Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at *Remote Sens.* **2020**, *12*, 1686; doi:10.3390/rs12101686

Article

Robust Visual-Inertial Integrated Navigation System Aided by Online Sensor Model Adaption for Autonomous Ground Vehicles in Urban Areas

Xiwei Bai[1], Weisong Wen[2], and Li-Ta Hsu[1,*]

1 Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China; xiwei.bai@connect.polyu.hk (X.W.B.); lt.hsu@polyu.edu.hk (L.-T.H.)

2 Department of Mechanical Engineering, the Hong Kong Polytechnic University, Kowloon, Hong Kong, China; weisong.wen@connect.polyu.hk (W.S.W.)

* Correspondence: lt.hsu@polyu.edu.hk

Abstract: Visual-inertial integrated navigation system (VINS) has been extensively studied over the past decades to provide accurate and low-cost positioning solutions for autonomous systems. Satisfactory performance can be obtained in an ideal scenario with sufficient and static environment features. However, there are usually numerous dynamic objects in deep urban areas, and these moving objects can severely distort the feature tracking process which is fatal to the feature-based VINS. The well-known method mitigates the effects of dynamic objects is to detect the vehicles using deep neural networks and remove the features belongs to the surrounding vehicle. However, excessive exclusion of features can severely distort the geometry of feature distribution, leading to limited visual measurements. Instead of directly eliminating the features from dynamic objects, this paper proposes to adopt the visual measurement model based on the quality of feature tracking to improve the performance of VINS. Firstly, a self-tuning covariance estimation approach is proposed to model the uncertainty of each feature measurements by integrating two parts: 1) the geometry of feature distribution (GFD), 2) the quality of feature tracking. Secondly, an adaptive M-estimator is proposed to correct the measurement residual model to further mitigate the impacts of outlier measurements, such as the dynamic features. Different from the conventional M-estimator, the proposed method effectively alleviates the reliance of excessive parameterization of M-estimator. Experiments are conducted in a typical urban area of Hong Kong with numerous dynamic objects, and the results show that the proposed method could effectively mitigate the effects of dynamic objects and improved accuracy of VINS is obtained when compared with the conventional method.

Keywords: Visual-inertial integrated navigation system (VINS); Visual odometry; Autonomous driving; Adaptive tuning; Urban canyons

1. Introduction

In recent years, visual-inertial integrated navigation system (VINS) has important applications in various fields due to its cost-efficiency, for example, unmanned aerial vehicles (UAV) [1, 2] and autonomous ground vehicle (AGV) positioning [3-5]. There have been significant research achievements conducted on VINS, like the VINS-Mono [5], visual-inertial direct sparse odometry (VI-DSO) [6], and semi-direct visual odometry (SVO) [7]. These existing methods have good performances in the ideal environment with sufficient texture information and static environmental features. In other words, the VINS relies heavily on the assumption that the surrounding features are static. However, the performance of VINS can be significantly impaired in dynamic outdoor scenarios due to the fact that motion blur of images damages the quality of features tracking [8]. As Figure 1

© 2020 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

shows that there are numerous dynamic objects, such as vehicles, pedestrians in the typical urban scenario. As a result, pose estimation from VINS can drift or even be lost due to the degraded features tracking caused by dynamic objects [9], such as moving vehicles and pedestrians. In fact, our previous study in [10] evaluates the performance of a state-of-the-art VINS method, the VINS-Mono [11], in diverse urban canyons with numerous dynamic objects. The results show [10] that dynamic objects are one of the major reasons that degrade the performance of VINS in urban areas. To mitigate the effects of dynamic objects on the accuracy of VINS, the major research streams include 1) dynamic objects detection based on motion tracking; 2) moving objects detection and removal based on deep learning; 3) mitigate the effects of dynamic objects using robust methods.

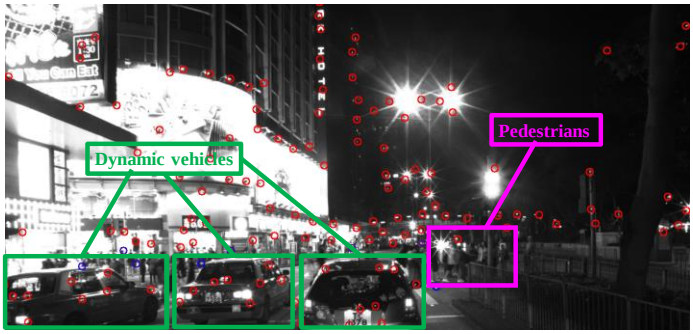


Figure 1. Illustration of the typical urban scenario with numerous dynamic objects, such as dynamic vehicles and pedestrians.

The motion tracking method [12, 13] is proposed to mitigate the effects of the dynamic objects by detecting and remodel the features belongs to the dynamic objects. Generally, the principle is to identify the features or pixels that are associated with moving objects. A pixel-wisely segmentation motion approach is introduced in [13], which was an online RGB-D data-based motion removal method, and it provides to filter out data related to moving objects. However, one of the major limitations is that large parallax can degrade the performance of foreground segmentation and cause motion tracking failure. Similar research studied in [14-16], RGB-D camera is used to provide data that has depth information of the image, which benefits the dynamic object detection and tracking. However, the maximum ranging of RGBD camera is limited (usually between 8~10 meters [17]) which is not satisfactory for outdoor applications such as UAVs and high-speed AGV. Moreover, the motion tracking based method to detect the dynamic object relies heavily on the accuracy of vehicle ego-motion estimation [18, 19], which is a fundamental challenge.

The straight forward method to mitigate the effects of dynamic objects is to detect and remove the features belong to the dynamic objects from visual simultaneous localization and mapping (SLAM) [20, 21]. Due to the many dynamic objects in complex environments, a detect-SLAM system [22] is proposed to integrate SLAM with a deep neural network to detect the moving objects and remove the unreliable features from moving objects. The DynaSLAM system [23] introduces to segment the images by the convolutional neural network (CNN) so that features belonging to the dynamic objects are rejected. Alternatively, an SSD detector [24] is presented to detect moving objects with prior knowledge, and selection tracking algorithm is proposed to eliminate dynamic objects. In addition, an ML-RANSAC algorithm [21] proposes to distinguish moving from stationary objects and classify the outliers belonging to moving objects. Although significant researches have been made in object detection [25-27], there are still many challenges in dynamic object detection. In fact, many object detection systems based on deep learning, such as the state-of-the-art YOLO [27] and FPN [28], can detect the objects (vehicles, trucks, pedestrians), but cannot determine the movement status of these objects (static or dynamic). On the other hand, these existing methods tend to remove the features from dynamic objects. However, the performance of VINS relies heavily on the number of features [9] and the geometry of feature distribution (GFD) [29]. Excessive exclusion of dynamic feature points (DFP) can severely degrade the quality of the feature tracking process. Therefore, removing all the DFP is not acceptable.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

Instead of directly removing all the detected DFP from visual SLAM, adaptively estimating the covariance of visual measurement to further de-weighting the effects of DFP for visual SLAM attracts lots of attention [30, 31]. The adaptive covariance intersection is proposed in [30] to enhance the resilience against dynamic objects in cooperative visual SLAM. However, instant information communication is required which is usually not available. Recently, the state-of-the-art method, the switchable constraints [32] is proposed, which can probabilistically detect and de-weight the outlier measurements for factor graph optimization (FGO), and improved performance is obtained. However, it relies heavily on the accuracy of the initial guess of prior switchable constraints [32]. Moreover, it requires redundancy of healthy measurements. In other words, the switchable constraints can obtain decent performance only when the healthy measurements exceed the outlier measurements. In addition, each feature can derive a switchable constraint factor in FGO which can cause an unacceptable computational load in VINS subsequently. Recently, the dynamic covariance estimation (DCE) [33] algorithm is proposed to mitigate the effects of GNSS outlier measurements and significantly improved accuracy is obtained with real-time performance. The uncertainty of GNSS measurement and the state are estimated simultaneously. However, the method relies heavily on the initial guess of the states to further calculate reliable residuals [33]. Similar work is done in [34]. Moreover, the M-estimator algorithm [35] is applied to further enhance robustness against GNSS outliers in [34]. The principle of M-estimator in factor graph optimization is to embed the standard error function with an additional robust function, such as Cauchy [36] and Huber [36] functions. However, the performance of the applied M-estimator relies heavily on tuning its parameters. In other words, the parameters of the M-estimator have to be carefully tuned based on the scenarios to obtain expected performance. Similarly, M-estimator is also used to resist the outliers measurements in VINS. In [5], the tightly-coupled integration of the visual-inertial system (VINS) is designed for state estimation of autonomous drones, and M-estimator is used to increase the robustness of VINS. However, the improvement of the performance of the VINS based on the M-estimator is limited in dynamic scenarios. In fact, similar researches are extended in [37, 38], and the same framework is used. M-estimator is applied to increase the robustness of the standard error function, and the improved performance is obtained. However, the M-estimator still has a limitation on parameter tuning.

In fact, the principle of visual and global navigation satellite systems (GNSS) is similar in the positioning that both require referenced positioning from visual measurements or receptions of satellites. The references for visual-based positioning are the tracked features and the ones for GNSS positioning are the pseudorange measurements from satellites, which can be seen in Figure 2. The major difference is that VINS requires abundant features tracking as the pose of features is unknown. However, GNSS only requires a minimum of five satellites to achieve localization, and the positions of satellites are known. Interestingly, similar positioning problems can also be seen in GNSS which is based on signals received from multiple satellites [39]. The non-line-of-sight (NLOS) receptions are similar to the dynamic feature points (DFP) in VINS as both belong to the unhealthy measurements. As Figure 2 shows, the satellite is blocked by the building, leading to the NLOS (red satellites) receptions which is similar to the DFP (the red one in the right figure). Exclusion all NLOS satellites will severely distort the geometry distribution of satellites in deep urban areas, and even cause a lack of satellites for further positioning. In our latest research [40] in GNSS positioning that makes use of both the NLOS and line-of-sight (LOS) measurements by giving them with different weightings and improved positioning performance is obtained. Therefore, we believe that remodel the outlier measurement is preferable. Interestingly, our previous work in [10] extensively evaluated the performance of VINS in urban canyons and we find that the positioning error is highly correlated with the quality of feature tracking with almost linear correlation [10]. Moreover, our recent work [41] shows that the excessive exclusion of DFP can distort the geometry of feature distribution, which can degrade the performance of VINS. Inspired by the work in [33, 34, 40] and our findings in [10, 41], this paper proposes to estimate the sensor model of visual measurement (tracked feature) online based on the quality of feature tracking. Firstly, a self-tuning covariance estimation approach is proposed to model the uncertainty of each feature measurements by integrating two parts: 1) the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

geometry of feature distribution (GFD), 2) the quality of feature tracking. Secondly, an adaptive M-estimator is proposed to correct the measurement residual model to further mitigate the impacts of outlier measurements, such as the dynamic features. Different from the conventional M-estimator, the proposed method effectively alleviates the reliance of excessive parameterization of M-estimator.

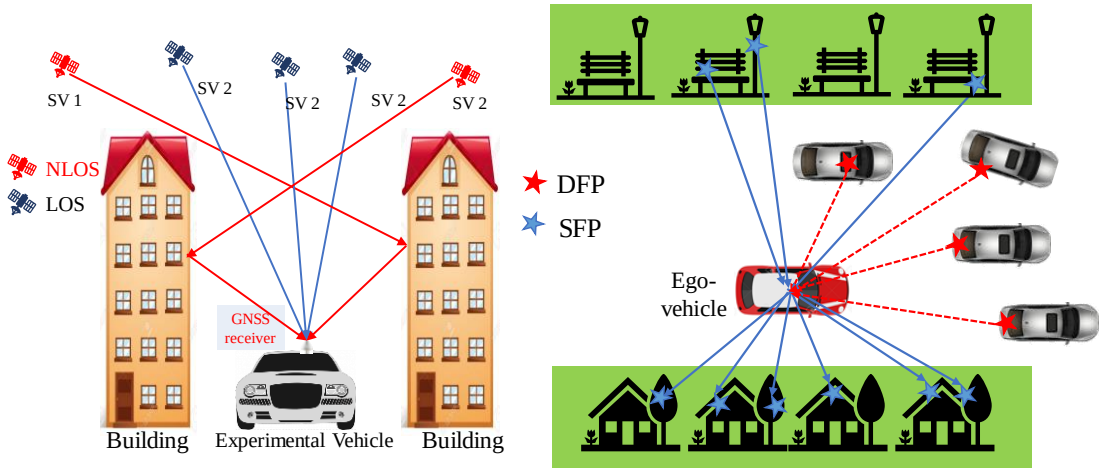


Figure 2. The positioning principles of GNSS and visual odometry. LOS denotes line of satellite. NLOS denotes non-line of satellite. SFP denotes static feature point DFP denotes dynamic feature point

The rest of the paper is organized as follows: In Section II, the overview of the proposed method is given, and then the framework of VINS is introduced in Section III. Next, the online sensor model estimation is presented in Section IV. The experiment result is given in Section V. Finally, the conclusion is summarized.

2. Overview of the Proposed Adaptive VINS

The structure of the proposed method is shown in Figure 3. The inputs of the system consist of two parts, the inertial measurement unit (IMU) and the monocular camera. The IMU provides acceleration and angular velocity measurements. The monocular camera provides raw image. In the modeling stage, due to the high data frequency of IMU, multiple IMU measurements are obtained between two consecutive frames. In order to reduce the computational loads, the IMU pre-integration [42] is employed to derive the motion between the consecutive frames. Then the pre-integration factor is obtained. Besides, the feature is extracted and tracking for the visual modeling. To guarantee the quality of features, there are two parameters considered in this paper. The geometry of feature distribution is derived from feature extraction. The number of times that the feature being tracked is derived from the feature tracking. Both are used in the proposed adaptive covariance estimation, which can remodel the uncertainty of visual measurements. Then the standard reprojection factor is obtained based on the adaptive covariance estimation. In addition, the quality of feature tracking is associated with the parameters of adaptive M-estimator, which can increase the robustness of the standard reprojection factor against outliers with additional robust function. Then the robust reprojection factor is obtained. Finally, the pre-integration factor and robust reprojection factor are integrated into a factor graph optimization (FGO). The optimization result can correct the bias of IMU measurements in turn. The detail of the proposed method is given in the following sections.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

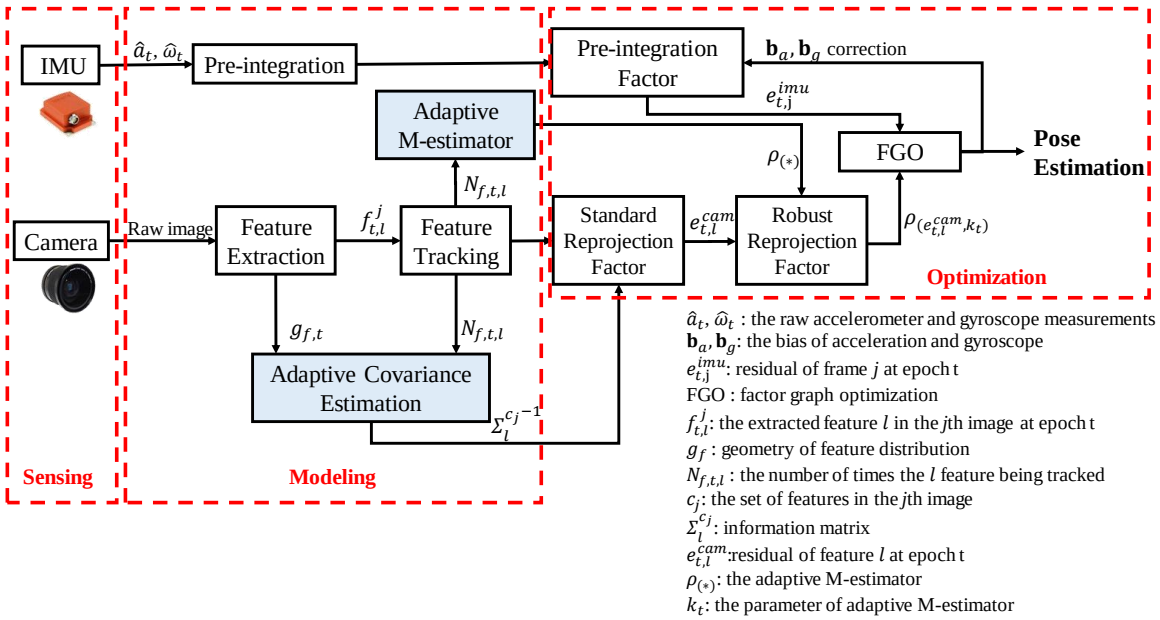


Figure 3. Flowchart of the proposed VINS that aided by adaptive covariance estimation and M-estimator method.

3. Tightly-Coupled Monocular-based Visual-inertial Integration based on Factor Graph Optimization

3.1. System States

The objective of factor graph optimization is to minimize the residuals derived from multiple sensor measurements [43]. In this paper, the residuals include the one from the IMU measurements and the one from visual measurements. The state vector considered in this paper is defined as follows,

$$\chi = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \mathbf{x}_c^b, \lambda_1, \lambda_2, \dots \lambda_M] \tag{1}$$

$$\mathbf{x}_k = [\mathbf{P}_{b_k}^w, \mathbf{V}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_{a,k}, \mathbf{b}_{g,k}], \quad k\epsilon[1,n] \tag{2}$$

$$\mathbf{x}_c^b = [\mathbf{P}_c^b, \mathbf{q}_c^b] \tag{3}$$

where the superscript w is the world frame, and subscript b_k is the body frame (same as IMU frame) while taking the kth image. $\mathbf{x}_k$ is the IMU state at the kth image. It contains the position ($\mathbf{P}_{b_k}^w$), velocity ($\mathbf{V}_{b_k}^w$), and orientation that represented by quaternion ($\mathbf{q}_{b_k}^w$) in the world frame, and acceleration bias ($\mathbf{b}_{a,k}$) and gyroscope bias ($\mathbf{b}_{g,k}$) in the IMU body frame. n is the total number of key frames considered for optimization and M is the total number of features considered. λ_l is the inverse depth of the lth feature observed for the first time, $l\in(1,M)$. $\mathbf{x}_c^b$ is the extrinsic parameter that transforms the camera frame into the IMU frame. To guarantee the computation efficiency, we only make use of the measurements inside a sliding window (which can be seen in Figure 4) to estimate the states. The images inside in the sliding window are between frame b_k and b_{k+n}, with the time of t_k and t_{k+n}, respectively. Regarding the implementation of the VINS, we make reference to the framework in [5].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

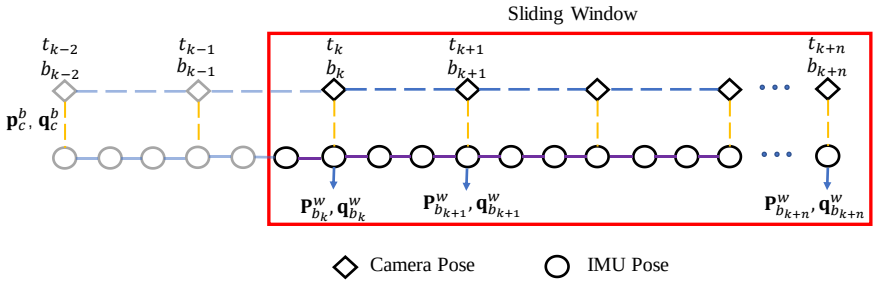


Figure 4. Illustration of the sliding window used in the proposed graph optimization.

3.2. IMU Measurement Modeling

This section presents the modeling of IMU measurement. IMU measurements are given in the body frame, which is affected by the additive noise and bias of acceleration and gyroscope. The raw accelerometer and gyroscope measurements are given in the body frame at a given time t, respectively by,

$$\hat{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{b}_{a_t} + \mathbf{n}_a \tag{4}$$

$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \mathbf{b}_{\omega_t} + \mathbf{n}_\omega \tag{5}$$

where $\hat{\mathbf{a}}_t$, $\hat{\boldsymbol{\omega}}_t$ are the raw measurements of IMU, $\mathbf{a}_t$, $\boldsymbol{\omega}_t$ are the expected measurements of acceleration and angular velocity of IMU. $\mathbf{g}^w = [0 \quad 0 \quad g]^T$ is the gravity vector in the world frame. $\mathbf{R}_w^t$ denotes the rotation matrix that transforms world frame into the body frame at time t. $\mathbf{b}_{a_t}$, $\mathbf{b}_{\omega_t}$ are the acceleration bias and gyroscope bias. $\mathbf{n}_a$, $\mathbf{n}_\omega$ are the additive noise, which is assumed that is Gaussian white noise, $\mathbf{n}_a \sim \mathcal{N}(0, \sigma_a^2)$, $\mathbf{n}_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$. The values of the $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are determined based on the specification of IMU.

The IMU measurements can be employed to constrain the motion between two epochs using the standard IMU mechanism [44], which can work efficiently in the filtering-based sensor fusion, such as the extended Kalman filter (EKF) [45]. However, the standard IMU mechanism [44] can cause a high computation load in sensor fusion using FGO [46], due to the high frequency of IMU measurement. We employ the state-of-the-art IMU pre-integration technique [47, 48] to integrate the IMU measurements, which can effectively alleviate the high computation load in FGO and the accuracy is guaranteed, by integrating multiple IMU measurements into a single factor in FGO. There are several inertial measurements in time interval $t \in [t_k, t_{k+1}]$ between two consecutive frames b_k and b_{k+1}. Given the bias estimation, the IMU pre-integration is integrated in the b_k frame as follows [5],

$$\boldsymbol{\alpha}_{b_{k+1}}^{b_k} = \iint_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{b_k} \left(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t}\right) dt^2 \tag{6}$$

$$\boldsymbol{\beta}_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{b_k} \left(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t}\right) dt \tag{7}$$

$$\boldsymbol{\gamma}_{b_{k+1}}^{b_k} = \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t}) \boldsymbol{\gamma}_t^{b_k} dt \tag{8}$$

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y & \omega_x \\ \omega_z & 0 & -\omega_x & \omega_y \\ -\omega_y & \omega_x & 0 & \omega_z \\ \omega_x & \omega_y & \omega_z & 0 \end{bmatrix} \tag{9}$$

where $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}$, $\boldsymbol{\beta}_{b_{k+1}}^{b_k}$, and $\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ are the pre-integration terms between frames b_k and b_{k+1}, which represent the changes of position, velocity, and orientation, respectively. $\mathbf{R}_t^{b_k}$ is the rotation matrix

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

that transforms the body frame at time t into reference frame b_k. In fact, this is one of the major differences compared with the IMU mechanism, as the pre-integration is performed in the body frame b_k and the IMU mechanism is conducted with respect to the world frame. $\boldsymbol{\gamma}_t^{b_k}$ is a quaternion that transforms the body frame at time t into reference frame b_k. The ω_x, ω_y and ω_z denote the angular velocities in the body frame.

The IMU pre-integration between the two consecutive frames takes the b_k as the reference frame. Based on the information, the position, velocity and orientation in the world frame can be derived as follows,

$$\mathbf{P}_{b_{k+1}}^{w} = \left(\mathbf{P}_{b_k}^{w} + \mathbf{V}_{b_k}^{w}\Delta t_k - \frac{1}{2}\mathbf{g}^{w}\Delta t_k^2\right) + \mathbf{R}_{b_k}^{w}\boldsymbol{\alpha}_{b_{k+1}}^{b_k} \tag{10}$$

$$\mathbf{V}_{b_{k+1}}^{w} = \left(\mathbf{V}_{b_k}^{w} - \mathbf{g}^{w}\Delta t_k\right) + \mathbf{R}_{b_k}^{w}\boldsymbol{\beta}_{b_{k+1}}^{b_k} \tag{11}$$

$$\boldsymbol{\gamma}_{b_{k+1}}^{b_k} = \mathbf{q}_{w}^{b_k} \otimes \mathbf{q}_{b_{k+1}}^{w} \tag{12}$$

According to the two known states of b_k and b_{k+1}, the residual for IMU pre-integration measurement in the two consecutive frames b_k and b_{k+1} can be defined as follows [5],

$$\mathrm{r}_{\mathcal{B}}\left(\hat{\mathbf{Z}}_{b_{k+1}}^{b_k}, \boldsymbol{\chi}\right) = \begin{bmatrix} \delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\theta}_{b_{k+1}}^{b_k} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_\omega \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{w}^{b_k}\left(\mathbf{P}_{b_{k+1}}^{w} - \mathbf{P}_{b_k}^{w} + \frac{1}{2}\mathbf{g}^{w}\Delta t_k^2 - \mathbf{V}_{b_k}^{w}\Delta t_k\right) - \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \mathbf{R}_{w}^{b_k}\left(V_{b_{k+1}}^{w} + \mathbf{g}^{w}\Delta t_k - \mathbf{V}_{b_k}^{w}\right) - \boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ 2\left[\mathbf{q}_{b_k}^{w^{-1}} \otimes \mathbf{q}_{b_{k+1}}^{w} \otimes \left(\boldsymbol{\gamma}_{b_{k+1}}^{b_k}\right)^{-1}\right]_{xyz} \\ \mathbf{b}_{a,b_{k+1}} - \mathbf{b}_{a,b_k} \\ \mathbf{b}_{\omega,b_{k+1}} - \mathbf{b}_{\omega,b_k} \end{bmatrix} \tag{13}$$

where $\hat{\mathbf{Z}}_{b_{k+1}}^{b_k}$ represents the observation measurements of IMU between frames b_k and b_{k+1}. The operator $[.]_{xyz}$ is used for extracting the vector part of a quaternion $\mathbf{q}$ for the orientation difference. $\otimes$ means multiplication operation between two quaternions. $\Delta\boldsymbol{\theta}_{b_{k+1}}^{b_k}$ represents the orientation constraint between frames b_k and b_{k+1}. The $\delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k}$ represents the derived position constraint between frames b_k and b_{k+1}. The $\delta\boldsymbol{\beta}_{b_{k+1}}^{b_k}$ denotes the velocity constraint. The $\delta\mathbf{b}_a$ and $\delta\mathbf{b}_\omega$ denote the accelerometer and gyroscope biases constraints, respectively. The $\left[\boldsymbol{\alpha}_{b_{k+1}}^{b_k}, \boldsymbol{\beta}_{b_{k+1}}^{b_k}, \boldsymbol{\gamma}_{b_{k+1}}^{b_k}\right]$ represents pre-integrated measurements between frames b_k and b_{k+1}. When the estimation of bias changes, the IMU measurements will be repropagated under the new bias estimation.

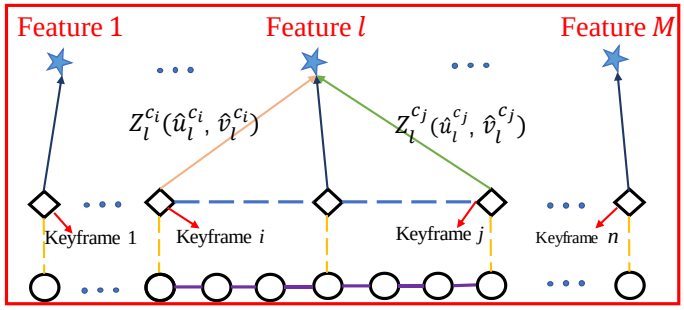
3.3. Visual Measurement Modeling

This section presents the modeling of visual measurement. The direct raw measurement from the camera is the raw image at a given epoch t. Similar to the work in [5], we formulate the visual measurement residual based on a reprojection error. For a given new image, the features are detected by the Shi-Tomasi [49] corner detection algorithm. Meanwhile, the Kanade-Lucas-Tomasi (KLT) sparse optical flow algorithm [50] is employed to track the features. The derivation of the reprojection error relies heavily on the quality of feature tracking. To guarantee that enough features are detected in a frame of the image, new corner features are also detected [49]. During the feature tracking, only certain images, the keyframes, are employed to perform the feature tracking to enforce the efficiency. The keyframes are chosen based on two criteria: 1) The first one is the average parallax criteria: if the average parallax of the tracked features between the current frame and the latest keyframe override a certain threshold, then the current frame is treated as new keyframe. 2) if the number of the tracked features inside the current image is lower than a certain threshold, then this frame is regarded as a new keyframe. Figure 5 denotes the feature tracking processing where n is the total number of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

keyframes in the sliding window. The lth feature is firstly observed in the ith image. The $Z_l^{c_i}(\hat{u}_l^{c_i}, \hat{v}_l^{c_i})$ represents first observation of lth feature in ith image. The $Z_l^{c_j}(\hat{u}_l^{c_j}, \hat{v}_l^{c_j})$ denotes the observation of the same feature in jth image. We can see from Figure 5 that the feature is tracked for several times.



M : total number of features in the sliding window
c_i, c_j : the set of features observed in the ith and jth images

$Z_l^{c_i}$, $Z_l^{c_j}$: observation of the lth feature in the ith and jth images
$(\hat{u}_l^{c_i}, \hat{v}_l^{c_i})$, $(\hat{u}_l^{c_j}, \hat{v}_l^{c_j})$: the pixel location of lth feature in ith and jth images

Figure 5. Illustration of the feature tracking process.

The traditional reprojection residual is defined in the image plane, but this model is not suitable for most of camera models. In this paper, we follow the work [5], the residual is defined on a unit sphere, which is appliable for almost all of camera models. The unit vector for observation of the lth feature in the jth image that is projected into the unit sphere, $\hat{\bar{p}}_l^{c_j}$, is calculated as follows [5] :

$$\hat{\bar{p}}_l^{c_j} = [\mathrm{I}_1 \quad \mathrm{I}_2]^T \cdot \pi_c^{-1}\left(\begin{bmatrix}\hat{u}_l^{c_j}\\ \hat{v}_l^{c_j}\end{bmatrix}\right) \tag{14}$$

where $[\mathrm{I}_1 \quad \mathrm{I}_2]^T$ are two arbitrarily selected orthogonal bases on the tangent plane corresponding to the feature observation. The π_c^{-1} is the back-projection function, which turns a pixel location into a unit vector using camera intrinsic parameters. To formulate the residual corresponding to the measurement $\hat{\bar{p}}_l^{c_j}$, the expected observation $p_l^{c_j}$ is needed. The direct method is to derive the $p_l^{c_j}$ based on the current state χ. To make best use of the feature tracking process which provide continuous geometry constraints, we derive the $p_l^{c_j}$ based on the keyframe i. For the sake of clearer explanation, we divide the formulation into several steps as follows:

Step 1: get the feature l from the pixel position in image i to the body frame (IMU frame) as follows:

$$\mathbf{S_1} = \mathbf{R}_c^b \frac{1}{\lambda_l}\pi_c^{-1}\left(\begin{bmatrix}\hat{u}_l^{c_i}\\ \hat{v}_l^{c_i}\end{bmatrix}\right) + \mathbf{P}_c^b \tag{15}$$

The $\mathbf{R}_c^b$ and $\mathbf{P}_c^b$ represent the rotation matrix and translation matrix from the camera frame to the body frame. Then the pixel location $(\hat{u}_l^{c_i}, \hat{v}_l^{c_i})$ in the ith image is transformed into the body frame.

Step 2: get the feature l in the ith image from the body frame to the world frame, and then translated to the jth image in the world frame as follows:

$$\mathbf{S_2} = \mathbf{R}_{b_i}^w(\mathbf{S_1}) + \mathbf{P}_{b_i}^w - \mathbf{P}_{b_j}^w \tag{16}$$

The $\mathbf{R}_{b_i}^w$ and $\mathbf{P}_{b_i}^w$ are the rotation matrix and translation matrix which transforms the lth feature detected in the ith image from the body frame to the world frame. The $\mathbf{P}_{b_j}^w$ is the translation matrix which transforms the lth feature detected in the jth image from the body frame to the world frame.

Step 3: get the feature l in the jth image from world frame to the body frame, and then transformed into the camera frame as follows:

$$\mathbf{S_3} = \mathbf{R}_w^{b_j}(\mathbf{S_2}) - \mathbf{P}_c^b \tag{17}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

$$p_l^{c_j} = \mathbf{R}_b^c(\mathbf{S_3}) \tag{18}$$

The $\mathbf{R}_w^{b_j}$ represents the rotation matrix which transforms the same feature in the jth image from the world frame to the body frame. The $\mathbf{P}_c^b$ is the translation matrix that transforms the camera frame to the body frame. The $p_l^{c_j}$ denotes the predicted feature measurement on the unit sphere by transforming its first observation in the ith image to jth image. The $\mathbf{R}_b^c$ is the rotation matrix that transforms the body frame to the camera frame.

Step 4: Therefore, the residual for lth feature measurement in keyframe j is defined as follows,

$$\mathrm{r}_C\left(\hat{\mathbf{Z}}_l^{\boldsymbol{c_j}}, \boldsymbol{\chi}\right) = [\mathrm{I}_1 \quad \mathrm{I}_2]^T \cdot (\hat{\hat{p}}_l^{c_j} - \frac{p_l^{c_j}}{\left\|p_l^{c_j}\right\|}) \tag{19}$$

The $\mathrm{r}_C(*)$ represents the residual of the lth feature measurement in the jth image. $\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{c_j}}$ denotes the observation measurement of lth feature in the jth image. Be noted that the degree of freedom of the feature is two dimensions, therefore the residual is projected in the tangent plane. The $\hat{\hat{P}}_l^{c_j}$ denotes the unit vector for the observation of the lth feature in the jth frame.

3.4. Marginalization

Each feature measurement corresponds to a factor in FGO. Therefore, the computational complexity will increase dramatically over time. The straightforward way is to remove part of old states and their associated measurements. However, this will fail to make use of historical data. In order to reduce the computational loads and guarantee the accuracy, the marginalization is used. The process of marginalization is to marginalize some older visual measurements. During the system optimization, some of the unsatisfactory IMU states and features are marginalized out from the sliding window into a prior. The two strategies proposed [5] to select marginalized measurements. Firstly, if the second latest frame is a keyframe, it will be kept in the sliding window, meanwhile, the oldest frame is marginalized out with its corresponding measurements. Conversely, if the second latest frame is a non-keyframe, the visual measurements will be left out, and IMU measurements are kept that connect to this non-keyframe, which can maintain the sparsity of the system. The marginalization is carried out by the Schur complement [51]. A new prior is constructed based on all marginalized measurements related to the removed state and the residual for the prior factor can be derived accordingly.

3.5. Visual-Inertia Optimization

Based on the derived residuals from 1) residual from IMU pre-integration, 2) residual from the visual measurement and 3) residual from marginalization. The objective of the FGO is to minimize the sum of prior and the Mahalanobis norm of all measurement residuals to obtain a maximum posterior estimation. The cost function of the system is as follows:

$$\min_{\boldsymbol{\chi}} \left\{ \left\|\mathbf{r}_{\boldsymbol{p}} - \mathbf{H}_{\boldsymbol{p}}\boldsymbol{\chi}\right\|^2 + \sum_{k\epsilon\mathcal{B}} \left\|\mathrm{r}_{\mathcal{B}}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{b_{k+1}}}^{\boldsymbol{b_k}}, \boldsymbol{\chi}\right)\right\|^2_{\mathbf{P}_{\boldsymbol{b_{k+1}}}^{\boldsymbol{b_k}}} + \sum_{(l,j)\epsilon C} \left\|\mathrm{r}_C\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{c_j}}, \boldsymbol{\chi}\right)\right\|^2_{\mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}}} \right\} \tag{20}$$

where $\{\mathbf{r}_{\boldsymbol{p}}, \mathbf{H}_{\boldsymbol{p}}\}$ is the prior information from the marginalization operation. $\mathrm{r}_{\mathcal{B}}(.)$ is the residual term for IMU pre-integration. $\mathrm{r}_C(.)$ Is the residual term for visual re-projection. $\mathcal{B}$ is the set of all IMU measurements, C is the set of features that have been observed at least twice in the current sliding window. $\mathbf{P}_{\boldsymbol{b_{k+1}}}^{\boldsymbol{b_k}}$ is the information matrix for IMU pre-integration. $\mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}}$ is the information matrix for visual re-projection, which represents the uncertainty of feature measurements. In [5], the $\mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}}$ is fixed and is correlated with the focal length. The information matrix is the inverse of the covariance matrix. The fixed information matrix can work well in an ideal scenario. Unfortunately, the positioning result will be significantly misled by unmodeled outliers. Therefore, in the next section, we propose an online sensor model to adaptively estimate the uncertainty of visual measurements.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

4. Online Sensor Model Estimation

According to our previous work in [10], the result shows that the dynamic feature points from dynamic objects are one of the major factors degrading the performance of VINS in the urban areas. In addition, the positioning error is highly correlated to the quality of feature tracking, the number of times that the feature is tracked. We propose to mitigate the effects of dynamic objects by adaptively estimating the uncertainty of visual measurement based on the quality of feature tracking from two aspects, the adaptive covariance estimation and adaptive M-estimator in the remainder of this section.

4.1. Adaptive Covariance Estimation

Based on our findings in [10], we propose to correlate the uncertainty of a given visual measurement with two parts: 1) the quality of feature tracking which is determined by the number of times the feature tracked (NTFT). In fact, the more times we see the same feature, the better is the feature quality. 2) the geometry distribution factor ($g_{f,t}$) which is determined by the geometry of feature distribution (GFD). Assuming that a set of tracked features at a given epoch t from the jth image are denoted by $\mathbf{F}_t^j$ as follows:

$$\mathbf{F}_t^j = \{\boldsymbol{f}_{t,1}^j, \boldsymbol{f}_{t,2}^j, \cdots, \boldsymbol{f}_{t,m}^j\} \tag{21}$$

where the m represents the number of features in the jth image. Each feature $\boldsymbol{f}_{t,l}^j$ is represented by $\boldsymbol{f}_{t,l}^j = \{u_{t,l}, v_{t,l}, N_{f,t,l},\}$. The $u_{t,l}$ and $v_{t,l}$ denote the pixel position of the feature in the image. The $N_{f,t,l}$ denotes the number of times that the feature l is tracked.

In fact, each feature corresponds to a landmark that constrains the pose of the camera in VINS. Different geometry distribution of features can result in the different performance of state estimation of the system. The ideal condition is that all the tracked features are uniformly distributed in an image. Unfortunately, this is usually not available due to environmental conditions. In other words, the distribution of the features relies highly on the distributions of surrounding objects, such as buildings, vehicles, etc. Figure 6 shows the geometry distribution of the features in two cases. The figure (a) shows a good geometry distribution of features where the features distribute over the whole image. The figure (b) shows a case where majority of the features locate in the middle of the Figure.

As shown in Figure 2, the feature-based VINS positioning is similar to the satellite-based GNSS positioning. The precision of GNSS positioning is highly related to the geometry distribution of satellites with respect to the GNSS receiver. The quality of the distribution is described using 3D position dilution of precision (PDOP) [52]. Inspired by this fact, we adopt the similar idea from GNSS positioning to describe the quality of geometry distribution of features, the $g_{f,t}$.

Firstly, given the estimated initial guess of the position of camera and the detected features using standard state initialization [53], the observation matrix correlating the position of both the camera and features is derived as follows,

$$\mathbf{H} = \begin{bmatrix} (x_1 - x)\lambda_1 & (y_1 - y)\lambda_1 & (z_1 - z)\lambda_1 \\ (x_2 - x)\lambda_2 & (y_2 - y)\lambda_2 & (z_2 - z)\lambda_2 \\ \vdots & \vdots & \vdots \\ (x_m - x)\lambda_m & (y_m - y)\lambda_m & (z_m - z)\lambda_m \end{bmatrix}_{m\times 3} \tag{22}$$

where (x, y, z) denotes the position of camera which can be derived from system state at epoch t, (x_m, y_m, z_m) denotes the 3D position of mth feature at epoch t. Based on the derivation of PDOP, the $\mathbf{Q}$ matrix can be derived as follows,

$$\mathbf{Q} = (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1} \tag{23}$$

where $\mathbf{Q}$ is 3x3 matrix as follows,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix} \tag{24}$$

where the σ_x^2 , σ_y^2 , σ_z^2 denote the uncertainty associated with the geometry distribution. The smaller $\mathrm{g}_{f,t}$ means that the features are more decentralized which can lead to better VINS estimation and vice versa. Therefore, the $\mathrm{g}_{f,t}$ is calculated as follows,

$$\mathrm{g}_{f,t} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{25}$$

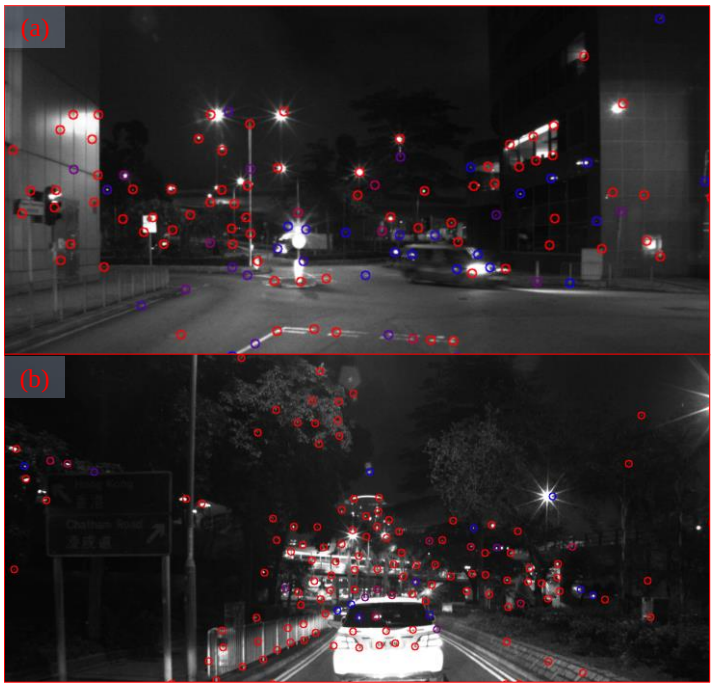


Figure 6. Illustration of the (a) decentralized feature distribution, and (b) partially centralized feature distribution due to the environmental conditions. The circles represent the detected and tracking features. The red circle means that the features are tracked more times than the blue one.

Therefore, the adaptive information matrix ($\Sigma_l^{c_j}$) is as follows:

$$\boldsymbol{\Sigma}_{\boldsymbol{l}}^{\boldsymbol{c_j}} = \mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}} N_{f,t,l} \cdot \frac{1}{\mathrm{g}_{f,t}} \cdot s \tag{26}$$

$$\text{with } \mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}} = \begin{bmatrix} \frac{\mathrm{F_c}}{1.5} & 0 \\ 0 & \frac{\mathrm{F_c}}{1.5} \end{bmatrix}$$

where $\mathbf{P}_{\boldsymbol{l}}^{\boldsymbol{c_j}}$ is the original information matrix from [5], s is the scaling factor which is experimentally determined. The $\mathrm{F_c}$ is the focal length of the given camera. Be noted that the covariance and information matrix are mutually inversed.

4.2. Adaptive M-estimator

The objective of the FGO is to minimize the summation of the residual function (20) to approach the optimal states set $\boldsymbol{\chi}$. Unfortunately, the non-linear function (20) is always a non-convex problem that has multiple sub-optimal, the local minimums. The outlier measurements, which dominate the overall residual, can easily lead to local minimum estimation. Instead of de-weighting the outlier measurement by tuning the covariance matrix, the M-estimator [54] is a promising technique that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

enhances the resilience of the optimizer by using an additional robust function. However, the M-estimator relies heavily on parameter tuning. Figure 7 shows the state-of-the-art Huber-based M-estimator with different parameters based on (27). In fact, the curvature of the M-estimator relies heavily on different k value, which is related to the robustness for outlier measurements. The smaller is k (black curve in Figure 7), the milder is the curve and the more robust it is meanwhile. However, a k with too smaller value can lead to an extremely small gradient of the error function (20), making the optimizer difficult to approach the optimal state. The researches in [33, 34] show that extensive parameter tuning is required to obtain satisfactory performance using M-estimator.

$$\rho(r) = \begin{cases} \frac{1}{2}r^2, & |r| \le k \\ k\left(|r| - \frac{1}{2}\mathrm{k}\right), & \text{otherwise} \end{cases} \tag{27}$$

where r denotes the residual measurement, k denotes the parameter that needs to be tuned and $\rho(*)$ represents the robust Huber function.

Different from the offline tuned M-estimator [55] and residual-based M-estimator [36] which relies on the initial guess of the state estimation, this paper proposes an adaptive M-estimator based on Huber function by correlating the parameters of M-estimator with the NTFT, which could correct the visual residual model to further mitigate the impacts of dynamic feature points. The parameter of the Huber function is estimated as follows at given epoch t:

$$\rho\left(\mathrm{r_C}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{cj}}, \boldsymbol{\chi}\right), N_{f,t,l}\right) = \begin{cases} \frac{1}{2}\left(\mathrm{r_C}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{cj}}, \boldsymbol{\chi}\right)\right)^2, & \mathrm{r_C}\left(\hat{Z}_l^{cj}, \chi\right) \le k \\ k_t\left(\left|\mathrm{r_C}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{cj}}, \boldsymbol{\chi}\right)\right| - \frac{1}{2}k_t\right), & \text{otherwise} \end{cases} \tag{28}$$

$$k_t = M_s \ N_{f,t,l} \tag{29}$$

where $\mathrm{r_C}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{cj}}, \boldsymbol{\chi}\right)$ is the visual residual of feature l in the jth image. k_t is the parameter of Huber function. M_s is a scaling factor correlating $N_{f,t,l}$ and k_t which is pre-determined. Therefore, the adaptive M-estimator is derived based on (28).

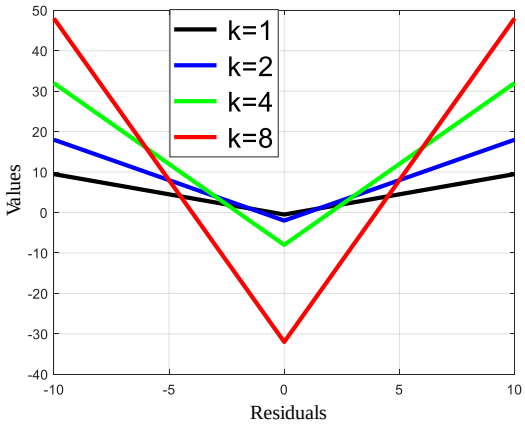


Figure 7. Huber M-estimator with different coefficients

4.3. Visual-Inertia Optimization with Online Sensor Model

Based on the derived VINS system in Section 3 and the online sensor model in Sections 4.1 and 4.2, the new optimization residual function is as follows,

$$\min_{\boldsymbol{\chi}} \left\{ \left\| \mathbf{r}_{\boldsymbol{p}} - \mathbf{H}_{\boldsymbol{p}}\boldsymbol{\chi} \right\|^2 + \sum_{k\epsilon\mathcal{B}} \left\| \mathrm{r}_{\mathcal{B}}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{b_{k+1}}}^{\boldsymbol{b_k}}, \boldsymbol{\chi}\right) \right\|_{\mathbf{P}_{\boldsymbol{b_{k+1}}}^{\boldsymbol{b_k}}}^2 + \sum_{(l,j)\epsilon\mathcal{C}} \left\| \rho\left(\mathrm{r_C}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{l}}^{\boldsymbol{cj}}, \boldsymbol{\chi}\right), N_{f,t,l}\right) \right\|_{\boldsymbol{\Sigma}_{\boldsymbol{l}}^{\boldsymbol{cj}}}^2 \right\} \tag{30}$$

where $\rho(*)$ is the robust Huber function. $\boldsymbol{\Sigma}_{\boldsymbol{l}}^{\boldsymbol{cj}}$ is the adaptive information matrix.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

5. Experimental Results

5.1. Experimental Setup

To validate the effectiveness of the proposed method, two experiments were conducted in typical urban canyons of Hong Kong. The experiments setup is shown in the left of Figure 8. An Xsens Mti 10 IMU is employed to collect raw measurements at a frequency of 200 Hz. A monocular camera (BFLY-U3-23S6C-C) is employed to collect colored images. In addition, the NovAtel SPAN-CPT, a GNSS (GPS, GLONASS, and Beidou) RTK/INS (with fiber-optic gyroscopes) integrated navigation system, is used to provide the ground truth of positioning. The gyro bias in-run stability of the FOG is 1 degree per hour and its random walk is 0.067 degree per hour. The baseline between the rover and GNSS base station is about 7 km. All the data were collected and synchronized based on the time stamp provided by the robot operation system (ROS) [56]. The coordinate systems between all the sensors were calibrated before the experiment. The Figure 8 (a) and (b) show the two tested urban canyons. The tested scenarios contain static buildings and dynamic objects, and the tested urban canyon 2 is more challenging with numerous dynamic objects than that in the tested urban canyon 1.

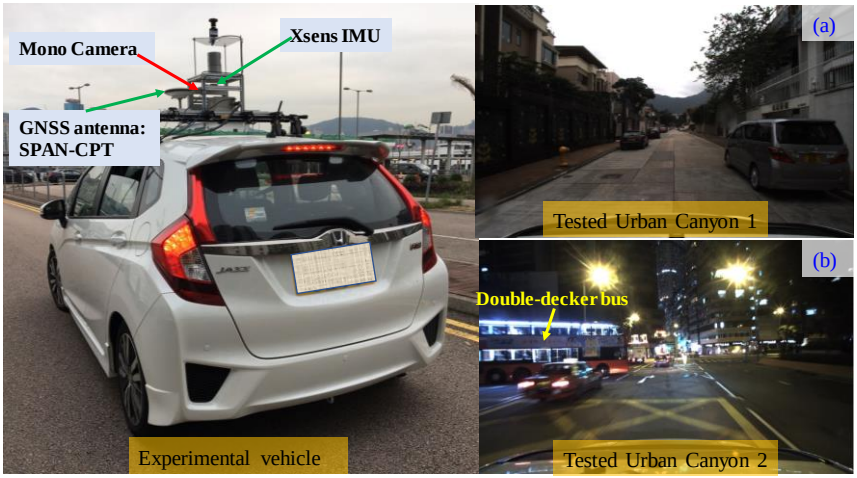


Figure 8. Sensor setup and the examples of tested scenarios

To verify the performance of the proposed method, several methods are compared.

(1) **VINS** [5]

(2) VINS-Adaptive Covariance (**VINS-AC**)

   a. VINS-Adaptive Covariance ($g_{f,t}$)

   b. VINS-Adaptive Covariance ($g_{f,t}, N_{f,t,l}$)

(3) VINS-Adaptive Covariance- Adaptive M-estimator (**VINS-AC-ME**)

The performance of the VINS is evaluated using the EVO toolkit [57], which is extensively used for the evaluation of odometry and SLAM. The parameters used in the experiments are shown in Table 1.

TABLE I. PARAMETER VALUES USED IN THIS PAPER

Window Size (n)	10
F_c	460
s	0.02
M_s	0.02

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

5.2. Evaluation of the Data Collected in Urban Canyon 1

We firstly conduct an experiment in urban canyon 1 to validate the performance of the proposed method. The positioning results for the listed methods are shown in Table II. The mean error of VINS is 0.33 meters with the maximum error reaching 1.84 meters. After considering the geometry distribution of features ($g_{f,t}$) into the adaptive covariance, the positioning error decreases to 0.32 meters. The positioning error is slightly mitigated by 3.03%. Furthermore, the number of feature tracking times ($N_{f,t,l}$) is also considered into the adaptive covariance, and the mean error decreases to 0.30 meters with the improvement of 9.09%. With the help of adaptive M-estimator, the mean error stays 0.30 meters with the maximum error decreasing to 1.44 meters. The positioning error of proposed method (VINS-AC-ME) is slightly mitigated by 9.09%. The result shows that the environment with few dynamic objects is more likely beneficial to the positioning accuracy of VINS.

TABLE II. POSITIONING PERFORMANCE COMPARISON BETWEEN THE LISTED METHODS BASED ON THE DATA COLLECTED IN URBAN CANYON 1

All data	**VINS**	**VINS-AC** ($g_{f,t}$)	**VINS-AC** ($g_{f,t}, N_{f,t,l}$)	**VINS-AC-ME**
Mean error	0.33 m	0.32 m	0.30 m	0.30 m
Std	0.31m	0.30 m	0.30 m	0.29 m
Max error	1.84 m	1.35 m	1.70 m	1.44 m
Improvement		3.03%	9.09%	9.09%

The trajectories of the listed methods and the reference are shown in Figure 9. The total length of the trajectory is 760 meters. Overall, the trajectory from VINS-AC-ME (blue curve) is the one closest to the reference trajectory (black curve). The relative positioning error throughout the test is shown in Figure 10. The accuracy of the proposed method is slightly improved with the help of the proposed online sensor model adaption. This is due to the fact that the static feature points dominate the visual measurements in the urban canyon 1, therefore our proposed method still has limitation in the environment like the urban canyon 1.

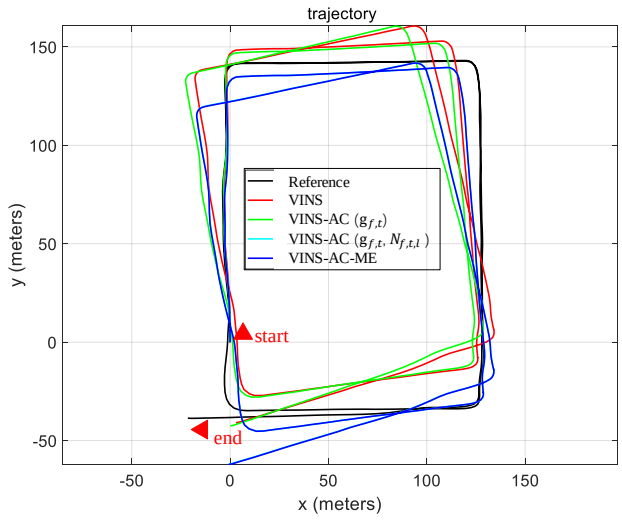


Figure 9. The trajectories of the VINS, proposed method and reference in urban canyon 1

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

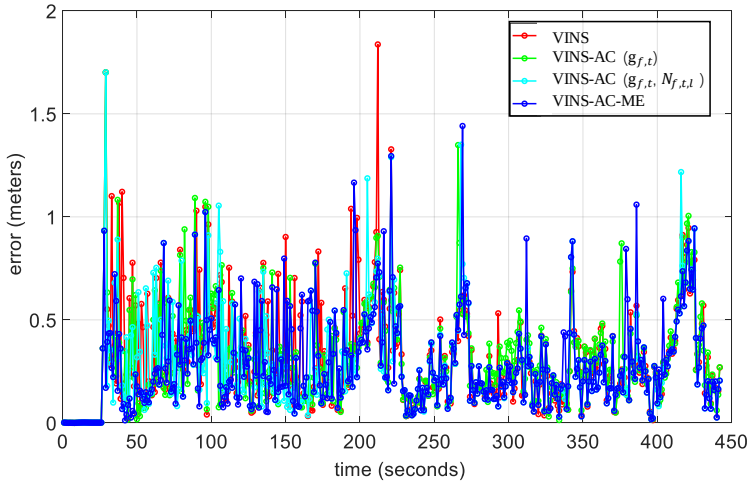


Figure 10. Relative positioning errors of the tested method in urban canyon 1

5.3. Evaluation of the Data Collected in Urban Canyon 2

To challenge the performance of the proposed method, we conduct the other experiment in urban canyon 2 with numerous dynamic objects and the data is collected in the night (see Figure 8 (b)). In this case, the number of DFP is far more than the one in urban canyon 1. Therefore, we believe that the urban canyon 2 is more challenging to test our proposed online sensor model compared with the test in urban canyon 1.

The positioning results for the listed methods are shown in Table III. The mean error of VINS is 0.79 meters with the maximum error reaching 5.58 meters. The mean error decreases to 0.69 meters after considering the geometry distribution of features ($g_{f,t}$) into the adaptive covariance. The number of feature tracking times ($N_{f,t,l}$) is also considered into the adaptive covariance, and the mean error decreases to 0.64 meters with the improvement of 18.99%. With the help of adaptive M-estimator, the mean error decreases to 0.59 meters with the improvement of 25.32%.

TABLE III. POSITIONING PERFORMANCE COMPARISON BETWEEN THE LISTED METHODS BASED ON THE DATA COLLECTED IN URBAN CANYON 2

All data	**VINS**	**VINS-AC** ($g_{f,t}$)	**VINS-AC** ($g_{f,t}$, $N_{f,t,l}$)	**VINS-AC-ME**
Mean error	0.79 m	0.69 m	0.64 m	0.59 m
Std	0.96m	0.86 m	0.84 m	0.75 m
Max error	5.58 m	6.39m	7.32 m	7.26 m
Improvement		12.66%	18.99%	25.32%

Figure 11 shows the trajectories of the listed methods and the reference. The total length of the trajectory is 1502 meters. We can see that the proposed method (blue curve) is the one closest to the reference trajectory (black curve). The detail of the relative positioning error is shown in Figure 12. To show the detail of the improvement, the four epochs are selected in Figure 12, and the snapshots of selected epochs are shown in Figure 13. The corresponding positioning errors are shown in Table IV. We can see from Table IV that the error of VINS reaches the maximum value 5.59 meters at the epoch 260 (A). After considering the geometry of features distribution ($g_{f,t}$) into the adaptive covariance, the error of VINS-AC ($g_{f,t}$) decreases to 2.81 meters. Moreover, the number of feature tracking times ($N_{f,t,l}$) is also introduced to the adaptive covariance, and the error of VINS-AC ($g_{f,t}$, $N_{f,t,l}$) decreases to 1.62 meters, which shows that the $g_{f,t}$ and $N_{f,t,l}$ can model the uncertainty of each feature measurements to improve the performance of VINS. With the help of adaptive M-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

estimator, the error of proposed method (VINS-AC-ME) decreases to 1.02 meters. Similar condition appears on the epoch 343 (B). Compared the VINS, the proposed method can obtain outperformance in positioning accuracy. Based on the proposed adaptive covariance (26), the $g_{f,t}$ and $N_{f,t,l}$ are used to evaluate the uncertainty of visual measurements, as the Figures 14 and 15 show that the system tends to rely on the visual measurement at epoch 260 (A) and epoch 343 (B). However, we find that the proposed method leads to large positioning error at epoch 29 (C). The error of VINS-AC ($g_{f,t}$) increases to 6.39 meters, and the error of VINS-AC ($g_{f,t}$, $N_{f,t,l}$) even increases to 7.26 meters. The error of VINS-AC-ME is also 7.26 meters. This is due to the fact that the proposed system tends to give a higher weighting on the visual measurement, while the quality of the track features is poor at the epoch. As can be seen from Figure 13 (c), the feature tracked on a blurred image. Therefore, the positioning error increases significantly at epoch 29 (C). Interestingly, the error of VINS-AC ($g_{f,t}$, $N_{f,t,l}$) can reach 7.32 meters at epoch 127 (D), which is far larger than that of VINS-AC ($g_{f,t}$) (1.43 meters). The error is mainly caused by other factors, such as the unstable illumination conditions. With the help of adaptive M-estimator, the error of VINS-AC-ME decreases to 1.32 meters, which shows that the adaptive M-estimator can enhance the resistance to the outliers ,as Figure 16 shows that the adaptive M-estimator can correct the visual residual model by using an additional robust function, especially in the challenging urban canyons.

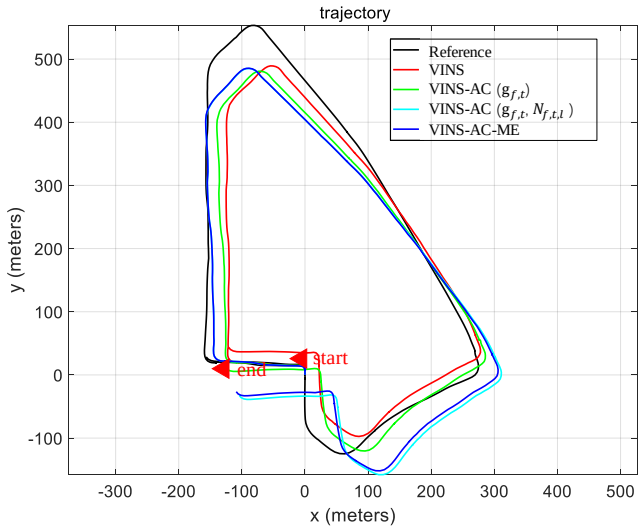


Figure 11. The trajectories of the VINS, proposed method and reference in urban canyon 2

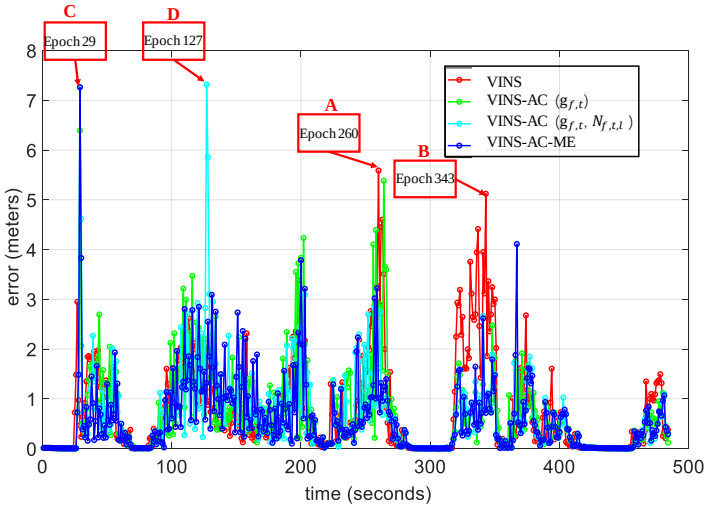


Figure 12. Relative positioning errors of the tested method in urban canyon 2

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

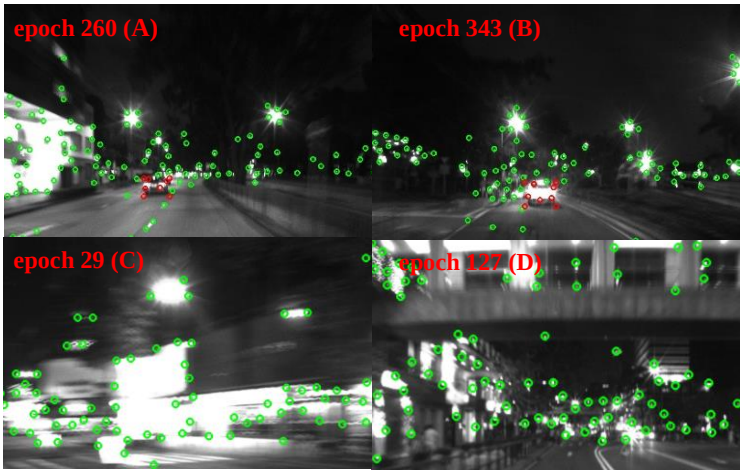


Figure 13. The images of the tested urban canyon 2 in the four selected epochs with respect to Figure 12. The green circle denotes the static feature points, and the red circle denotes the dynamic feature points.

TABLE IV. POSITIONING PERFORMANCE COMPARISON BETWEEN THE LISTED METHODS ON THE FOUR SELECTED EPOCHS IN FIGURE 12

Mean error	VINS	VINS-AC ($g_{f,t}$)	VINS-AC ($g_{f,t}, N_{f,t,l}$)	VINS-AC-ME
Epoch 260 (A)	5.59 m	2.81 m	1.62 m	1.02 m
Epoch 343 (B)	5.12 m	0.98 m	0.78 m	0.72m
Epoch 29 (C)	0.47 m	6.39m	7.26 m	7.26 m
Epoch 127 (D)	1.65 m	1.43 m	7.32 m	1.32 m

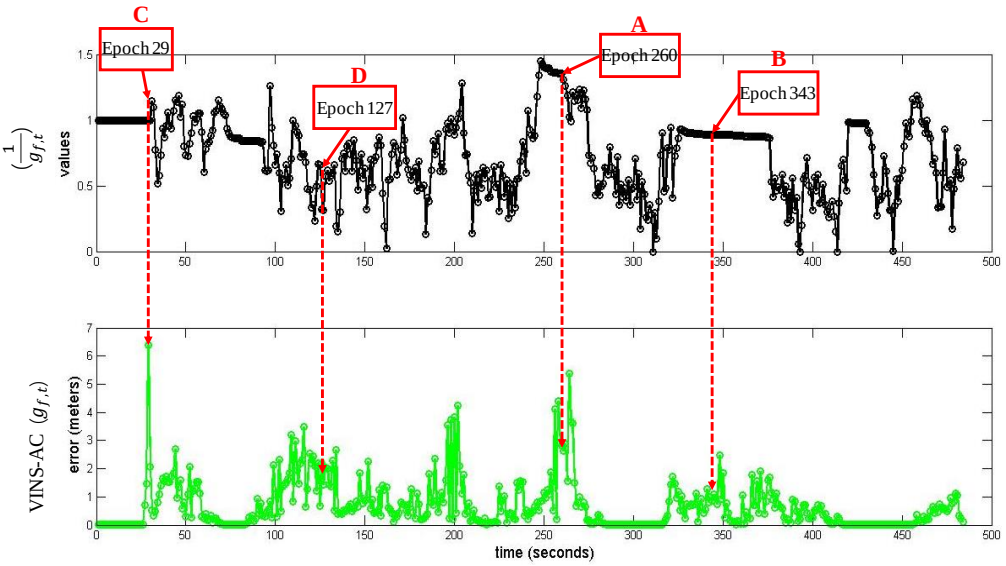


Figure 14. (Top) Geometry of feature distribution ($g_{f,t}$) on the error of the VINS-AC ($g_{f,t}$) (Bottom) Error of the VINS-AC ($g_{f,t}$).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

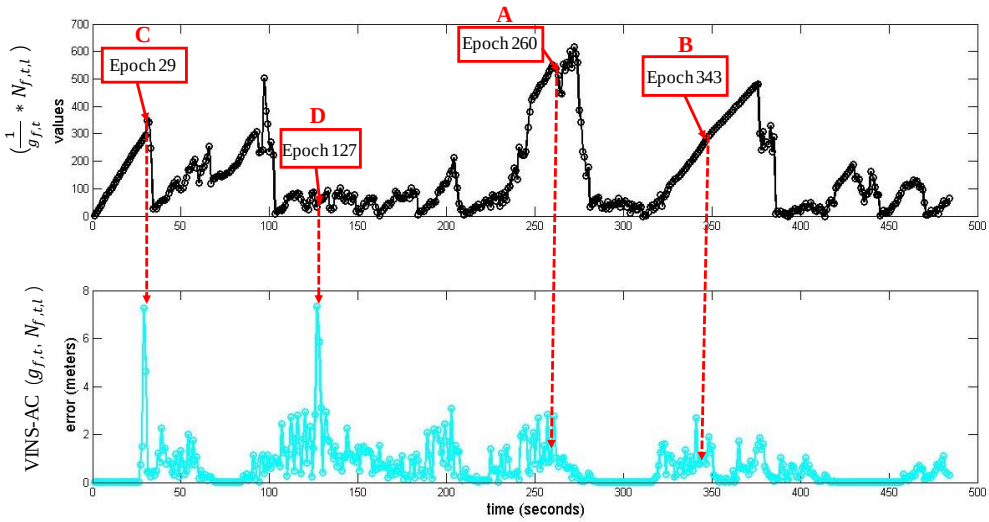


Figure 15. (Top) Geometry of feature distribution ($g_{f,t}$) and the number of feature tracking times ($N_{f,t,l}$) on the error of the VINS-AC ($g_{f,t}$, $N_{f,t,l}$). (Bottom) Error of the VINS-AC ($g_{f,t}$, $N_{f,t,l}$).

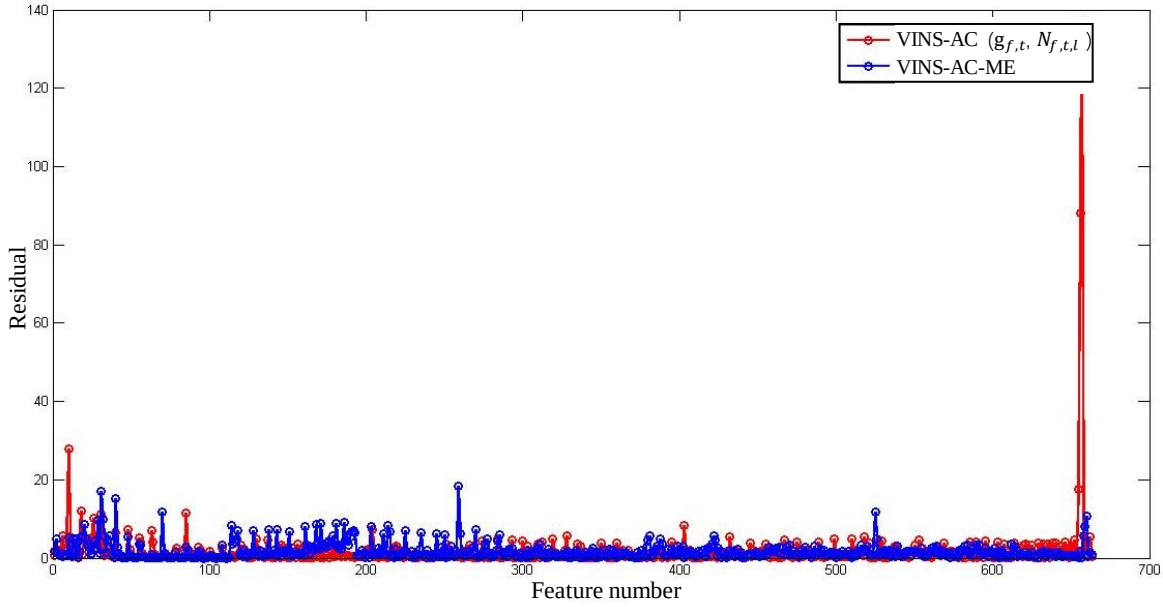


Figure 16. The residual comparison between VINS-AC ($g_{f,t}$, $N_{f,t,l}$) and VINS-AC-ME at epoch 127 (D). The red curve denotes the visual residual on the VINS-AC ($g_{f,t}$, $N_{f,t,l}$) and the blue curve denotes the residual on the VINS-AC-ME.

6. Conclusions and Future Work

The dynamic environment is very challenging for the autonomous ground vehicles in urban canyons. In order to model the uncertainty of visual measurements and improve the system resistance to outliers, this paper proposes to adapt the adaptive covariance and adaptive M-estimator to evaluate the performance of VINS. The accuracy is improved both in the two experiments, especially in the urban canyon 2, which shows the effectiveness of the proposed method.

Actually, this paper only contributes to mitigate the effects of DFP. The remaining error is mainly caused by other factors, such as the unstable illumination conditions, failure of feature extraction. In the future, we will further study how to acquire and to estimate the quality of the detected features for VINS positioning in urban canyons.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

Author Contributions: Conceptualization, X.W.B. and L.-T.H.; methodology, X.W.B.; software, X.W.B.; formal analysis, X.W.B., W.S.W.; data collection, X.W.B. and W.S.W.; writing—original draft preparation, X.W.B.; writing—review and editing, X.W.B., W.S.W., L.-T.H.; supervision, L.-T.H.

Funding: This research is funded by Hong Kong Polytechnic University. The project ZVKZ is "Positioning and Navigation for Autonomous Driving Vehicle by Sensor Integration"

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bloesch, M., et al. *Robust visual inertial odometry using a direct EKF-based approach*. in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 2015. IEEE.
2. Li, R., et al. *LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments*. in *2014 DGON Inertial Sensors and Systems (ISS)*. 2014. IEEE.
3. Qin, T., et al., *A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors.* arXiv preprint arXiv:1901.03638, 2019.
4. Pfrommer, B., et al. *Penncosyvio: A challenging visual inertial odometry benchmark*. in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017. IEEE.
5. Qin, T., P. Li, and S. Shen, *Vins-mono: A robust and versatile monocular visual-inertial state estimator.* IEEE Transactions on Robotics, 2018. **34**(4): p. 1004-1020.
6. Von Stumberg, L., V. Usenko, and D. Cremers. *Direct sparse visual-inertial odometry using dynamic marginalization*. in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018. IEEE.
7. Xu, W., D. Choi, and G. Wang, *Direct visual-inertial odometry with semi-dense mapping.* Computers & Electrical Engineering, 2018. **67**: p. 761-775.
8. Rebecq, H., T. Horstschaefer, and D. Scaramuzza. *Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization*. in *BMVC*. 2017.
9. Saputra, M.R.U., A. Markham, and N. Trigoni, *Visual SLAM and structure from motion in dynamic environments: A survey.* ACM Computing Surveys (CSUR), 2018. **51**(2): p. 37.
10. X. Bai, W.W., and L.-T. Hsu, *Performance Analysis of Visual/Inertial Integrated Positioning in Diverse Typical Urban Scenarios of Hong Kong*, in *Asian-Pacific Conference on Aerospace Technology and Science*. 2019: Taiwan.
11. Qin, T., P. Li, and S.J.I.T.o.R. Shen, *Vins-mono: A robust and versatile monocular visual-inertial state estimator.* 2018. **34**(4): p. 1004-1020.
12. Sun, Y., M. Liu, and M.Q.-H. Meng, *Improving RGB-D SLAM in dynamic environments: A motion removal approach.* Robotics and Autonomous Systems, 2017. **89**: p. 110-122.
13. Sun, Y., M. Liu, and M.Q.-H. Meng, *Motion removal for reliable RGB-D SLAM in dynamic environments.* Robotics and Autonomous Systems, 2018. **108**: p. 115-128.
14. Wang, Y. and S. Huang. *Motion segmentation based robust RGB-D SLAM*. in *Proceeding of the 11th World Congress on Intelligent Control and Automation*. 2014. IEEE.
15. Herbst, E., X. Ren, and D. Fox. *Rgb-d flow: Dense 3-d motion estimation using color and depth*. in *2013 IEEE International Conference on Robotics and Automation*. 2013. IEEE.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

16. Mur-Artal, R. and J.D. Tardós, *Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras.* IEEE Transactions on Robotics, 2017. **33**(5): p. 1255-1262.
17. Endres, F., et al. *An evaluation of the RGB-D SLAM system*. in *Icra*. 2012.
18. Yamaguchi, K., T. Kato, and Y. Ninomiya. *Vehicle ego-motion estimation and moving object detection using a monocular camera*. in *18th International Conference on Pattern Recognition (ICPR'06)*. 2006. IEEE.
19. Zhou, D., et al. *On modeling ego-motion uncertainty for moving object detection from a mobile platform*. in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014. IEEE.
20. Milz, S., et al. *Visual slam for automated driving: Exploring the applications of deep learning*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.
21. Bahraini, M.S., A.B. Rad, and M. Bozorg, *SLAM in Dynamic Environments: A Deep Learning Approach for Moving Object Tracking Using ML-RANSAC Algorithm.* Sensors, 2019. **19**(17): p. 3699.
22. Zhong, F., et al. *Detect-SLAM: Making object detection and slam mutually beneficial*. in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018. IEEE.
23. Bescos, B., et al., *DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes.* IEEE Robotics and Automation Letters, 2018. **3**(4): p. 4076-4083.
24. Xiao, L., et al., *Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment.* Robotics and Autonomous Systems, 2019. **117**: p. 1-16.
25. Liu, W., et al. *Ssd: Single shot multibox detector*. in *European conference on computer vision*. 2016. Springer.
26. Labbe, M. and F. Michaud. *Online global loop closure detection for large-scale multi-session graph-based SLAM*. in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014. IEEE.
27. Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
28. Lin, T.-Y., et al. *Feature pyramid networks for object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
29. Belter, D., M. Nowicki, and P. Skrzypczyński. *Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features*. in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016. IEEE.
30. Denim, F., et al., *Cooperative Visual SLAM based on Adaptive Covariance Intersection.* Journal of Advanced Engineering and Computation, 2018. **2**(3): p. 151-163.
31. Demim, F., et al. *A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle*. in *2017 6th International Conference on Systems and Control (ICSC)*. 2017. IEEE.
32. Sünderhauf, N. and P. Protzel. *Switchable constraints for robust pose graph SLAM*. in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012. IEEE.
33. Pfeifer, T., S. Lange, and P. Protzel. *Dynamic Covariance Estimation—A parameter free approach to robust Sensor Fusion*. in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 2017. IEEE.
34. Watson, R.M. and J.N.J.a.p.a. Gross, *Robust navigation in GNSS degraded environment using graph optimization.* 2018.
35. Tyler, D.E.J.T.a.o.S., *A distribution-free $ M $-estimator of multivariate scatter.* 1987. **15**(1): p. 234-251.
36. Agamennoni, G., P. Furgale, and R. Siegwart. *Self-tuning M-estimators*. in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015. IEEE.
37. Lin, Y., et al., *Autonomous aerial navigation using monocular visual-inertial fusion.* Journal of Field Robotics, 2018. **35**(1): p. 23-51.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 March 2020 doi:10.20944/preprints202003.0018.v1

Peer-reviewed version available at Remote Sens. 2020, 12, 1686; doi:10.3390/rs12101686

38. Qiu, K., et al. *Estimating metric poses of dynamic objects using monocular visual-inertial fusion*. in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018. IEEE.
39. Hsu, L.-T., Y. Gu, and S. Kamijo, *NLOS correction/exclusion for GNSS measurement using RAIM and city building models.* Sensors, 2015. **15**(7): p. 17329-17349.
40. Wen, W., et al., *Tightly Coupled GNSS/INS Integration Via Factor Graph and Aided by Fish-eye Camera.* IEEE Transactions on Vehicular Technology, 2019.
41. X. Bai, W.W., L.-T. Hsu, H. Li, *Perception-aided Visual-Inertial Integrated Positioning in Dynamic Urban Areas (accepted)*, in *ION/IEEE PLANS*. 2020: Portland, Oregon.
42. Forster, C., et al. *IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation*. 2015. Georgia Institute of Technology.
43. Dellaert, F., M.J.F. Kaess, and T.i. Robotics, *Factor graphs for robot perception.* 2017. **6**(1-2): p. 1-139.
44. Groves, P.D., *Principles of GNSS, inertial, and multisensor integrated navigation systems*. 2013: Artech house.
45. Thrun, S.J.A.M., *Probabilistic algorithms in robotics.* 2000. **21**(4): p. 93-93.
46. Wen, W., et al., *Tightly Coupled GNSS/INS Integration Via Factor Graph and Aided by Fish-eye Camera.* 2019.
47. Forster, C., et al., *On-Manifold Preintegration for Real-Time Visual--Inertial Odometry.* IEEE Transactions on Robotics, 2016. **33**(1): p. 1-21.
48. Forster, C., et al., *Supplementary material to: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation*. 2015, Georgia Institute of Technology.
49. Shi, J. *Good features to track*. in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. 1994. IEEE.
50. Senst, T., V. Eiselein, and T. Sikora. *II-LK–a real-time implementation for sparse optical flow*. in *International Conference Image Analysis and Recognition*. 2010. Springer.
51. Zhang, F., *The Schur complement and its applications*. Vol. 4. 2006: Springer Science & Business Media.
52. Groves, P.D.J.I.A. and E.S. Magazine, *Principles of GNSS, inertial, and multisensor integrated navigation systems, [Book review].* 2015. **30**(2): p. 26-27.
53. Qin, T. and S. Shen. *Robust initialization of monocular visual-inertial estimation on aerial robots*. in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017. IEEE.
54. Lucas, A.J.C.i.S.-T. and Methods, *Robustness of the student t based M-estimator.* 1997. **26**(5): p. 1165-1182.
55. Li, W., et al., *A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles.* 2018. **23**(6): p. 724-732.
56. Quigley, M., et al. *ROS: an open-source Robot Operating System*. in *ICRA workshop on open source software*. 2009. Kobe, Japan.
57. Grupp, M., *evo: Python package for the evaluation of odometry and slam.* Note: https://github. com/MichaelGrupp/evo Cited by: Table, 2017. **7**.