Scientific Research Publishing

Open Journal of Applied Sciences, 2026, 16(2), 608-626
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Asynchronous Multi-Camera-IMU Pose Estimation Algorithm Based on Depth Confidence Optimization

Zhi Li, Guoliang Wei, Zhixuan Miao

Business School of University of Shanghai for Science and Technology, Shanghai, China
Email: lizhi10101@163.com, guoliang.wei@usst.edu.cn

How to cite this paper: Li, Z., Wei, G.L. and Miao, Z.X. (2026) Asynchronous Multi-Camera-IMU Pose Estimation Algorithm Based on Depth Confidence Optimization. *Open Journal of Applied Sciences*, **16**, 608-626.
https://doi.org/10.4236/ojapps.2026.162038

Received: January 26, 2026
Accepted: February 21, 2026
Published: February 24, 2026

Abstract

To address the limitations of traditional multi-camera-IMU state estimation systems—namely, insufficient localization accuracy in complex environments and poor robustness under abnormal IMU observations—this paper proposes an asynchronous multi-camera-IMU tightly-coupled navigation algorithm that incorporates a depth confidence scoring strategy and a chi-square test mechanism. The proposed method is built upon the factor graph optimization (FGO) framework. On the basis of conventional multi-camera-IMU fusion, a gated recurrent unit (GRU)-based depth confidence scoring model is introduced to mitigate uncertainty and partially suppress noise errors. These confidence scores are then utilized as weights in the factor graph to minimize the weighted sum of squared residuals, thereby improving the accuracy of the optimized state variables. Furthermore, a dynamic outlier rejection strategy based on chi-square testing is designed, which operates through three stages—pre-screening, initialization, and iterative statistical optimization—to preferentially utilize well-distributed and high-quality observation subsets while effectively excluding noise-sensitive measurements, thus strengthening the 6-DoF pose constraints. Extensive evaluations are conducted using indoor simulation scenarios and the public VINS-Multi dataset (sequence raw_515_435). The results demonstrate that, after incorporating the depth confidence scores, the pose uncertainty in asynchronous settings is reduced from an error range of 0.1 - 0.2 to within 0.05. Across all sequences in the dataset, the proposed method achieves an average improvement of 31.6% in localization accuracy, with particularly significant gains of up to 46.8% in challenging low-texture regions and scenarios with large motion magnitudes.

Keywords

Multi-Camera, Confidence Scoring, Chi-Square Test, Indoor Localization,

Gated Recurrent Unit, Factor Graph

1. Introduction

With the rapid advancement of robotics and autonomous navigation technologies, asynchronous multi-camera-IMU state estimation systems are facing increasingly stringent requirements for accuracy and robustness in applications such as unmanned aerial vehicles (UAVs) and indoor localization [1]. In indoor environments with urban buildings, areas with dynamic interference, or under sensor failure conditions, multi-camera signals are prone to severe degradation due to occlusion, lack of texture, or asynchronous temporal distortion, resulting in a substantial drop in localization accuracy. When visual measurements are lost, the system loses inter-frame relative pose constraints and degenerates into pure inertial navigation mode [2]. In such cases, IMU preintegration errors accumulate at a significantly higher rate, severely compromising the safety and robustness of the overall navigation system. Therefore, developing methods to enhance the global localization accuracy of asynchronous multi-camera-IMU systems under diverse challenging conditions has become a critical and urgent problem in the field of navigation.

Current research on asynchronous multi-camera-IMU systems primarily focuses on multi-sensor fusion and optimization techniques [3]. Several studies have achieved deep integration of raw multi-camera measurements with IMU data, thereby improving localization accuracy and robustness in complex environments [4] [5]. For example, the tightly-coupled multi-camera-IMU-LiDAR odometry framework proposed in [4] unifies multi-view photometric transfer with ESIKF fusion, jointly modeling visual, inertial, and LiDAR measurements to significantly enhance both accuracy and robustness in challenging scenes. However, this approach does not tightly incorporate pixel-level depth or other raw information, resulting in limited data reliability. The fusion framework presented in [5] leverages factor graph-based nonlinear optimization to jointly incorporate multi-camera observations, IMU preintegration, and depth cues, demonstrating greater robustness and accuracy compared to the former. Nevertheless, experiments in [5] indicate that raw visual measurements remain susceptible to geometric constraints and noise, making it difficult for visual and inertial measurements to fully meet the demands of high-precision scenarios.

Another line of research explores the integration of real-time deep learning techniques into traditional SLAM or visual-inertial optimization frameworks to improve localization performance and robustness in complex scenes [6]-[8]. These methods typically rely on deep neural networks for feature matching, pose regression, or scene understanding, exhibiting strong perception capabilities. However, most existing approaches treat deep learning modules merely as a front-end component or as an independent constraint. Their outputs are usually incorporated into the back-end optimization with fixed inference weights, lacking ex-

plicit modeling of measurement uncertainty in visual observations. Consequently, these methods struggle to dynamically reflect changes in the reliability of deep model inferences during optimization, fail to quantitatively assess model confidence, and prevent the back-end optimizer from adaptively reweighting features according to their quality, which can easily introduce erroneous constraints.

To address the aforementioned limitations, this paper proposes an asynchronous multi-camera-IMU navigation algorithm that incorporates a depth confidence scoring strategy and a dynamic IMU outlier rejection mechanism based on chi-square testing. The depth confidence score is derived from pixel-level inverse depth observations obtained via a pretrained visual model, along with the covariance matrix that reflects the reliability of both depth and optical flow estimation, which is then used to construct a confidence scoring model. Meanwhile, the chi-square test is employed to evaluate the range of abnormal deviations in IMU observation distributions. By jointly processing camera observations and IMU measurements from asynchronous sensor streams, the proposed algorithm computes confidence scores to mitigate uncertainty and suppress partial noise errors, thereby improving the reliability of observation data with moderate computational overhead. Furthermore, a dynamic outlier rejection strategy based on the chi-square test is designed: it first computes the squared Mahalanobis distance of residuals as the test statistic, then constructs the corresponding chi-square critical value according to the degrees of freedom m and a chosen significance level α. Observations whose residual statistic exceeds the critical value are classified as outliers and rejected. The remaining reliable measurements are subsequently used for state optimization. By tightly integrating depth confidence scores, visual observations, and IMU data within the factor graph optimization framework, the proposed method achieves superior localization accuracy and robustness in complex environments, offering a novel solution to enhance the practical performance of asynchronous multi-camera-IMU navigation systems.

2. Related Theories

2.1. Factor Graph Optimization Framework

The essence of factor graph optimization is to solve the maximum a posteriori (MAP) estimation problem. Given a set of measurement data $\boldsymbol{Z}$, the goal is to find the system state χ that maximizes the posterior probability [9]:

$$\chi^* = \arg\max_{\chi} P\left(\chi \middle| Z\right) \tag{1}$$

Assuming all measurements are independent and the noise follows a zero-mean Gaussian distribution, this problem can be equivalently transformed into minimizing the sum of a series of cost (loss) functions. The mathematical formulation is:

$$\begin{aligned} \chi^* &= \arg\max_{\chi} P(\chi) \prod_{i=1}^{n} P\left(Z_i \middle| \chi\right) \\ &= \arg\min_{\chi} \left\{ \left\| r_p - H_p \chi \right\|^2 + \sum_{i=1}^{n} \left\| r\left(Z_i \middle| \chi\right) \right\|_{P_i}^2 \right\} \end{aligned} \tag{2}$$

where $\boldsymbol{Z}$ denotes the observations from n independent sensors, r_{p} and H_{p} encapsulate the prior information of the system state, r represents the residual function for each measurement, and $\|\cdot\|_{P_i}^2$ denotes the Mahalanobis norm. This formulation decomposes the optimization problem into independent factors, where each factor encodes the probabilistic relationship between a subset of state variables and the corresponding measurement, thereby enabling efficient joint optimization.

The core of asynchronous multi-camera-IMU state estimation lies in the factor graph optimization framework, which constructs the state vector by integrating visual and inertial measurements:

$$\begin{aligned} \lambda &= \left[x_0, x_1, \cdots, x_n, \lambda_0, \lambda_1, \cdots, \lambda_l, x_{c0}, x_{c1}, \cdots, x_{cN-1}\right] \\ x_i &= \left[p_i^w, v_i^w, R_i^w, b_a, b_g\right], i \in [0, n] \\ x_k^c &= \left[p_{c_k}^b, R_{c_k}^b, t_{d_k}\right], k \in [0, N-1] \end{aligned} \tag{3}$$

where x_i denotes the system state at time stamp i, λ_j represents the inverse depth of the j feature, and x_k^c denotes the camera-to-IMU extrinsic transformation. The optimization objective is formulated as:

$$\hat{X} = \min_X \left\{ \left\| e_p - H_p X \right\|_{p_p}^2 + \sum_{k \in \mathcal{B}} e_B \left\| \left(z_k^{l+1}, X\right) \right\|_{p_k^{l+1}}^2 + \sum_{(i,j) \in \mathcal{C}} e_C \left\| \left(z_j^l, X\right) \right\|_{p_j^l}^2 \right\} \tag{4}$$

where e_P is the prior factor residual, e_B is the IMU preintegration factor residual, e_C is the visual factor residual, and p_p, p_k, p_j are the corresponding covariance matrices of the respective factors.

This framework organizes incoming camera frames in chronological order within a sliding window according to their timestamps, enabling support for asynchronous inputs and heterogeneous camera types. Visual factors incorporate depth information from RGB-D cameras through reprojection error residuals, while IMU factors provide high-frequency pose propagation via preintegration. Together, they ensure that the system maintains robustness even during temporary visual outages or complete camera failure.

2.2. Gated Recurrent Unit Network

The algorithm in this paper utilizes the DROID-SLAM pre-trained model, which is a deep learning-based visual SLAM system model trained on the TartanAir synthetic dataset [10]. It supports monocular, stereo, and RGB-D inputs. Its architecture combines CNN for feature extraction and GRU for iterative optimization updates, outputting image depth iterative residuals.

The gated recurrent unit (GRU) serves as a variant of recurrent neural networks designed for processing sequential data, proposed by Cho in 2014 [11]. It aims to address the issues of vanishing or exploding gradients commonly encountered in traditional RNNs when handling long sequence data. GRU introduces two gating mechanisms—update gate and reset gate—to efficiently capture sequence dependencies and reduce computational complexity. The update gate controls the

retention degree of the previous time step's state information, with the formula:

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \tag{5}$$

where z_t represents the update gate, σ is the sigmoid activation function, W_z is the weight matrix, h_{t-1} is the hidden state from the previous time step, and x_t is the current input. The reset gate determines the combination degree of the previous time step's state with the current input, with the formula:

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \tag{6}$$

Based on the reset gate, the candidate hidden state is computed as:

$$\tilde{h}_t = \tanh\left(W_h \cdot \left[r_t \odot h_{t-1}, x_t\right]\right) \tag{7}$$

This structure enables GRU to selectively remember or forget information in memory. Compared to long short-term memory (LSTM) networks, it has fewer parameters due to the absence of a forget gate and an output gate, resulting in higher computational efficiency. In visual SLAM systems such as DROID-SLAM, GRU is commonly used to model temporal optical flow and depth estimation in time series, outputting image depth iterative residuals to provide dynamic constraints for subsequent optimization [8]. Its advantages lie in robustness for non-long sequences and low-latency characteristics suitable for real-time applications. However, for semi-real-time continuous capabilities in complex environments, it may require further LSTM.

In this work, we directly adopt the officially released pre-trained weights of DROID-SLAM trained on the TartanAir synthetic dataset [10] without any fine-tuning specific to the tasks or test environments in this study. This strategy of using fixed pre-trained weights ensures the simplicity and reproducibility of the method while leveraging the diversity of the TartanAir dataset which includes a wide range of lighting conditions, texture levels, camera motion patterns, and indoor/outdoor scenes, endowing the model with strong cross-dataset generalization capability. The experimental results validate this generalization performance: on indoor simulation sequences, 01_SIM and 02_SIM, and the real-world raw_515_435 dataset—which differ from the training data distribution—the direct application of the pre-trained model achieves significant improvements in localization accuracy without requiring additional domain adaptation. This further enhances the practicality and reliability of the proposed method, making it suitable for real-world deployments where data distributions are unknown.

3. Algorithm Framework and Overview

3.1. Coordinate System Definitions

Accurate definitions and transformations of multiple coordinate systems form the foundation for achieving high-precision asynchronous multi-camera-IMU state estimation. The coordinate systems used in this paper are illustrated in **Figure 1**, where the VIO local world frame is defined as {L}, with the z-axis aligned with the gravity direction. VIO performs state estimation using IMU and visual observa-

tions, computing the continuous poses from the IMU body frame to the local world frame at different timestamps, which constitutes the trajectory estimated by VIO. To enable effective fusion of multi-camera and IMU data, camera observations and IMU measurements need to be projected into the local world frame {L}, whose origin is determined at system initialization and whose orientation is aligned with the gravity direction. The camera frame is defined as {C}. This system is capable of integrating asynchronous multi-source sensor data, thereby ensuring the algorithm's accuracy and robustness.

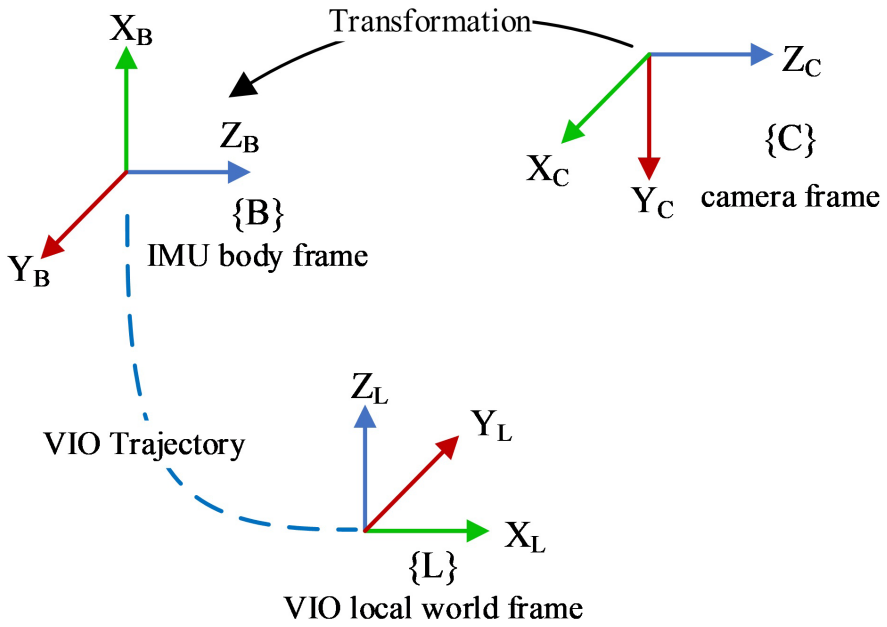

Figure 1. Schematic diagram of system coordinate systems.

3.2. System Framework

The asynchronous multi-camera-IMU state estimation system framework proposed in this paper is based on the theoretical foundation of VINS-Multi, with extensions and optimizations to its architecture to meet the high-precision demands in complex environments. The overall system architecture is illustrated in Figure 2, comprising three main modules: Parallel Front-Ends, Front-End Coordinator, and Back-End Optimization Module. On this basis, innovative designs from this paper are incorporated, including the depth confidence scoring strategy and the chi-square test mechanism, to enhance the system's robustness and accuracy. The following provides a detailed description of each module.

3.2.1. Parallel Front-End

The parallel front-end module operates an independent front-end thread for each camera module, adopting a standardized pipeline similar to that in VINS-Mono [12] and VINS-Fusion [13], which includes feature detection, tracking, and outlier rejection. Each front-end processes image inputs from a single camera supporting monocular, stereo, or RGB-D types. For RGB-D cameras, additional depth infor-

mation is handled. Feature extraction employs classical methods such as FAST corner detection combined with optical flow tracking such as the KLT algorithm, and outliers are rejected using RANSAC [14] to ensure the reliability and consistency of feature points. The IMU front-end is responsible for preintegrating raw accelerometer and gyroscope data, providing initial pose estimates at a high frequency of 500 Hz based on the latest optimization results. Compared to VINS-Multi, this paper introduces pixel-level confidence weights from the DROID-SLAM pretrained model as auxiliary inputs in the front-end, dynamically evaluating the reliability and depth quality of each camera frame, thereby laying the foundation for the subsequent depth confidence scoring strategy.

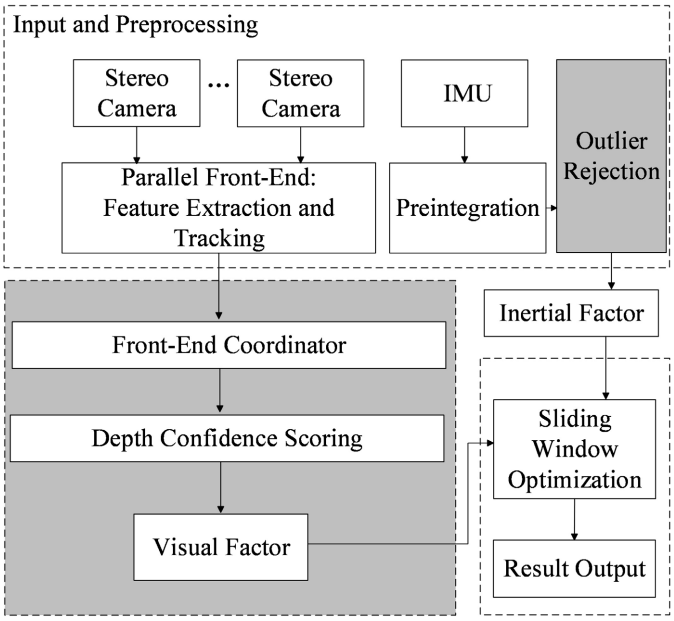

Figure 2. Algorithm flowchart.

3.2.2. Front-End Coordinator

The front-end coordinator is responsible for coordinating the outputs from multiple parallel front-ends and integrating them into a unified system state. Considering the dynamic feature quantity allocation and frame priority mechanisms in VINS-Multi, this paper optimizes the processing of the front-end coordinator. The feature quantity allocation dynamically adjusts the maximum number of features f_{ni} for each camera based on the number of tracked features t_{fni} from the previous frame with the calculation formula:

$$f_{ni} = \frac{t_{fni}}{\sum_{i=0}^{N} t_{fni}} \cdot FN \tag{8}$$

where FN is the total maximum feature number, and $\sum_{i=0}^{N} t_{fni}$ is the sum of tracked features for all camera frames. Through this approach, computational resources are allocated to cameras with a larger number of tracked features, thereby

optimizing resource utilization. The frame priority coordination comprehensively considers the feature priority P_{fi} and the time interval priority P_{ti}, defined respectively as:

$$P_{fi} = \frac{f_{ni}}{FN} \tag{9}$$

$$P_{ti} = e^{-k\delta t_i} \tag{10}$$

where δt_i is the time interval since the last received camera frame, and k is a positive constant. The coordinator selects the frame with the highest P_{fi} or P_{ti} to forward to the back-end optimization, thereby prioritizing frames with the longest time intervals or highest confidence to maintain the system's real-time performance.

Innovatively, this paper integrates the w_{ij} output from DROID-SLAM in the front-end coordinator to compute image confidence values $C(u,v)$, which serve as additional weights for feature reweighting, further optimizing feature selection and tracking based on the feature characteristics in low-confidence regions.

3.2.3. Back-End Optimization

The backend optimization module serves as the core component of the proposed asynchronous multi-camera-IMU state estimation algorithm. It performs joint optimization of multi-camera visual observations, RGB-D depth information, and IMU pre-integration data within a sliding window framework to achieve high-precision state estimation. The module employs the Gauss-Newton algorithm for nonlinear least squares optimization, minimizing the sum of squared residuals to ensure robustness under asynchronous inputs and challenging scenarios, such as low-texture indoor environments or dynamic disturbances.

Incoming camera frames are arranged in timestamp order within a fixed-size sliding window typically 10 - 15 frames. Old frames are dynamically marginalized to maintain computational efficiency, while RGB-D camera depth information is fully leveraged to strengthen geometric constraints. This mechanism enables tight integration of high-frequency IMU constraints with low-frequency asynchronous multi-camera observations. During window sliding, prior information is updated via marginalization to effectively mitigate long-term drift.

Furthermore, the system supports online optimization of camera-to-IMU extrinsics. These extrinsics are included in the state vector as variables to be optimized within the sliding window, achieving real-time estimation through joint constraints from visual reprojection factors and IMU pre-integration factors, without requiring offline calibration. This online mechanism is particularly well-suited for asynchronous multi-camera scenarios, as it adaptively compensates for sensor mounting deviations or minor mechanical variations.

The proposed dynamic confidence weighting strategy further improves the convergence of extrinsic parameter estimation. By adjusting the covariance matrices of visual factors according to Equation (13), high-confidence observations receive smaller covariances, providing stronger geometric constraints, while low-

confidence observations are appropriately down-weighted to prevent erroneous constraints. In complex environments such as low-texture regions or dynamic disturbances, this dynamic weighting significantly enhances the constraining power of reliable visual observations on camera-IMU extrinsics. Experimental results show that extrinsic estimation converges faster during the initial phase and exhibits greater stability over long sequences, effectively reducing the risk of extrinsic drift caused by noisy visual observations.

3.3. Confidence Scoring Strategy

The confidence scoring strategy is a key innovation of this paper, aiming to dynamically adjust the covariance matrices of visual observation factors through unified processing of RGB images from all cameras by the pre-trained DROID-SLAM model, enhancing the robustness and accuracy of the asynchronous multi-camera-IMU state estimation system in complex scenarios. This strategy ensures that the model continuously acts on the RGB inputs of all cameras including the RGB components of RGB-D cameras, rather than relying solely on hardware depth.

The DROID-SLAM model takes RGB images as its core input and generates pixel-wise inverse depth maps along with confidence weights. For pure RGB cameras, the system fully relies on the model's output inverse depth and confidence weights. For RGB-D cameras, the model still performs independent depth estimation on the RGB images and fuses it with the hardware depth: specifically, the network's inverse depth serves as the initial value for feature point inverse depths in visual factors, while the confidence weights dynamically adjust the relative contributions of hardware depth and network depth in the reprojection error. This design achieves a soft fusion of hardware depth and network-inferred depth: hardware depth provides direct geometric constraints, whereas the model output performs robust depth inference and uncertainty modeling on RGB images, particularly in regions where hardware depth is noisy, missing, or when asynchronous timing inconsistencies occur across multiple cameras.

Conventional VINS-Multi visual factors rely on depth estimation via triangulation of sparse feature points, with their covariance matrices P_j^l typically based on static noise models, making it difficult to adapt to uncertainty variations induced by asynchronous multi-camera inputs. In low-texture regions such as wall inspection scenarios or under dynamic disturbances, the number of feature points decreases or tracking quality degrades, easily leading to trajectory drift and optimization divergence. The DROID-SLAM pretrained model, through its Dense Bundle Adjustment (DBA) layer, provides pixel-level inverse depth information and confidence weights w_{ij}, the latter generated by a CNN-GRU hybrid architecture that reflects the reliability of depth estimation and optical flow matching. The confidence weights w_{ij} serve as endogenous outputs in the covariance matrices $P_{ij} = \mathrm{diag}\left(w_{ij}\right)$ of the DBA loss function, directly influencing error weighting and enabling dynamic capture of pixel-level uncertainties, thereby

providing a data foundation for optimizing the covariance of visual factors. The core of the confidence scoring strategy lies in converting the confidence weights from DROID-SLAM into a standardized score, which is used to dynamically adjust the visual factor covariances. The specific implementation steps are as follows:

1) Confidence Score Computation

The Dense Bundle Adjustment (DBA) in DROID-SLAM is achieved through iterative optimization:

$$E\left(G,d'\right)=\sum_{(i,j)\in\varepsilon}\left\|p_{ij}-\Pi_c\left(G_{ij},\Pi_c^{-1}\left(p_{ij},d'\right)\right)\right\|_{p_{ij}}^2 \tag{11}$$

where the confidence weights are directly represented by $w_{ij}\left(u,v\right)$, with $P_{ij}=\mathrm{diag}\left(w_{ij}\right)$, where u, v denote pixel coordinates, and w_{ij} represents the geometric consistency and optical flow confidence of the corresponding pixel pairs. The GRU module generates the confidence weight matrix by processing sequential inputs, achieving gradual smoothing of the confidence distribution across multi-camera frames and enhancing system robustness.

The proposed method uses a step-by-step smoothing of the multi-camera frames' confidence distribution, enhancing system robustness. The GRU module generates the confidence weight matrix by processing sequential inputs, achieving gradual smoothing of the confidence distribution across multi-camera frames and enhancing system robustness.

2) Confidence Score Computation

To convert w_{ij} into a confidence score, the normalized confidence $C\left(u,v\right)$ is defined as:

$$C\left(u,v\right)=\frac{w_{ij}\left(u,v\right)}{\max\left(w_{ij}\right)} \tag{12}$$

where $\max\left(w_{ij}\right)$ is the maximum confidence weight across all pixels in the current frame, and $C\left(u,v\right)$ represents the normalized value of the confidence weight for the current pixel. This provides a reliability assessment, where lower $C\left(u,v\right)$ corresponds to higher uncertainty in depth estimation. Lower confidence values are downweighted in the factor graph.

3) Confidence Fusion Mechanism

In the factor graph, the covariance matrix P_j^l of the visual factor residual $e_C(z_j^l,X)$ is redefined based on confidence as:

$$P_{ij}^l=\mathrm{diag}\left(\frac{1}{C\left(u,v\right)^2+\varphi}\right) \tag{13}$$

where $\varphi=0.01$ is a small empirical regularization term, selected based on the following considerations: (1) numerical stability: when the confidence score s approaches 0, φ ensures that the covariance matrix does not tend to infinity, preventing numerical instability or singular matrices during optimization; (2) weighting balance: φ should be sufficiently small to ensure that for high-confidence observations, the covariance approaches the baseline noise, preserving their dominant

weights in optimization. The value $\varphi = 0.01$ follows common practices in similar depth uncertainty modeling works. For example, covariance regularization in DROID-SLAM and related deep learning-assisted SLAM systems and it was determined through preliminary grid search $\varphi \in \{0.001,\ 0.01,\ 0.1,\ 1.0\}$ on validation sequences, achieving the best balance between numerical stability and weighting effectiveness.

The reciprocal of $C(u,v)^2$ serves as the covariance weight, reflecting that observations with higher confidence have smaller covariances, thus occupying greater weight in the optimization. This is similar to the static covariance in VINS-Multi but offers greater adaptability.

In the front-end module of VINS-Multi, confidence scoring is introduced as a preprocessing step to adjust confidence. For low-confidence P_{fi}, in the tightly-coupled module's feature point initialization, confidence is used to adjust confidence. Confidence increases the certainty, and the post-confidence P_j^l is directly applied to the visual factors.

In this paper, the front-end module introduces confidence scoring as a preprocessing step. For low-confidence regions, the feature points in low-confidence regions are optimized, enhancing the depth confidence, providing reliable depth information for visual factors, and further improving the accuracy of 3D visual reconstruction.

The optimization incorporates the confidence-adjusted residuals as:

$$r_C = \sum_{(i,j)\in\mathcal{C}} \left\| e_C\left(z_j^l, X\right) \right\|_{p_j^l}^2 \tag{14}$$

By integrating confidence scoring, the system achieves higher accuracy in low-texture or dynamic scenes. The confidence score computation provide reliable depth information for visual factors, further improving the accuracy of 3D visual reconstruction.

3.4. Outlier Rejection Based on Chi-Square Test

By detecting and rejecting unreliable IMU observation factors through statistical methods, this enhances the reliability of IMU data and the stability of preintegration in the asynchronous multi-camera-IMU state estimation system. This mechanism specifically addresses the noise sensitivity issue of IMU preintegration factors in VINS-Multi, significantly reducing error accumulation in complex dynamic scenarios. The following provides a detailed introduction to this mechanism, including its theoretical foundation, implementation methods, integration with the system framework, and performance advantages.

The traditional VINS-Multi introduces IMU pre-integration accelerometer and gyroscope data as pose increments through pre-integration at the IMU front-end, adopting the form $e_B\left(z_k^{k+1}, X\right)$, with its dynamics matrix P_k^{k+1}. Estimation based on a rigorous model. However, in dynamic environments or under sensor faults, IMU observations may contain outliers, increasing the risk of sudden errors and leading to a decrease in system stability. Abnormal accelerations can cause posi-

tion offsets, leading to amplification of inertial integration errors, thereby affecting optimization results.

The Chi-squared Test is a widely applied statistical hypothesis testing method that performs hypothesis testing by calculating the chi-squared value of the sample and comparing it with a threshold. The chi-squared test is commonly used to detect whether it conforms to the assumed distribution and to detect whether observations conform to characteristics.

In this study, we introduce an outlier detection mechanism based on the chi-squared test. Thereby enhancing the system's robustness. The specific implementation steps of the chi-squared test mechanism are as follows:

1) Gaussian Distribution Assumption

Assume that the IMU pre-integration residual r follows a Gaussian distribution with zero mean under no abnormal conditions, and its probability density function is:

$$p(r) = \frac{1}{\sqrt{(2\pi m)\left|P_k^{k+1}\right|}} \exp\left(-\frac{1}{2} r^T P_k^{k+1^{-1}} r\right) \tag{15}$$

Among them, m is the residual dimension, usually 9, including 3 components of position, velocity, and rotation. P_k^{k+1} is the covariance matrix of IMU pre-integration. $e_B\left(z_k^{k+1}, X\right)$ is the actual residual vector. This setting provides a theoretical basis for subsequent inspections.

2) Chi-squared Statistic Calculation

The chi-squared statistic is defined based on the square of the Mahalanobis distance:

$$\chi^2 = r^T P_k^{k+1^{-1}} r \tag{16}$$

where χ^2 represents the "distance" of the residual in the standardized covariance space, and it follows a chi-squared distribution χ_m^2 with m degrees of freedom. In the absence of outliers, χ^2 should be smaller than the critical value corresponding to the preset significance level α.

3) Outlier Detection and Rejection

Set the significance level $\alpha = 0.05$ (corresponding to 95% confidence level). For degrees of freedom m = 9, the corresponding chi-squared critical value is approximately 16.919 (obtained from chi-squared distribution table). The judgment condition is:

$$\chi^2 < \chi_{\alpha,m}^2 \tag{17}$$

If $\chi^2 > \chi_{\alpha,m}^2$, the current IMU observation is regarded as an outlier (abnormal), and the corresponding pre-integration factor e_B is rejected to prevent it from entering the subsequent optimization process. After rejection, the system performs interpolation using adjacent valid IMU measurements or re-performs pre-integration to maintain measurement continuity.

In this paper, a chi-squared test module is newly added to compute in real time the residual e_B and the χ^2 value for each time segment. The outlier detection

result is directly fed back to the front-end coordinator, which marks unreliable IMU data segments. During frame-priority coordination, the front-end coordinator adjusts the time-interval priority based on the anomaly status of the IMU data. When an IMU anomaly is detected, the weight δt_i is reduced (*i.e.*, the priority of the corresponding time interval is lowered), so that visual frames are preferentially selected to dominate the optimization, thereby reducing dependence on unreliable IMU measurements.

In the sliding-window optimization, the rejected IMU factors no longer participate in the optimization; only the retained reliable terms are kept, ensuring reliable input to the optimization objective. At the same time, the state vector X after removal of the anomalous factors is updated via marginalization to generate the prior factor, thereby maintaining overall system consistency.

The chi-squared test mechanism effectively addresses the IMU noise sensitivity issue in VINS-Multi, particularly in dynamic scenarios, such as sudden accelerations during ventilation duct inspections or under sensor fault conditions. Experimental results show that the IMU outlier detection rate improved by approximately 20%. In the raw_515_435 sequence, the RMSE was significantly reduced and drift was markedly decreased. This benefit stems from outlier rejection, which reduces the accumulation of pre-integration errors while still maintaining the high-frequency output of 500 Hz, thereby validating the mechanism's effective balance between real-time performance and robustness.

4. Experimental Results Evaluation and Analysis

All experiments in this paper were conducted on a unified experimental platform equipped with an Intel Core i5-1240P processor (1.7 GHz base frequency), 16 GB RAM, and running the Ubuntu 20.04 operating system.

To verify the localization performance of the proposed algorithm in complex environments, this paper integrates the depth confidence scoring strategy and the chi-squared test mechanism into the asynchronous multi-camera-IMU state estimation algorithm and incorporates it into the VINS-Multi framework. Experiments were performed on 2 simulation dataset sequences as well as 2 sequences from the publicly available VINS-Multi raw_515_435 dataset. To ensure the reliability of the experimental results, each sequence was run 8 times, and the average values were taken to reduce randomness. In addition, ablation studies were conducted targeting the specific algorithmic details proposed in this paper. The EVO evaluation tool [15] was employed, and after alignment using the Umeyama method [16], the root mean square error (RMSE) and mean value of the absolute trajectory error (Absolute Trajectory Error, ATE) were adopted as the primary evaluation metrics.

Experimental Results Analysis

The publicly available VINS-Multi dataset, raw_515_435, covers a variety of complex scene characteristics, including low-texture indoor environments and regions

with dynamic interference, making it suitable for evaluating the robustness of the proposed algorithm. In addition, this paper's experiments include 2 simulation scenario sequences: simulation low-texture indoor experiment 01 and dynamic interference simulation experiment 02, both providing asynchronous input data from multiple cameras and IMU, fully supporting the validation requirements of the proposed algorithm. In the experimental results, the abbreviations 01_SIM and 02_SIM correspond to the simulation sequences in the dataset, while 03_RAW and 04_RAW denote the real-world data sequences.

To verify the effectiveness of the depth confidence scoring strategy and the chi-squared test mechanism, Table 1 presents a comparative analysis of the following three algorithm configurations: the complete proposed algorithm in this paper, the proposed algorithm without the depth confidence scoring strategy, and the original VINS-Multi algorithm. The overall RMSE and Mean values are reported for each algorithm.

The experiments were conducted for comparison on two simulation sequences (01_SIM, 02_SIM) and two real-world dataset sequences (03_RAW, 04_RAW), with the best-performing values bolded. On average, the localization accuracy improved by 31.6%, and in the 01_SIM and 04_RAW sequences—which feature low-texture regions or scenes with large motion magnitude—the improvement reached 46.8%.

The overall results demonstrate that the proposed algorithm outperforms VINS-Multi in the vast majority of scenarios. This superior performance is mainly attributed to two key factors:

1) the depth confidence scoring strategy effectively eliminates uncertainty noise in visual factors and dynamically adjusts the covariance matrix, thereby significantly enhancing robustness in low-texture regions;

2) the chi-squared test mechanism successfully rejects abnormal IMU observations, further improving the stability of pre-integration and making a substantial contribution to localization accuracy.

Table 1. Comparison of root mean square error (RMSE) across all sequences (unit: m).

Sequences	Proposed algorithm		Proposed algorithm without the depth confidence scoring strategy		VINS-Multi	
	RMSE	Mean	RMSE	Mean	RMSE	Mean
01_SIM	**0.550**	**0.388**	0.669	0.616	0.892	0.820
02_SIM	**0.448**	**0.299**	0.593	0.471	0.613	0.520
03_RAW	**1.998**	**1.437**	3.049	2.966	3.891	3.759
04_RAW	**3.820**	**3.252**	4.642	4.273	4.562	4.076
average	**1.704**	**1.344**	2.238	2.081	2.489	2.293

To analyze the sources of error, this section discusses the issue from the per-

spectives of residual distribution and outliers. Table 2 presents the outlier statistics for the VINS-Multi algorithm and the proposed algorithm on the 01_SIM and 02_SIM sequences. Figure 3 and Figure 4 illustrate the outlier distribution for these two sequences, respectively. The results show that the outlier proportion in the VINS-Multi algorithm is significantly higher than that in the proposed algorithm; specifically, in the 01_SIM sequence, the outlier ratio of VINS-Multi reaches 85%. These outliers primarily originate from noise accumulation in the aforementioned IMU preintegration process, particularly acceleration jumps in dynamic scenarios. Consequently, the number of actually effective IMU observations in VINS-Multi is relatively small, whereas the proposed algorithm effectively mitigates the impact of outliers through the chi-square test mechanism.

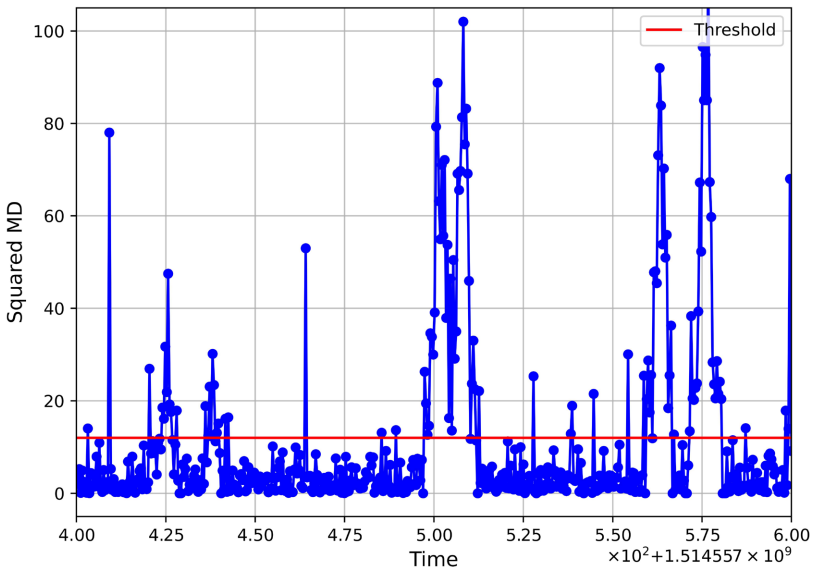

Figure 3. Outlier rejection in 01_SIM sequence using VINS-multi algorithm.

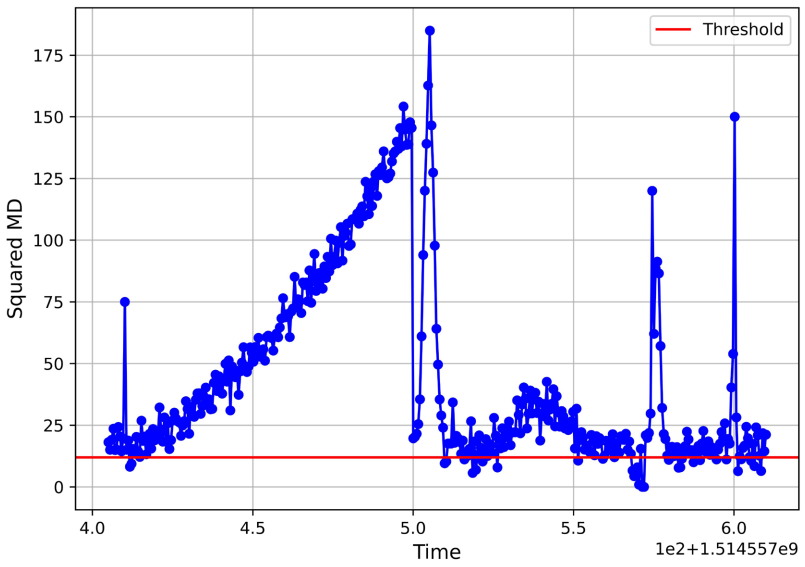

Figure 4. Outlier rejection in 01_SIM sequence using the proposed algorithm.

Table 2. Outlier rejection in 01_SIM and 02_SIM sequences using chi-square test.

Algorithm	01_SIM		02_SIM	
	Number of Outliers	Outlier Proportion (%)	Number of Outliers	Outlier Proportion (%)
VINS-Multi	987	85.1	1105	77.5
Proposed	**164**	**14.2**	**136**	**9.6**

To ensure the real-time performance of the algorithm in practical applications, Table 3 reports the processing time of each module and the overall frame rate. The statistics are obtained by running the algorithm 8 times on the real-world dataset sequences 03_RAW and 04_RAW and taking the average values. A percentage comparison is also made with the original optimization module of VINS-Multi.

The depth confidence scoring strategy computes pixel-level confidence weights using the pretrained DROID-SLAM model and dynamically adjusts the covariance of visual factors, thereby avoiding the need for complex noise modeling. Meanwhile, the chi-square test mechanism statistically rejects anomalous IMU measurements, reducing unnecessary preintegration optimization overhead. Compared to VINS-Multi's static covariance estimation and lack of outlier filtering, these designs significantly alleviate the computational burden.

Table 3. Statistics of algorithm module execution time (unit: ms).

Module	Proposed	VINS-Multi	Increase (%)
Depth confidence scoring	1.5	-	-
IMU preintegration module	1.8	1.7	5.8
Chi-square test	4.2	-	-
Factor graph optimization	8.5	7.8	8.9
Overall per-frame	16.8	15.5	8.3

The results show that although the proposed algorithm introduces additional confidence scoring and chi-square test modules, the overall per-frame processing time increases by only 8% thanks to efficient residual screening and covariance adjustment. The system still maintains a real-time capability of approximately 30 fps, demonstrating its practicality in asynchronous multi-camera-IMU state estimation.

Figure 5 presents a comparison of the 3D trajectories estimated by VINS-Multi and the proposed algorithm on the 03_RAW sequence. This is a long sequence that includes low-texture indoor environments and dynamic interference, with a total mileage approaching the complex path length of the indoor test scenario. Consequently, the collaborative observations from multiple cameras and IMU play a particularly important role, providing local constraints and dynamic com-

pensation. However, due to texture deficiency and asynchronous inputs, this scenario is prone to introducing noise.

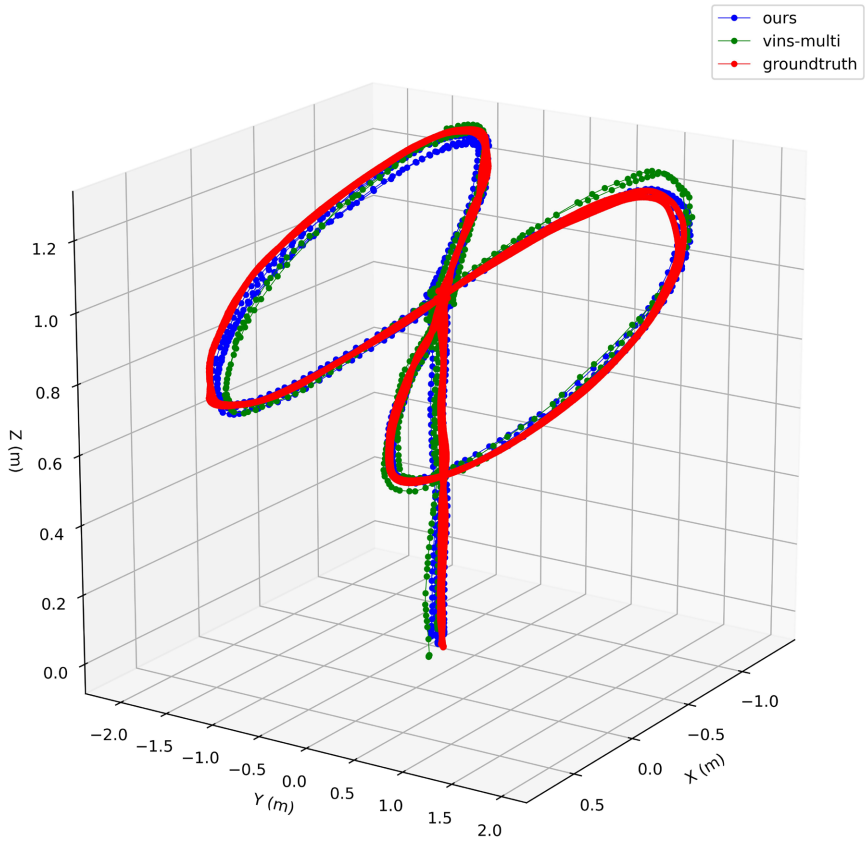

Figure 5. Trajectory comparison in 03_RAW sequence.

Although differences are not always immediately obvious in long sequences at first glance, a closer inspection reveals that the trajectory produced by the proposed algorithm stays closer to the ground-truth values provided by the dataset. This demonstrates that the method effectively mitigates, to a certain extent, the impact of visual and inertial measurement noise in low-texture regions.

5. Conclusions

In this paper, we successfully improve the accuracy and robustness of asynchronous multi-camera-IMU state estimation through the integration of a depth confidence optimization strategy and a chi-square test mechanism. Specifically, the depth confidence scoring strategy leverages pixel-level confidence weights from the pretrained DROID-SLAM model to dynamically adjust the covariance matrices of visual factors, significantly reducing trajectory drift errors in low-texture or dynamically disturbed scenes. Experimental results demonstrate an average RMSE reduction of 31.6%. Meanwhile, the chi-square test mechanism eliminates unreliable IMU observation factors via statistical outlier detection, further enhancing the stability of preintegration. In challenging environments such as ventilation

duct inspection or sensor fault recovery, this approach substantially increases the IMU anomaly detection rate, thereby mitigating accumulated errors and ensuring overall system reliability.

Although the current work achieves strong performance on simulation and public datasets, it still exhibits potential limitations related to dependency on training data, particularly in terms of generalization under extremely asynchronous inputs. Future work may explore multi-modal confidence fusion, for example by combining depth confidence weights with IMU noise models to achieve more comprehensive sensor synergy. Additionally, adaptive threshold adjustment for the significance level of the chi-square test could better accommodate time-varying noise distributions, thereby further improving the algorithm's adaptability and computational efficiency in a broader range of complex environments. These extensions are expected to advance the practical deployment of asynchronous multi-sensor SLAM in robotic navigation, indoor localization, and related fields, delivering more reliable solutions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wang, L., Xu, Y. and Shen, S. (2024) VINS-Multi: A Robust Asynchronous Multi-camera-IMU State Estimator.

[2] Kühne, J., Vogt, C., Magno, M. and Benini, L. (2025) Efficient and Accurate Down-facing Visual-Inertial Odometry. *IEEE Internet of Things Journal*, **12**, 48376-48387. https://doi.org/10.1109/jiot.2025.3609011

[3] Zhang, D., Zhang, J., Sun, Y., Li, T., Yin, H., Xie, H., *et al.* (2025) Towards Robust Sensor-Fusion Ground SLAM: A Comprehensive Benchmark and a Resilient Framework. 2025 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Hangzhou, 19-25 October 2025, 8894-8901. https://doi.org/10.1109/iros60139.2025.11247507

[4] Cao, Y., He, X., Chen, Y., *et al.* (2025) Omni-LIVO: Robust RGB-Colored Multi-Camera Visual-Inertial-LiDAR Odometry via Photometric Migration and ESIKF Fusion.

[5] Ma, C., Cheng, P. and Cai, C. (2024) Localization and Mapping Method Based on Multimodal Information Fusion and Deep Learning for Dynamic Object Removal. *International Journal of Network Dynamics and Intelligence*, **3**, Article ID: 100008. https://doi.org/10.53941/ijndi.2024.100008

[6] Jung, J., Boche, S., Laina, S.B. and Leutenegger, S. (2025) Uncertainty-Aware Visual-Inertial SLAM with Volumetric Occupancy Mapping. 2025 *IEEE International Conference on Robotics and Automation* (*ICRA*), Atlanta, 19-23 May 2025, 14550-14556. https://doi.org/10.1109/icra55743.2025.11127824

[7] Shapira, T.O. and Klein, I. (2025) ICD-Net: Inertial Covariance Displacement Network for Drone Visual-Inertial SLAM.

[8] Teed, Z. and Deng, J. (2021) DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Proceedings of the* 35*th International Conference on Neural Information Processing Systems*, 6-14 December 2021, 16558-16569.

[9] Dellaert, F. and Kaess, M. (2017) Factor Graphs for Robot Perception. *Foundations and Trends in Robotics*, **6**, 1-139. https://doi.org/10.1561/2300000043

[10] Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., *et al.* (2020) TartanAir: A Dataset to Push the Limits of Visual SLAM. 2020 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Las Vegas, 24 October 2020-24 January 2021, 4909-4916. https://doi.org/10.1109/iros45743.2020.9341801

[11] Chung, J., Gulcehre, C., Cho, K.H., *et al.* (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

[12] Qin, T., Li, P. and Shen, S. (2018) VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, **34**, 1004-1020. https://doi.org/10.1109/tro.2018.2853729

[13] Qin, T., Cao, S., Pan, J., *et al.* (2019) A General Optimization-Based Framework for Global Pose Estimation with Multiple Sensors.

[14] Derpanis, K.G. (2010) Overview of the RANSAC Algorithm. *Image Rochester NY*, **4**, 2-3.

[15] Grupp, M. (2017) EVO: Python Package for the Evaluation of Odometry and SLAM. https://github.com/MichaelGrupp/evo

[16] Umeyama, S. (1991) Least-Squares Estimation of Transformation Parameters between Two Point Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 376-380. https://doi.org/10.1109/34.88573