind of metric to measure the similarity of two time series. The characteristic of DTW makes it suitable for our qualification. When we apply DTW to detect the abnormal

Algorithm 1 DTW

Input Two slice $\mathbf{P} \in \mathbb{R}^{N \times d}$ and $\mathbf{Q} \in \mathbb{R}^{M \times d}$.

Output distance

```

1:  $T \leftarrow \text{array}(M + 1)$  ▷ index from 0 to M
2:  $T[0] \leftarrow 0$ 
3:  $T[i] \leftarrow \infty$  for  $i \in \{1, \dots, M\}$ .
4: for  $level \in \{1, \dots, N\}$  do
5:    $ul\_corner \leftarrow T[0]$ 
6:    $T[0] \leftarrow \infty$ 
7:   for  $i \in \{1, \dots, M\}$  do
8:      $min\_v \leftarrow \text{dist}(\mathbf{p}_{level}, \mathbf{q}_i) + \min(ul\_corner, T[i -$ 
9:        $1], T[i])$ 
10:     $ul\_corner \leftarrow T[i]$ 
11:     $T[i] \leftarrow min\_v$ 
12:   end for
13: end for
14:  $distance \leftarrow T[M]$ 

```

IMU measurement, the stream of the IMU measurements are grouped into slices of N non-overlapping samples. Our approach is then detecting possible glitches slice-wise.

The detection is made by comparing the slices (time series) of IMU readings against templates of normal IMU readings. Those templates are previously collected and contain representative examples of acceptable motion patterns of the robot, e.g. driving on smooth ground with different control input.

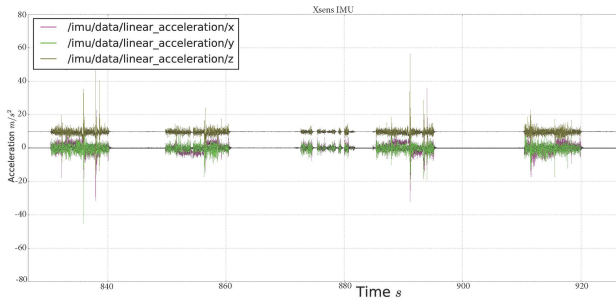
The measured IMU slices are compared against all templates. If the distance of the best fit is below a certain threshold that measurement slice is classified as normal. Otherwise that slice is considered abnormal.

The implementation details of DTW used in this work can be found in Algorithm 1. It is with two slices of measurement $\mathbf{P} \in \mathbb{R}^{N \times d}$ and $\mathbf{Q} \in \mathbb{R}^{M \times d}$. We use the concatenation of signals to represent the slice. N , M are the discretized lengths and d is the dimension of original measurements for comparison. The dimension d is typically 3 for a 3-axis accelerometer or 6 when also taking a 3-axis gyroscope into account.

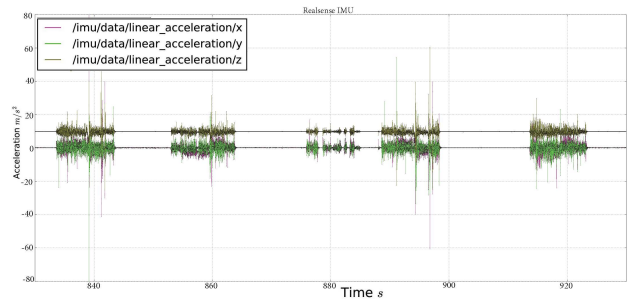
The basic distance function for two vectors $\mathbf{p}_i \in \mathbb{R}^d$, $\mathbf{q}_j \in \mathbb{R}^d$, which are the i -th and j -th rows of \mathbf{P} and \mathbf{Q} , respectively, is

$$\text{dist}(\mathbf{p}_i, \mathbf{q}_j) = \|\mathbf{p}_i - \mathbf{q}_j\|^2. \quad (8)$$

To detect the abnormal measurement, multiple templates are required, because different measurement patterns may vary. The computational efficiency of the detection algorithm is also important, due to the high frequency of the IMU measurements. Assuming k templates with length N and IMU measurements with length M , and $M \geq N$ for convenience, then the time and space complexity of detection for each test are $\mathcal{O}(kM^2)$ and $\mathcal{O}(M)$, respectively. Moreover, parallel computing is utilized to increase the speed of the detection, since comparing with each template is independent. In this case, the time complexity is $\mathcal{O}(\frac{k}{l}M^2)$ (l is the number of parallel threads), and the space complexity increases to $\mathcal{O}(lM)$.



(a) Xsens IMU Measurement Sample



(b) RealSense IMU Measurement Sample

Fig. 2. The run-time IMU measurements

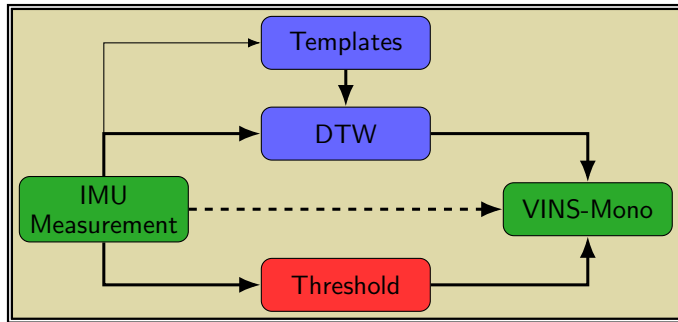


Fig. 3. Overview of the three versions of our improved VINS-Mono framework: red: Threshold version; blue: DTW-based approach; dashed line: original VINS-mono.

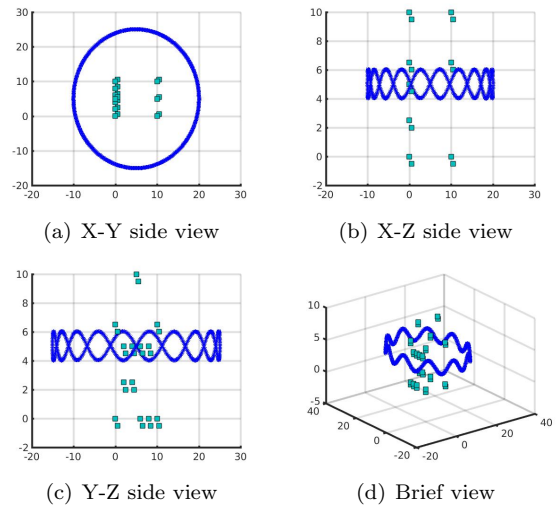


Fig. 4. Simulated trajectories and landmarks

3.2 Fault Mitigation

After detecting abnormal IMU measurements we need to replace those samples with other data, which will hopefully lead to better visual-inertial localization results. We have implemented the following strategies for the two detection methods:

1) *Threshold*: After the fault is detected by a specific threshold, we can replace the measurement with either the threshold or the average of the last n samples. The method inhibits the abnormal measurements, but it may also break the balance of the original measurements.

2) *Dynamic Time Warping*: In contrast to replacing the single measurement in the *Threshold*-based method, the DTW-based method exploits a better way that combines the replaced values with reasonable close previous measurements. As discussed previously, we collect data templates from the robot at different states for fault detection. If a fault is detected, the complete slice of IMU measurements will be replaced by the best fitting template. We think that this better keeps the balance of the acceleration measurements. The templates represent series of the normal IMU measurements. It is reasonable, because once abnormal measurements appear, a large DTW distance arises on the different templates. Since the captured measurement slice still contains useful data beside the abnormal, the closest template is still a good choice on the pattern distribution.

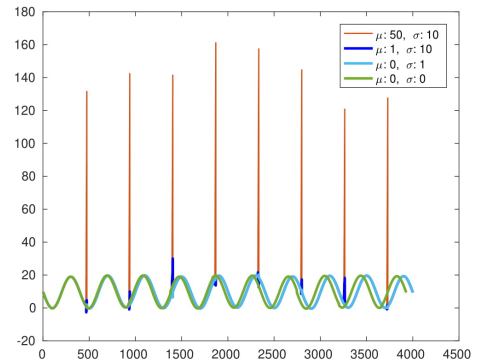


Fig. 5. Noise on acceleration along z -axis

3.3 Integration to VINS Framework

To evaluate the performance of the two different fault mitigation methods, we embedded the methods into the VINS-Mono (Qin et al., 2018) system. Fig. 3 demonstrates the improved system with the IMU abnormality mitigation mechanism. Instead of feeding IMU measurements directly into VINS-Mono (dashed line), fault detection and mitigation are applied beforehand. The proposed *Threshold*-based (red part) and DTW-based (blue part) methods are applied separately on IMU measurements. For the *Threshold*-based method, we replace the abnormal values

with the threshold when the abnormal measurements are detected. For the DTW-based approach, we first try to match the IMU measurements with all pre-obtained templates. If none is well-matched, it means the measurements might be the glitch error which we want to mitigate. If that is the case, the best fitting template is used as replacement. Then the corresponding replacement takes over the original data, leading localization into better performance.

4. EXPERIMENTS AND ANALYSIS

In this section, we compare the performance of the *Threshold*-based and DTW-based framework with original VINS-Mono on both simulation and real robot data. All the pose estimation accuracy in the experiments is evaluated by `rpg_trajectory_evaluation`² (Zhang and Scaramuzza, 2018).

4.1 Simulation

To exclude the influence of localization based on vision, we perform simulation experiments with the same number of feature points in each view. Thus we can concentrate on the influence of the IMU to the visual-inertial framework. In this experiment, the number of templates is $k = 10$ and they have a length of $N = 10$, the length of the measured slices is $M = 40$ and the dimension for the 3-axis accelerometer with 3-axis rate gyroscope is $d = 6$. The simulated IMU and keypoints data are generated from given trajectories and 3D landmarks, where 2000 poses and 36 landmarks are used in this simulation (see Fig. 4).

To make this data closer to real data, we add Gaussian noise to the idealistic raw data. In addition, to simulate the glitches, we add noise with bigger mean μ (offset) and covariance σ to a few samples. As shown in Fig. 5, the acceleration along the z -axis is influenced by noise with different μ and σ and the green line represents acceleration along z -axis with no noise.

Then we run the improved VINS-Mono system on these datasets with different amounts of glitch to compare the two fault mitigation approaches. Fig. 6 shows the results of boosting localization accuracy with two different methods and three different glitch settings. It is found that the original VINS-Mono (red line) works well when the glitch error is small ($\mathcal{N}(0, 1)$) but its ability on pose estimation decreases as the noise gets bigger. It is obvious that the original VINS-Mono fails to localize when the glitch is too large, where *Threshold*-based and DTW-based methods improve the estimation results. In addition, Table 2 gives a quantitative comparison, which shows that the DTW-based method performs best. Tables 3 and 4 display the overall relative translation and yaw error based on the three glitch settings, which indicates that both methods improve the performance of the original VINS-Mono system. It can also be observed that the DTW-based method yields better results than the *Threshold*-based one.

4.2 Real Robot Experiment

Finally, we test our approach also on a real robot. Firstly, we collect IMU data templates from measurements when

² https://github.com/uzh-rpg/rpg_trajectory_evaluation

Table 2. Root mean square of translation on simulation data

	noise_0.1	noise_1_10	noise_50_10
trunc	0.04	0.14	0.32
dtw	0.06	0.15	0.06
original	0.06	0.15	18.05

Table 3. Overall relative translation error on simulation data

	trunc	dtw	original
7.00m	0.16	0.07	2.50
14.00m	0.26	0.10	3.03
21.00m	0.37	0.14	3.54
28.00m	0.43	0.17	4.02
35.00m	0.51	0.20	4.29

Table 4. Overall relative yaw error on simulation data

	trunc	dtw	original
7.00m	0.61	0.08	9.47
14.00m	0.49	0.08	9.66
21.00m	0.53	0.09	10.96
28.00m	0.51	0.10	12.81
35.00m	0.55	0.10	14.71

the robot moves smoothly on different states, including straight moving, circling, climbing and downhill. Then IMU data is captured in the test terrain to compare with templates during detection.

In this experiment, the number of templates is $k = 10$ and they have a length of $N = 40$, the length of the measured slices is $M = 40$ and the dimension for the 3-axis accelerometer is $d = 3$.

We do not take the angular velocity into account for two reasons. Firstly, the angular velocity is captured by a gyroscope, which is not largely affected by the robot's acceleration; Secondly, the visual odometry already has trustworthy results on orientation estimation, which is different from the translation of visual odometry, which suffers from an unknown scale factor.

The real robot experiments are performed on the small rescue robot standard test scenarios build with wooden ramps. The hardware platform to run VINS-Mono is an Intel NUC (i7-8559U@2.7-4.5GHz and 16GB Memory without GPU) connected to an Intel RealSense D435i camera. Image and IMU data are all captured by this camera. Additionally, the ground truth robot poses are captured with an OptiTrack³ System. The multi-threaded C++ DTW implementation only takes about $0.6ms$ for each computation with all templates, which is about four times faster than single thread.

When collecting templates for the DTW-based method, we should ensure that no abnormal IMU data is included. For that, these templates can be generated from either high-end IMUs (like Xsens) or the reliable historical measurements of cheap IMUs. In these experiments we use the latter approach.

Fig. 7 depicts the IMU measurement and fault detection and mitigation results, which show the efficiency of the

³ <https://www.optitrack.com>

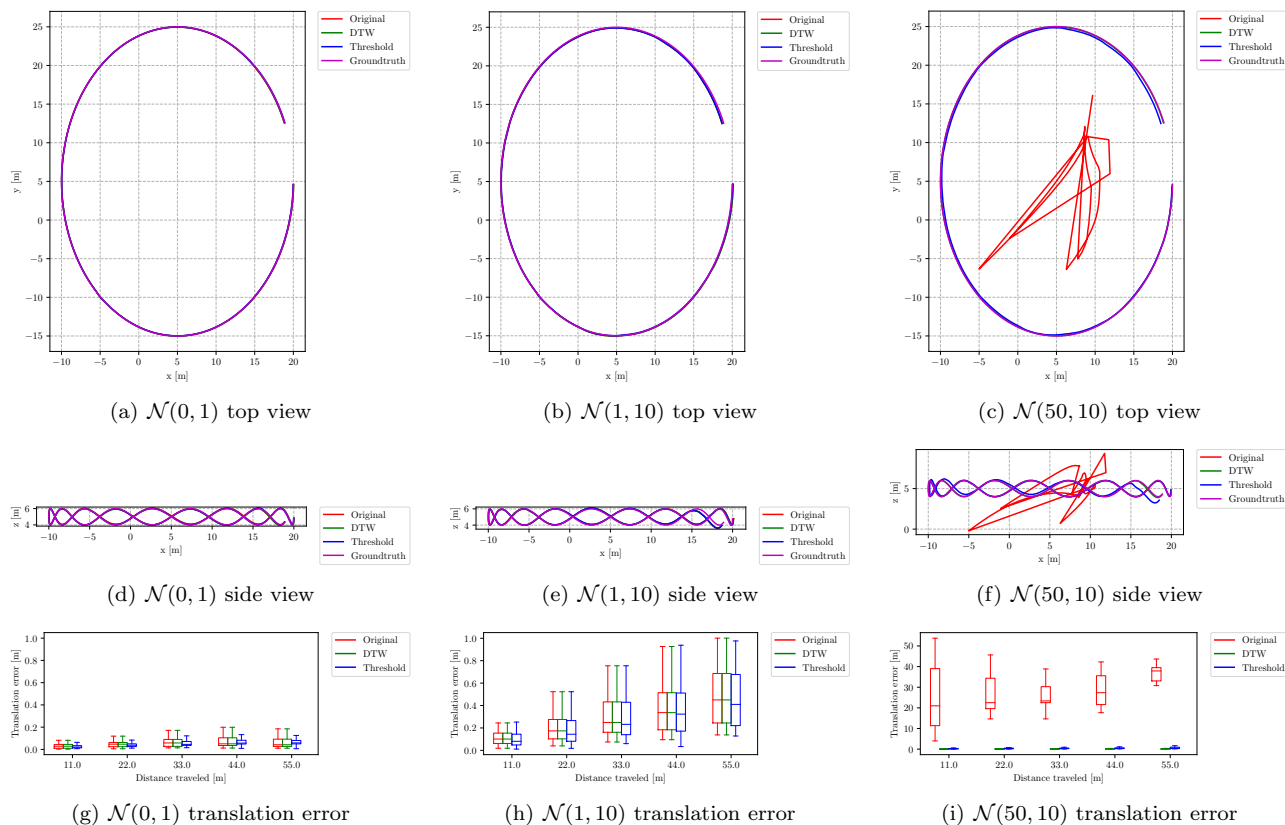


Fig. 6. Improvement on localization with different methods and glitch errors

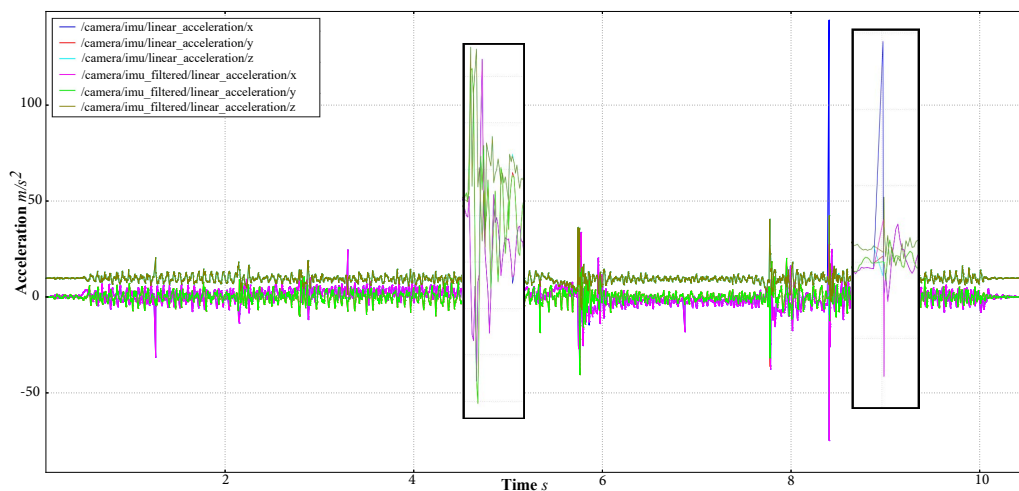


Fig. 7. A sample of IMU measurement and fault detection and mitigation result by DTW detection method for trial 2.

proposed method. The *Threshold* detection method has a quite straightforward result, which is not displayed in the paper. The first one within red box is detected when robot fall down from a ramp. The second one within red box is detected when the x -acceleration has an extremely high impulse.

Fig. 8 shows the results of the real robot experiments, which indicate that both methods improve the performance. The DTW keeps a better appearance compared to the ground truth, ignoring the scale factor from Trial 1. It should be noted that the groundtruth is shorter than

the estimated ones due to the scaling problem instead of missing data. The scale factor can be reduced further when loop closures happen. In the Trial 2, our algorithm prevents the odometry drift a lot on the x -axis. As introduced in Sec 2, the presence of high acceleration impulses will result in unpredictable estimations. The original VINS-Mono has a large continuous drift due to a faulty IMU measurement, which leads to a wrongly estimated large velocity. However, the other two improved methods have no endless drift. Even though there is a big gap between the estimated result and the ground truth, the improved

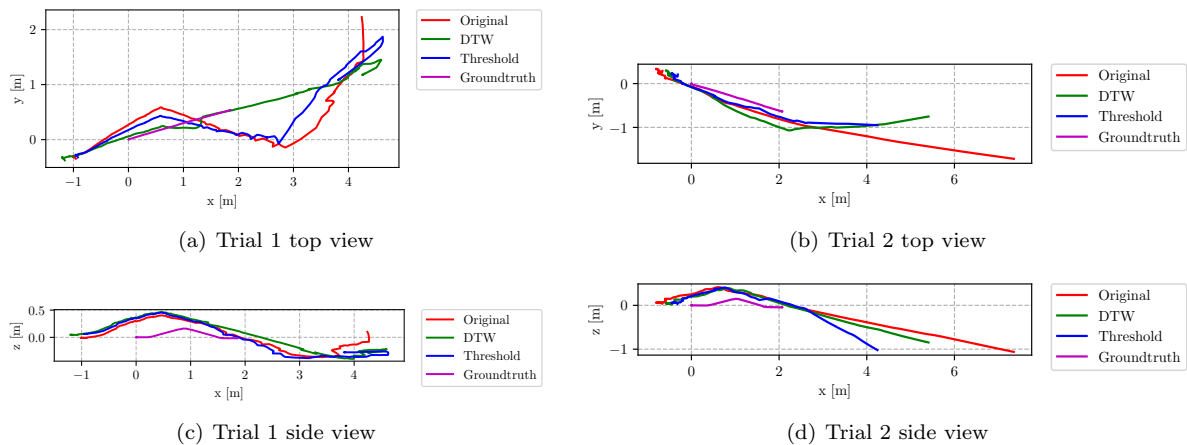


Fig. 8. Real robot trials on localization with different methods.

result shows larger probability to be reduced by further optimization.

5. CONCLUSIONS

This paper analyzes the effect of abnormal IMU measurements on visual inertial systems. Glitches in the IMU measurement can make the estimated odometry inaccurate and thus may result in failure of localization, a problem which is especially severe for ground robots. This is because lots of noise is present on the IMU measurements due to the ground contact forces. This paper proposes two different IMU preprocessing methods, which are the *threshold*-based and the *DTW*-based method.

We demonstrate our methods on both simulated data and real robot experiments. The simulation experiment shows that both methods are very well capable of detecting and mitigating simulated abnormal IMU data and the quantitative data reveals the DTW-based approach as superior to the threshold method. Furthermore, these two methods are able to improve the accuracy of visual-inertial localization. The real robot experiments also indicate a good detection of IMU glitches. But the experimental results of the integration into the VINS-Mono framework remain somewhat inconclusive, since the framework exhibits a significant overall scaling problem for all experiments. Moreover, the proposed two methods have limited improvement on localization for rescue robots.

As future work will further improve the VINS integration to remedy the scaling problems. Firstly, a more flexible template generation will be developed instead of fixed templates, to better represent the terrains information. Secondly, the mitigation strategy will be changed from the templates to data sampling from the distribution of the original IMU measurements. Finally, we will test our method on more difficult terrains and scenarios to show the robustness.

REFERENCES

Avram, R.C., Zhang, X., Campbell, J., and Muse, J. (2015). Imu sensor fault diagnosis and estimation for quadrotor uavs. *IFAC-PapersOnLine*, 48(21), 380–385.

Berndt, D.J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, 359–370. Seattle, WA.

D’Amato, E., Mattei, M., Mele, A., Notaro, I., and Scordamaglia, V. (2017). Fault tolerant low cost imus for uavs. In *2017 IEEE International Workshop on Measurement and Networking (M&N)*, 1–6. IEEE.

Do, T., Carrillo-Arce, L.C., and Roumeliotis, S.I. (2019). High-speed autonomous quadrotor navigation through visual and inertial paths. *The International Journal of Robotics Research*, 38(4), 486–504.

Eckenhoff, K., Geneva, P., and Huang, G. (2019). Sensor-failure-resilient multiimu visual-inertial navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, 3542–3548. IEEE.

Huang, G. (2019). Visual-inertial navigation: A concise review. *arXiv preprint arXiv:1906.02650*.

Indelman, V., Williams, S., Kaess, M., and Dellaert, F. (2013). Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8), 721–738.

Kuang, H., Xu, Q., Long, X., and Schwertfeger, S. (2019). Pose estimation for omni-directional cameras using sinusoid fitting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, IEEE Press.

Li, M. and Mourikis, A.I. (2012). Improving the accuracy of ekf-based visual-inertial odometry. In *2012 IEEE International Conference on Robotics and Automation*, 828–835. IEEE.

Ling, Y., Liu, T., and Shen, S. (2016). Aggressive quadrotor flight using dense visual-inertial fusion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1499–1506. IEEE.

Mourikis, A.I. and Roumeliotis, S.I. (2007). A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 3565–3572. IEEE.

Mur-Artal, R. and Tardós, J.D. (2017). Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2), 796–803.

Nguyen, H., Berbra, C., Lesecq, S., Gentil, S., Barraud, A., and Godin, C. (2009). Diagnosis of an inertial measurement unit based on set membership estimation. In *2009 17th Mediterranean Conference on Control and Automation*, 211–216. IEEE.

Qin, T., Li, P., and Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020.

Quan, M., Piao, S., Tan, M., and Huang, S.S. (2019). Tightly-coupled monocular visual-odometric slam using wheels and a mems gyroscope. *IEEE Access*, 7, 97374–97389.

Shan, Z., Li, R., and Schwertfeger, S. (2019). Rgb-d-inertial trajectory estimation and mapping for ground robots. *Sensors*, 19(10), 2251.

Sun, K., Mohta, K., Pfommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Taylor, C.J., and Kumar, V. (2018). Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2), 965–972.

Usenko, V., Engel, J., Stückler, J., and Cremers, D. (2016). Direct visual-inertial odometry with stereo cameras. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1885–1892. IEEE.

Wu, K.J., Guo, C.X., Georgiou, G., and Roumeliotis, S.I. (2017). Vins on wheels. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5155–5162. IEEE.

Xu, Q., Chavez, A.G., Bülow, H., Birk, A., and Schwertfeger, S. (2019). Improved fourier mellin invariant for robust rotation estimation with omni-cameras. In *2019 IEEE International Conference on Image Processing (ICIP)*, 320–324. IEEE.

Zhang, Z. and Scaramuzza, D. (2018). A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*.