# Enhancing VINS with Smart Feature Grading: Overcoming Cautious and Excessive Removal of Dynamic Features for Robust Urban Localization

Mahmoud Adham [1,2], Wu Chen [1], Ahmed Mansour[1,2] , Mostafa Mahmoud [1,2], Yaxin Li [1*]

[1] The Hong Kong Polytechnic University (PolyU), The Department of Land Surveying and Geo-Informatics (LSGI), Hong Kong S.A.R. China (mahmoud.arif, ahmed.m.mostafa, mostafa.mahmoud)@connect.polyu.hk, (wu.chen, yaxin.pu.li)@ polyu.edu.hk
[2] Cairo University, the Public Works Department, Faculty of Engineering, Giza, Egypt

**Keywords:** Visual-inertial systems (VINS), dynamic features, visual feature grading, urban localization, autonomous vehicles

**Abstract**

Visual-inertial navigation systems (VINS) have emerged as a popular and effective solution for autonomous navigation due to their accuracy, real-time capabilities, and cost-effectiveness. However, while traditional VINS methods excel in static environments with well-distributed features, they struggle in highly dynamic urban environments where moving objects distort feature tracking, leading to pose estimation errors and localization inaccuracies. Recent approaches, such as image geometric constraints-based methods, aim to address these challenges but are limited when moving objects dominate the scene. Deep learning (DL)-based methods, which directly remove potential dynamic objects, often degrade accuracy in low-texture scenes and overlook the resulting uneven feature distribution, further impacting state estimation. To address these issues, we propose a novel VINS method that combines visual and inertial information with a smart feature grading module to overcome cautious and excessive dynamic feature removal, effectively handling the complexities of dominant and ambiguous dynamic objects beyond the limitations of traditional DL and vision-based methods. The method's performance shows effective identification and filtering of dynamic features while preserving static ones. Tests carried out on multiple datasets in urban dynamic environments highlight the method's enhanced accuracy and robustness.

## 1. Introduction

Localization and navigation in a GPS-denied environment always pose great challenges (Rabbou et al., 2021; Li et al., 2023; Wang et al., 2024). Over the past decades, visual-inertial navigation systems (VINS) have seen significant advances, making them an attractive solution for robust positioning in various environments (Reid et al., 2019). Prominent methods like ORB-SLAM3 (Campos et al., 2021), Open-VINS (Geneva et al., 2020), and VINS-mono (Qin et al., 2018) have demonstrated high performance in feature-rich, static environments. While VINS has showcased significant achievements, it mainly relies on the static nature of the surrounding features. However, the performance of VINS can be significantly hindered in highly dynamic scenarios such as large-scale urban environments where the quality of feature tracking is affected by moving objects (Adham et al., 2024). Visual features associated with these moving objects become distorted or dislocated, leading to degraded feature tracking accuracy, pose deviations, trajectory drift, and reduced system robustness in such environments. Traditional VINS methods filter out the moving features as outliers. Typical methods such as VINS-Mono and ORB-SLAM3 adapt RANSAC to eliminate such features but they're less effective when the environment has many moving objects. Therefore, achieving robust and drift-free positioning in large-scale outdoor dynamic environments is still challenging (Mahmoud et al., 2022a; Mansour and Chen, 2022). Researchers have developed various algorithms to tackle this issue, focusing on direct detection and removal of these objects using either visual data alone or combined visual and IMU data (He and Rajkumar, 2021).

Recent studies reveal a clear gap in current approaches, highlighting some limitations in effectively removing moving objects. Geometric constraint-based approaches struggle in highly dynamic areas like urban environments, where moving objects introduce noise in feature correspondences, degrading the accuracy of the fundamental matrix affecting camera motion estimation. Epipolar geometry methods fail when dynamic features move along the epipolar line, and optical flow-based methods stumble when objects move toward/away from the camera or have large displacements between frames. IMU-integrated VINS algorithms improve robustness but struggle with prolonged stops or pure rotations. DL-based methods, limited by predefined knowledge, often fail in low-light conditions, incomplete object capture, poor detection of distant moving objects or with unpredictable object motion. Moreover, DL methods tend to indiscriminately between stationary and dynamic objects, leading to excessive feature removal and geometric distortions, which affect pose estimation accuracy.

To address these challenges, we propose a novel VINS method that integrates visual and inertial sensor data with a smart feature grading module. Our approach overcomes the limitations of traditional DL and vision-based methods by introducing a real-time perceptual feature grading and processing module. This module categorizes tracked image features into stable, fixed, and fickle categories, enabling reliable transformation calculations from static points. A Hybrid Geometric Correspondence Constraints (HGCC) module is introduced to maintain geometric consistency across static points between frames, applying multiple constraints to handle fickle features effectively.

Furthermore, to ensure robustness and eliminate missed dynamic objects, we propose a VI-based motion consistency constraint. This addresses cases such as unknown moving objects, distant moving objects, and partially captured dynamic objects. Finally, to mitigate uneven feature distribution caused by excessive removal of dynamic points, we introduce an auto-adaptive covariance estimation method. This dynamically estimates a weight factor based on feature distribution in each frame,

* corresponding author

remodelling the covariance matrix of visual measurements during VIO pose estimation.

## 2. Literature Review

Recent research has focused on improving VINS robustness in dynamic environments by addressing the challenges posed by moving objects. Approaches can be categorized into geometric constraint-based methods, deep learning (DL)-based methods, and hybrid techniques integrating visual and IMU data.

Geometric constraint methods leverage camera pose estimation and geometric relationships to detect and remove dynamic features. For instance, (Tan et al., 2013) used Adaptive RANSAC to remove invalid feature points, while (Sun et al., 2017) employed optical flow (Ali et al., 2021) and homography computation to eliminate moving objects. Epipolar geometry-based methods, such as those by (Fan et al., 2019) and (Cheng et al., 2019), face challenges with slow-moving or large dynamic objects and errors in positional transformations. IMU-aided methods, such as those by (Fu et al., 2021) and (Reginald et al., 2022), combine IMU data with epipolar constraints to filter dynamic features. However, these methods struggle when dynamic features move along the epipolar line or when IMU data is biased. DynaVINS (Song et al., 2022) introduced a loss function integrating IMU pre-integration into Bundle Adjustment, but its practicality is limited by hyperparameter tuning (Mohammed et al., 2023).

DL-based methods, such as those by (Zhang et al., 2018) and (Wu et al., 2022), use object detection models like YOLO to filter dynamic features. However, these methods often fail to distinguish between stationary and moving objects, leading to excessive feature removal. Dynamic-VINS (Liu et al., 2022) combine object detection (Mahmoud et al., 2022b) with motion models or depth information but face challenges in large-scale environments. Hybrid approaches, such as by (Zhang et al., 2021) integrate DL segmentation (Mahmoud et al., 2024a) with geometric constraints but struggle with real-time execution and excessive feature removal. These mentioned DL-methods overlook missed detections or object detection failures, which can degrade feature tracking and cause pose deviations at detected locations.

Our approach introduces an innovative semantic-aware and multi-level geometric constraint framework designed to address the challenges posed by dominant and ambiguous dynamic objects. By surpassing the limitations of traditional DL and vision-based methods, our method enhances localization accuracy and state estimation robustness, making it particularly effective for dynamic outdoor environments.

## 3. Methodology

In this section, the details of the proposed system are introduced as follows.

### 3.1 System Overview

The proposed system, illustrated in Figure 1, integrates a novel front-end and adaptive back end combining camera and IMU measurements. The framework begins with IMU pre-integration between consecutive frames, followed by parallel processing of visual and semantic information to efficiently handle moving objects. A feature grading module categorizes tracked features into fixed, stable, and fickle categories, enabling dynamic feature filtering. A developed motion consistency procedure is proposed to ensure robust tracking by combining IMU pre-integration states and optimized pose estimation. Subsequently, outlier culling is applied to eliminate unsatisfactory tracked feature observations and associated landmarks based on two reprojection error processes. The proposed auto-adaptive covariance estimation method is introduced to address uneven feature distribution caused by excessive removal of dynamic features. The back-end employs pose graph optimization in a local visual-inertial odometry estimator ensuring accurate trajectory estimation in dynamic environments.

### 3.2 Scene Understanding and Tracking

This module combines semantic object detection with geometric feature tracking to enable robust scene understanding in dynamic environments. A custom-trained YOLOv5 model (Redmon and Farhadi, 2018) is employed for real-time object detection, classifying common outdoor objects into two categories: fixed (e.g., traffic lights, benches) and potentially dynamic (e.g., cars, pedestrians). Objects belonging to the second category demand a more thorough analysis by the system to ascertain their likelihood of being in motion. The model, accelerated using TensorRT, provides semantic labels and bounding boxes for detected objects, allowing the system to prioritize static features while flagging dynamic ones for further analysis. Concurrently, geometric feature tracking is performed using the Shi-Tomasi corner detector (Shi and Tomasi, 1994) and the pyramidal KLT optical flow algorithm (Lucas and Kanade, 1981). The Shi-Tomasi detector ensures a uniform distribution of feature points, while the pyramidal KLT algorithm handles challenges such as fast camera motion, large displacements, and illumination changes. Keyframes are selected based on parallax and overlap criteria to optimize pose estimation efficiency, ensuring reliable landmark triangulation and reprojection factors for graph optimization. This dual-threaded approach enables the system to maintain high accuracy and computational efficiency in real-time applications.

### 3.3 Smart Feature Grading Module

The pyramidal KLT tracking method, while effective for tracking features in most scenarios, faces challenges in dynamic environments due to abnormal motion patterns caused by moving objects. Additionally, relying solely on deep learning models to determine the motion status of objects such as distinguishing between moving and stationary vehicles proves insufficient. To address these limitations, the system leverages semantic information from detected objects to classify features into three distinct categories: stable, fixed, and fickle. This classification is based on the object category associated with each feature, as outlined in Section 3.2.

The feature grading process begins by clustering visual feature points based on their location relative to detected bounding boxes. Features outside the bounding boxes are classified as stable, as they are unlikely to belong to moving objects. These features are refined using the RANSAC algorithm to exclude outliers, ensuring only reliable points are used for transformation calculations. Features inside bounding boxes are further categorized as fixed or fickle. Fixed features correspond to absolutely static objects, such as traffic lights, stop sign, poles, or benches, and are used directly for transformation calculations if they meet specific conditions, such as being tracked over a minimum number of consecutive frames and having a depth below a predefined threshold. Thus, the calculations exclusively rely on features near the vehicle's body, thereby maximizing the reliability of the outcomes.

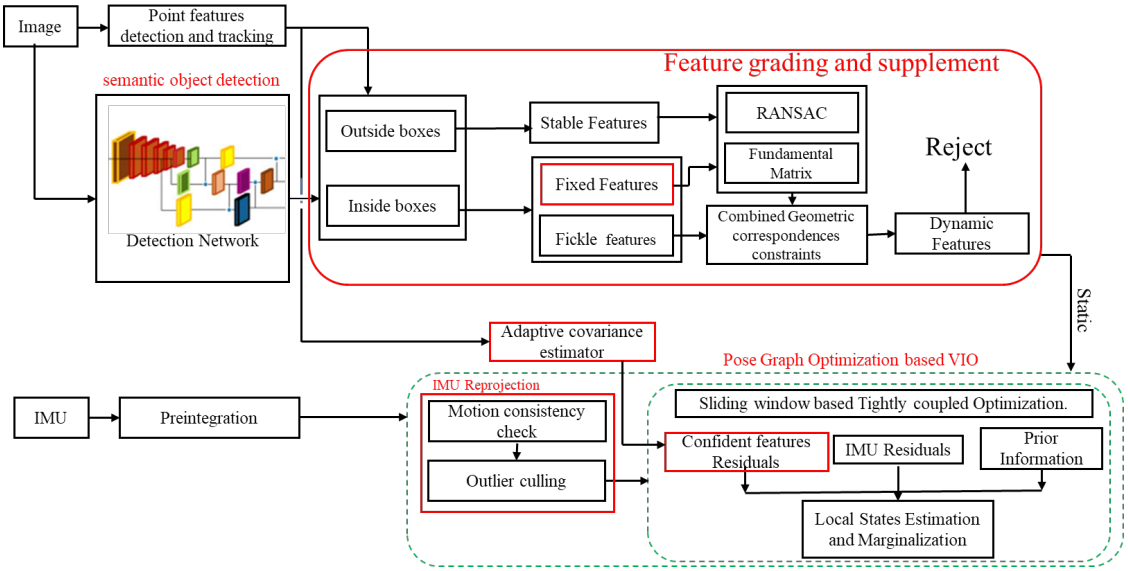


Figure 1. Overview of our proposed system; boxes highlighted in red are the main contributions of the proposed approach.

Fickle features, associated with potentially dynamic objects like vehicles or pedestrians, require further analysis to determine their motion status. To handle these features, a Hybrid Geometric Correspondences Constraints (HGCC) module is introduced. The HGCC module combines multiple geometric constraints to reduce the limitations of using individual constraints. It first calculates the fundamental matrix (F) using stable and fixed features which capture the geometric relationships among static feature points in consecutive frames and then applies two geometric constraints to mitigate interference from dynamic objects. This module uses epipolar geometry (Hartley and Zisserman, 2003) and novel sliding window-optical flow constraints to distinguish between static and dynamic features.

The first constraint is based on the principles of epipolar geometry, which states that a matched point in a subsequent frame should lie on its corresponding epipolar line as depicted in Figure 2. Using the accurately calculated Fundamental matrix (F), the system computes the epipolar lines for feature points in the current frame. For static point P as shown in Figure 2, its corresponding feature points $p_1$ and $p_2$ in consecutive frames satisfy the epipolar constraint:

$$\boldsymbol{p}_{j,i+1}^{T}\boldsymbol{F}\boldsymbol{p}_{j,i} = \boldsymbol{p}_{j,i+1}^{T}\boldsymbol{l}' = 0,\ j = 1,2,3,\dots,n. \quad (1)$$

However, the presence of dynamic objects introduces abnormality, as illustrated by the case of point P' in figure 2 (a). To address this interference, we calculate the distance (D) from a feature point to its corresponding epipolar line $l = Ax + By + c = 0$, using:

$$D = \frac{|Ax_j + By_j + c|}{\sqrt{A^2+B^2}} = \frac{|\mathbf{p}_{j,i+1}^{T}\mathbf{F}\mathbf{p}_{j,i}|}{\sqrt{A^2+B^2}} \quad (2)$$

The epipolar constraint distance of a feature point to its corresponding epipolar line with static features exhibiting small distances while dynamic features showing larger deviations. By evaluating the D distance with a predefined threshold, we can effectively distinguish between static and dynamic points. However, this method faces challenges when dynamic features move along the optical center, leading to inaccurate determinations. To address this, the system proposes a sliding window-based optical flow constraint, which analyzes the displacement of features across multiple frames. In traditional VINS, feature point matching typically involves two adjacent keyframes to calculate the relative camera motion. However, our approach proposes a constraint that considers the tracked features in the current frame and all other frames within a sliding window. This sliding window encompasses all the frames on which the feature is located. The displacement of the correspondence point in the current frame is compared with all frames where the tracked point is located after applying the reliable F to wrap the sequence frames. This analysis allows us to distinguish between static points, which exhibit minimal or zero displacement, and dynamic features, which display varying degrees of displacement. To quantify the average displacement $\gamma_m$ of the observed feature $m^{\text{th}}$ that is initially observed in the frame ($i$), we compute the average of its displacements within a sliding window spanning $N$ frames where the feature is observed. This calculation is performed according to the equation shown as follows:

$$\gamma_m = \frac{1}{N}\sum_{i\neq j}\left\| p_m^{c_i} - \left(F_{c_j}^{c_i} p_m^{c_j}\right)\right\|, \quad (3)$$

where $p_m^{c_i}$ the observation of the $m^{\text{th}}$ feature in the $i^{\text{th}}$ frame. $p_m^{c_j}$ is the $m^{\text{th}}$ correspondence feature coordinates in the j$^{\text{th}}$ frame. $F_{c_j}^{c_i}$ is the transformation between the sliding window frames. When the $\gamma_m$ is over a preset threshold, the $m^{\text{th}}$ feature is considered as a dynamic feature. By integrating both the epipolar and sliding window constraints, the HGCC module overcomes the limitations of relying on a single geometric constraint. This dual approach enhances the system's robustness, enabling it to accurately distinguish between static and dynamic features in complex environments.

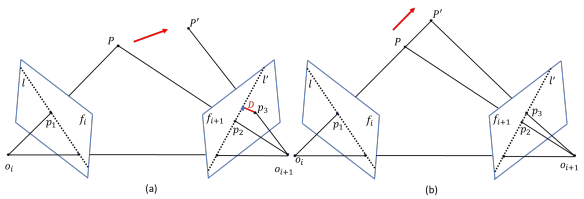


Figure 2. Epipolar constraint for Fickle features. (a) general feature point motion. (b) motion along the optical center. $p_1$ and $p_2$ are correspondence points in frames, while $p_3$ is the projection in the second frame based on the moving point (P).

### 3.4 Visual-Inertial Odometry (VIO) Based on Factor Graph Optimization

The proposed system employs a Visual-Inertial Odometry (VIO) algorithm to tightly integrate visual and IMU measurements for local pose estimation. Building upon the VINS-Mono framework (Qin et al., 2018), we introduce an adaptive backend alongside a novel frontend to enhance performance in complex environments. The backend utilizes a sliding window approach to fuse measurements within a factor graph optimization framework. The full state vector $\chi$ within the sliding window is defined as:

$$\chi = [\boldsymbol{x}_0, \boldsymbol{x}_1, \dots\dots, \boldsymbol{x}_{n1}, \boldsymbol{p}_c^b, \boldsymbol{q}_c^b, s_0, s_1 \dots, x_{n2}]^T \quad (4)$$

$$\boldsymbol{x}_m = \left[\boldsymbol{p}_{b_m}^w, \boldsymbol{v}_{b_m}^w, \boldsymbol{q}_{b_m}^w, \boldsymbol{b}_a^{b_m}, \boldsymbol{b}_g^{b_m}\right]^T \; m \in [0, n_1] \quad (5)$$

where $\boldsymbol{x}_m$ represents the IMU state at the mth frame, including position, velocity, orientation, and IMU biases. The state vector also includes the inverse depth of visual features and extrinsic parameters between the camera and IMU. Subsequently, the local VIO optimization problem is formulated as a maximum a posteriori (MAP) estimation, minimizing the residuals of prior information, IMU measurements, and visual measurements:

$$\min_{\chi}\left\{\left\|\boldsymbol{r}_p - \boldsymbol{J}_p\boldsymbol{\chi}\right\|^2 + \sum_{i\epsilon B}^{n}\left\|\boldsymbol{r}_B\left(\hat{\boldsymbol{z}}_{b_{i+1}}^{b_i}, \boldsymbol{x}\right)\right\|_{\boldsymbol{P}_{b_{i+1}}^{b_i}}^2 + \sum_{(k,j)\epsilon c}^{n}\left\|\boldsymbol{r}_c\left(\hat{\boldsymbol{z}}_k^{c_j}, \boldsymbol{x}\right)\right\|_{\boldsymbol{P}_k^{c_j}}^2\right\} \quad (6)$$

where $\boldsymbol{r}_p$ and $\boldsymbol{J}_p$ represent the prior residual and Jacobian from marginalization, $\boldsymbol{r}_B$ is the IMU residual, and $\boldsymbol{r}_c$ is the visual reprojection residual. The Ceres solver (Agarwal, Sameer, Mierle, 2023)is used to solve this optimization problem. The visual reprojection residual is computed as the difference between the observed feature location and its projected location. The residuals related to the IMU is computed based on the IMU preintegration measurements. The IMU pre-integration (Forster et al., 2017) approach merges several IMU measurements into a single integrated measurement, to mitigate the high computation load. To minimize computing complexity, IMU states and features are marginalized from the sliding window, converting the correlated measurements into prior information.

In dynamic environments, our approach faces two key difficulties in solving this optimization problem. First, the object recognition model's vulnerability to missed detections impacts state optimization due to features originating from unidentified moving objects. Second, excessive removing dynamic features, especially in scenes with a high prevalence of movement, alters the surrounding feature distribution and reduces state estimation accuracy. To address these issues, we have modified the local back end, as detailed in the following subsections.

**3.4.1 Dynamic Feature Culling via IMU-Visual Reprojection Constraint:** We incorporate an extra constraint to remove remaining dynamic features, enhancing system reliability. In outdoor environments, certain challenging scenarios arise, such as undetected unknown moving objects, rapidly moving objects at a distance, or instances where only a portion of a dynamic object appears in the image. These unresolved factors persistently influence the system's state optimization. This module utilizes visual-inertial fusion, as accurately projecting a feature point onto the current frame is difficult without knowing the camera's pose.

This constraint leverages IMU preintegration to estimate the camera pose and compute reprojection errors across all frames observing a feature not only between two consecutive frames overcoming issues arise from scenarios involving fast camera, degenerate cases (e.g., planar scenes or pure rotational motion). The camera poses for the j-th frames are obtained through IMU preintegration then refined and utilized to determine the 3D coordinates $P_m^{c_j}$= [x, y, z] of feature landmarks visible across multiple frames. Then two reprojection residuals $r_{c1}$ *and* $r_{c2}$ for the feature measurements in the observed frames are then computed as follows.

$$\boldsymbol{p}_m^{c_j} = \pi\left(\boldsymbol{R}_b^c\boldsymbol{R}_w^{b_j}\boldsymbol{R}_{b_i}^w\boldsymbol{R}_c^b\boldsymbol{P}_m^{c_j}\right) \quad (7)$$

$$r_{c1} = \frac{\boldsymbol{p}_m^{c_j}}{p_m^{c_j}.z} - \boldsymbol{p}_m^{c_i}, r_{c2} = \frac{\boldsymbol{p}_m^{c_j} - \boldsymbol{p}_m^{c_i}}{depth} \quad (8)$$

$$r_m = \frac{1}{N}\sum_{i\neq j}\|r_c\| \quad (9)$$

where $\boldsymbol{p}_m^{c_j}$ is the observed feature location, and $r_m$ is the average reprojection residual. Features failing to meet predefined thresholds are culled as outliers. This process helps ensure that only reliable features and observations are retained in the system.

**3.4.2 Geometry-Guided visual Covariance Estimator:** Uniform feature distribution around the camera is optimal for accurate state estimation, but deviations from this optimal arrangement can degrade performance. To address uneven feature distribution caused by excessive dynamic feature removal, an adaptive covariance estimation method is introduced. This module dynamically estimates a weight factor based on feature distribution in each frame, remodelling the covariance matrix of the visual measurement during VIO Pose estimation. The uncertainty in feature geometry is quantified by the distance Di between the camera pose (cx,cy) calculated from the calibration process (Mahmoud et al., 2020) and each feature position (pxi,pyi) and the weight distribution coefficient M is computed based on the spread of features relative to the camera pose:

$$M = \frac{\frac{1}{n}\sum_i D_i}{\sqrt{\frac{\sum_i\left(D_i - \frac{1}{n}\sum_i D_i\right)^2}{n}}} \quad (10)$$

The adaptive information matrix calculation is derived as follows to enhance the MAP problem:

$$\sum_k^{cj\;\;-1} = \boldsymbol{P}_k^{c_j\;-1}.\|\mathrm{M}\| \quad (11)$$

$$\min_{\chi}\left\{\left\|\boldsymbol{r}_p - \boldsymbol{J}_p\boldsymbol{\chi}\right\|^2 + \sum_{i\epsilon B}^{n}\left\|\boldsymbol{r}_B\left(\hat{\boldsymbol{z}}_{b_{i+1}}^{b_i}, \boldsymbol{x}\right)\right\|_{\boldsymbol{P}_{b_{i+1}}^{b_i}}^2 + \sum_{(k,j)\epsilon c}^{n}\left\|\boldsymbol{r}_c\left(\hat{\boldsymbol{z}}_k^{c_j}, \boldsymbol{x}\right)\right\|_{\sum_k^{cj}}^2\right\} \quad (12)$$

where $\boldsymbol{P}_k^{c_j}$ is the original covariance matrix of the residual term when the kth feature is observed by jth camera. The covariance matrix, which is inherently tied to the focal length as described in (Qin et al., 2018), remains fixed. Its inverse, $\boldsymbol{P}_k^{c_j\;-1}$, functions as the original information matrix. To address variations in feature distribution, this matrix is dynamically adjusted using the coefficient M, as defined in Equation (11), yielding the adaptive covariance matrix $\sum_k^{cj\;\;-1}$. This adaptive matrix is then applied to modify the visual error term in Equation (6), as shown in

Equation (12). Figure 3 illustrates the spatial arrangement of features following the removal of dynamic points associated with moving objects across various frames. The results highlight instances of degraded geometric distribution in certain frames, primarily caused by the extensive elimination of features.

Figure 3. Geometry distribution of the features in different frames after eliminating the dynamic features.

## 4. Experiments, Results, and Discussion

### 4.1 Experimental Setup and Evaluation

The proposed system was evaluated using different datasets: the UrbanNav public dataset (Hsu et al., 2023) and a custom dataset collected with our intelligent vehicle platform. The UrbanNav dataset, captured in Hong Kong's complex urban environment, includes GNSS, INS, camera, and LiDAR measurements, with ground truth system offering centimeter-level accuracy. Our custom platform, equipped with a RealSense D435 camera, Xsens MTi-G-710 IMU, and Ublox M8T GNSS receiver, was used to conduct real-world experiments in two scenarios: a campus garden and a challenging urban road network in Hong Kong. These environments featured dynamic objects, GNSS-degraded areas, and varying levels of complexity.

For evaluation, we employed both qualitative and quantitative metrics. The feature grading module was qualitatively assessed by analyzing the assignment of feature grades, with dynamic features excluded under the static world assumption. Quantitative analysis focused on positioning accuracy, measured using the Root Mean Square Error (RMSE) of Relative Pose Error (RPE). To demonstrate the effectiveness of the proposed framework, our positioning method is compared with the state of the art methods.

### 4.2 Smart Feature Grading Performance

The feature grading module is designed to identify and retain only absolute static features while filtering out dynamic ones. Its performance was evaluated through qualitative analysis, with results visualized in Figure 4. Features are categorized as follows: blue for detected but untracked points, green for eliminated dynamic features, yellow for tracked fixed features, and red for absolute static features. Dynamic objects, such as moving vehicles and pedestrians, were accurately excluded, while static features (e.g., traffic lights, signs) were retained for backend optimization. In Figure 4, frames (a) and (d) depict static scenes with red features, while frames (b) and (c) demonstrate the system's ability to filter dynamic features, even when semantic detection fails. Blue features represent newly added points to maintain a consistent feature count. In the first row of images depicted in Figure 4, various scenarios are observed for a similar segment. In the captured frame (a), all objects within the scene remain static, resulting in the appearance of red-colored features. However, in frames (b) and (c), the vehicles within the view start to move, including our driving vehicle, which is partially visible. In frame (b), the object detection model successfully detects our vehicle, and the associated features are filtered out as dynamic features and represented as green-colored features. This highlights the effectiveness of our HGCC constraint to filter out the semantic masked potential dynamic features. The absolute static features from the fickle and stable feature classes, which are depicted as red points, were recognized by our dynamic features discarding scheme present across all frames. Compared with conventional methods, only these two categories of features form the fundamental basis for subsequent optimization in the backend, ensuring the accurate transformation matrix without any interruptions from dynamic features.

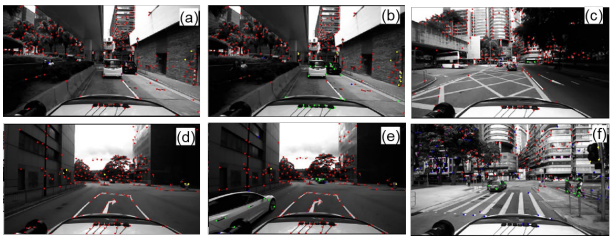


Figure 4. The grading of features obtained from our semantic front-end across multiple scenes. Detected but untracked features are depicted in blue, while green, yellow, and red indicate eliminated dynamic features, tracked fixed features, and absolute static features outside the masks, respectively.

### 4.3 Analysis of the Effect of Geometry Distribution of Features

The spatial distribution of features in each frame varies based on the presence of surrounding objects and the removal of dynamic points. This section analyzes the effects of uneven feature distribution caused by the elimination of dynamic objects in highly dynamic environments. Figure 5 illustrates the feature distribution and weight coefficients $M$ for selected frames, showing that frames with well-distributed features around the camera pose exhibit higher $M$ values, reflecting better spatial accuracy and pose estimation. Table 1 presents corresponding relative positioning errors, for instance, in Frame (c), RMSE improved from 0.536 meters to 0.457 meters after considering geometry distribution. Similarly, in Frame (e), RMSE significantly decreased from 0.802 meters to 0.207 meters, demonstrating greater accuracy gains in poorly distributed scenarios compared to well-distributed ones. These results emphasize the importance of incorporating geometry distribution coefficients to reduce uncertainty and enhance state estimation performance.

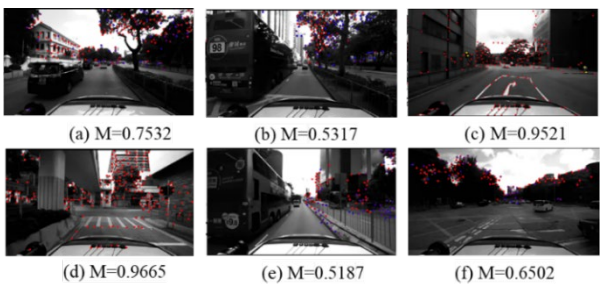


Figure 5. Geometry distribution of the features in different frames with the weight distribution coefficients.

| *RMSE* | M | Proposed_w/o M | Proposed_M | Imp.% |
|---|---|---|---|---|
| ***Frame (c)*** | 0.952 | 0.536 | 0.457 | 14.73 |
| ***Frame(a)*** | 0.753 | 0.542 | 0.367 | 32.28 |
| ***Frame (e)*** | 0.518 | 0.802 | 0.207 | 74.18 |

Table 1. Positioning performance at different geometry distribution's weight coefficients across different frames.

### 4.4 Evaluation of the Localization Performance

To ensure a comprehensive validation, we conducted these evaluations using the publicly available UrbanNav dataset and our experiment that was conducted in the complex urban environment of the Hong Kong Road network. Our method was validated using the UrbanNav dataset, which contains dynamic objects in approximately 75% of frames, posing challenges for state estimation due to incorrect data associations. Figure 6 shows a heatmap of relative positioning errors for state of the art methods, highlighting the effectiveness of our method. Table 2 provide quantitative results, showing that our method improved RMSE by 39.30% over the baseline VINS-Mono thanks to the adaptive covariance estimator. In contrast, DynaVINS performed poorly (1.598 meters RMSE), struggling with hyperparameter tuning and dynamic feature elimination. Compared to Wu et al. (2022), which removes all detected features without considering motion status, our method improved accuracy by 18% by leveraging motion analysis and addressing feature distribution. Unlike pixel-level segmentation (Mahmoud et al., 2024b), our method uses object detection masks, reducing computational overhead while effectively eliminating dynamic features. Overall, our approach enhances accuracy and robustness in dynamic environments.

| Strategies | RMSE | Mean | STD | Impro. % |
|---|---|---|---|---|
| (Qin et al., 2018) | 1.154 | 0.74 | 0.884 | - |
| (Wu et al., 2022) | 0.891 | 0.531 | 0.715 | 22.70% |
| (Song et al., 2022) | 1.598 | 1.121 | 1.139 | -38.4% |
| **Our method** | **0.702** | **0.435** | **0.551** | **39.30%** |

Table 2. Performance comparison of the proposed method with SOTA methods using the UrbanNav dataset.

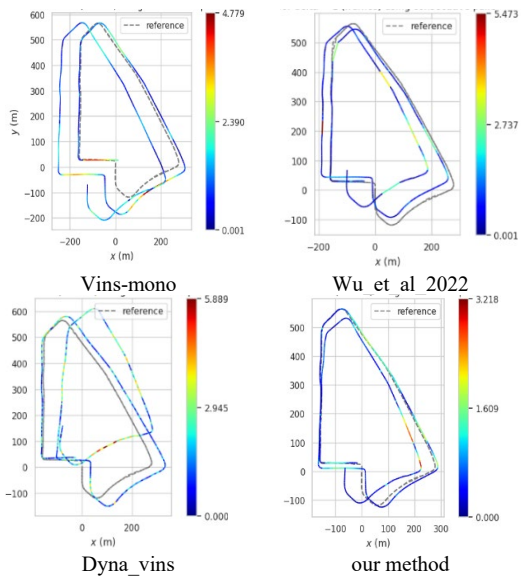


Figure 6. Accuracy heat map illustrates the positioning errors for the comparison methods.

To thoroughly assess the performance of the proposed method, we conducted our experiment in a complex outdoor urban driving environment characterized by high dynamics, with approximately 50% of the objects in motion. Figure 7 shows a heatmap of positioning errors for five methods, highlighting the reduced drift achieved by our approach. Quantitative results in Table 3 reveal that the baseline VINS-Mono achieved an RMSE of 1.423 meters but suffered from pose deterioration in static scenes with dynamic objects. Our method improved RMSE by 10.26% by filtering dynamic features and retaining only static ones. Further enhancement was achieved by addressing feature distribution skewness using an adaptive covariance estimator and IMU reprojection constraints with an improved RMSE of 12.09%. Our method outperformed Wu et al. (2022) by 9%, as their approach removed all detected features without considering motion status or feature distribution. DynaVINS performed poorly (1.586 meters RMSE), struggling in dynamic environments. Overall, the results introduced in this section highlight the effectiveness of our proposed method in reducing drifting and improving positioning accuracy and enhancing the system's robustness.

| Strategies | RMSE | Mean | STD | Improv. |
|---|---|---|---|---|
| **VINS_mono** | 1.423 | 0.675 | 1.253 | - |
| **Wu et al 2022** | 1.368 | 0.655 | 1.201 | 3.86% |
| **Dyna_vins** | 1.586 | 0.78 | 1.381 | -11.45% |
| **Ours*** | 1.277 | 0.627 | 1.111 | 10.26% |
| **Our overall method** | **1.251** | **0.61** | **1.092** | **12.09%** |

*Ours: VINS with only our smart feature grading and filtering

Table 3. Performance comparison of the proposed method with different methods using our real-world dataset.

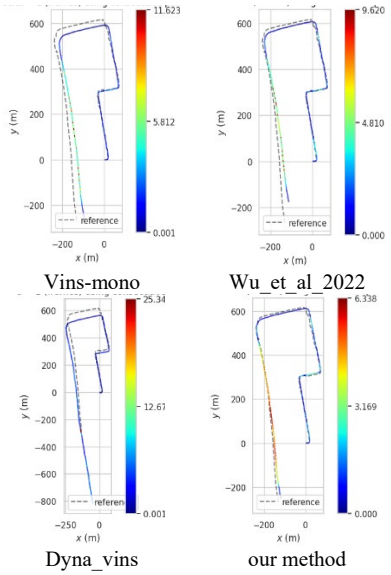


Figure 7. Accuracy heat map illustrates the positioning errors for five local comparison methods.

## 5. Processing Time Efficiency

We tested our system's computational efficiency using the ROS framework on a laptop with an Intel® Core™ i9-10900 CPU, NVIDIA GeForce RTX 3060 GPU, and 64 GB RAM. We compared our method's running time with Vins-Mono, focusing on time per frame across modules. Vins-Mono averaged 73 ms/frame, while our method added 18 ms/frame, maintaining real-time performance. The object detection module consumed the largest increase at 17 ms/frame, while feature grading took 0.67 ms/frame. The optimization module differed by 0.4 ms/frame due to an IMU reprojection constraint and adaptive geometry-based estimator. Despite the overhead, our method ensures accurate feature tracking and robustness, achieving real-time performance suitable for timely and accurate processing.

## 6. Conclusion

We propose a novel VINS method that leverages a smart feature grading module to overcome cautious and excessive dynamic feature removal, effectively handling the complexities of dominant and ambiguous dynamic objects. The proposed system integrates semantic-aware and multi-level geometric constraints, a VI-based motion consistency constraint, and an auto-adaptive covariance estimation method to enhance feature tracking and pose estimation accuracy. Experimental validation in dynamic urban environments demonstrates the method's superior accuracy and robustness, with minimal impact on computational time, ensuring real-time processing. Future work will focus on integrating additional sensors to further enhance system performance in challenging environments.

## References

Adham, M., W. Chen, Y. Li, and T. Liu. 2024. Towards Robust Global VINS : Innovative Semantic- Aware and Multi-Level Geometric Constraints Approach for Dynamic Feature Filtering in Urban Environments. IEEE Trans. Intell. Veh. PP: 1–24. doi: 10.1109/TIV.2024.3487593.

Agarwal, Sameer, Mierle, K. 2023. Ceres Solver. https://github.com/ceres-solver/ceres-solver.

Ali, E., W. Xu, and X. Ding. 2021. Spatiotemporal variability of dune velocities and corresponding uncertainties, detected from optical image matching in the north sinai sand sea, egypt. Remote Sens. 13(18). doi: 10.3390/rs13183694.

Campos, C., R. Elvira, J.J.G. Rodriguez, J.M.M. Montiel, and J.D. Tardos. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. IEEE Trans. Robot. 37(6): 1874–1890. doi: 10.1109/TRO.2021.3075644.

Cheng, J., Y. Sun, and M.Q.H. Meng. 2019. Improving monocular visual SLAM in dynamic environments: an optical-flow-based approach. Adv. Robot. 33(12): 576–589. doi: 10.1080/01691864.2019.1610060.

Fan, Y., H. Han, Y. Tang, and T. Zhi. 2019. Dynamic objects elimination in SLAM based on image fusion. Pattern Recognit. Lett. 127: 191–201. doi: 10.1016/j.patrec.2018.10.024.

Forster, C., L. Carlone, F. Dellaert, and D. Scaramuzza. 2017. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. IEEE Trans. Robot. 33(1): 1–21. doi: 10.1109/TRO.2016.2597321.

Fu, D., H. Xia, and Y. Qiao. 2021. Monocular visual-inertial navigation for dynamic environment. Remote Sens. 13(9). doi: 10.3390/rs13091610.

Geneva, P., K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. 2020. OpenVINS : A Research Platform for Visual-Inertial Estimation. IEEE International Conference on Robotics and Automation (ICRA). IEEE, Paris, France. p. 4666–4672

Hartley, R., and A. Zisserman. 2003. Multiple view Geometry in Computervision.

He, M., and R.R. Rajkumar. 2021. Extended VINS-Mono: A Systematic Approach for Absolute and Relative Vehicle Localization in Large-Scale Outdoor Environments. IEEE Int. Conf. Intell. Robot. Syst.: 4861–4868. doi: 10.1109/IROS51168.2021.9636776.

Hsu, L.T., F. Huang, H.F. Ng, G. Zhang, Y. Zhong, et al. 2023. Hong Kong UrbanNav: An Open-Source Multisensory Dataset for Benchmarking Urban Navigation Algorithms. Navig. J. Inst. Navig. 70(4). doi: 10.33012/navi.602.

Li, Y., W. Chen, J. Wang, and X. Nie. 2023. Precise Indoor and Outdoor Altitude Estimation Based on Smartphone. IEEE Trans. Instrum. Meas. 72: 1–11. doi: 10.1109/TIM.2023.3315391.

Liu, J., X. Li, Y. Liu, and H. Chen. 2022. RGB-D Inertial Odometry for a Resource-Restricted Robot in Dynamic Environments. IEEE Robot. Autom. Lett. 7(4): 9573–9580. doi: 10.1109/LRA.2022.3191193.

Lucas, B.D., and T. Kanade. 1981. Iterative Image Registration Technique With an Application To Stereo Vision. IEEE. p. 121–130

Mahmoud, M., M. Abd Rabbou, and A. El Shazly. 2022a. Land Vehicle Navigation Using Low-Cost Integrated Smartphone GNSS Mems and Map Matching Technique. Artif. Satell. 57(3): 138–157. doi: 10.2478/arsa-2022-0007.

Mahmoud, M., W. Chen, Y. Yang, and Y. Li. 2024a. Automated BIM generation for large-scale indoor complex environments based on deep learning. Autom. Constr. 162(March): 105376. doi: 10.1016/j.autcon.2024.105376.

Mahmoud, M., W. Chen, Y. Yang, T. Liu, and Y. Li. 2024b. Leveraging Deep Learning for Automated Reconstruction of Indoor Unstructured Elements in Scan-to-BIM. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. p. 479–486

Mahmoud, A., M.G. Mohamed, and A. El Shazly. 2020. Railway Tracks Detection of Railways Based on Computer Vision Technique and Gnss Data. ICCSTE’20. p. 1–8

Mahmoud, A., M.G. Mohamed, and A. El Shazly. 2022b. Low-cost framework for 3D reconstruction and track detection of the railway network using video data. Egypt. J. Remote Sens. Sp. Sci. 25(4): 1001–1012. doi: 10.1016/j.ejrs.2022.11.001.

Mansour, A., and W. Chen. 2022. SUNS: A User-Friendly Scheme for Seamless and Ubiquitous Navigation Based on an Enhanced Indoor-Outdoor Environmental Awareness Approach. Remote Sens. 14(20). doi: 10.3390/rs14205263.

Mohammed, E., N. Elshaboury, E. Ali, G. Alfalah, A. Mansour, et al. 2023. A Hyper Parametrized Deep Learning Model for Analyzing Heating and Cooling Loads in Energy Efficient Buildings. Proceedings of the International Conference on New Trends in Applied Sciences

Qin, T., P. Li, and S. Shen. 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Trans. Robot. 34(4): 1004–1020. doi: 10.1109/TRO.2018.2853729.

Rabbou, M.A., M. Mahmoud, and A. El Shazly. 2021. Performance Evaluation of Single Frequency PPP using Smartphone’s Raw GNSS Observations. Al-Azhar Univ. Civ. Eng. Res. Mag. 43(2).

Redmon, Ja., and A. Farhadi. 2018. YOLOv3: An Incremental Improvement. http://arxiv.org/abs/1804.02767.

Reginald, N., O. Al-Buraiki, B. Fidan, and E. Hashemi. 2022. Confidence Estimator Design for Dynamic Feature Point Removal in Robot Visual-Inertial Odometry. IECON Proceedings (Industrial Electronics Conference). IEEE. p. 1–6

Reid, T.G.R., S.E. Houts, R. Cammarata, G. Mills, S. Agarwal, et al. 2019. Localization Requirements for Autonomous Vehicles. SAE Int. J. Connect. Autom. Veh. 2(3): 1–16. doi: 10.4271/12-02-03-0012.

Shi, J., and C. Tomasi. 1994. Good Features to Track. Image (Rochester, N.Y.). p. 593–600

Song, S., H. Lim, A.J. Lee, and H. Myung. 2022. DynaVINS: A Visual-Inertial SLAM for Dynamic Environments. IEEE Robot. Autom. Lett. 7(4): 11523–11530. doi: 10.1109/LRA.2022.3203231.

Sun, Y., M. Liu, and M.Q.H. Meng. 2017. Improving RGB-D SLAM in dynamic environments: A motion removal approach. Rob. Auton. Syst. 89: 110–122. doi: 10.1016/j.robot.2016.11.012.

Tan, W., H. Liu, Z. Dong, G. Zhang, and H. Bao. 2013. Robust monocular SLAM in dynamic environments. 2013 IEEE Int. Symp. Mix. Augment. Reality, ISMAR 2013: 209–218. doi: 10.1109/ISMAR.2013.6671781.

Wang, J., X. Mi, W. Chen, H. Luo, A. Mansour, et al. 2024. Tightly Coupled Bluetooth Enhanced GNSS/PDR System for Pedestrian Navigation in Dense Urban Environments. IEEE Trans. Instrum. Meas. 73: 1–13. doi: 10.1109/TIM.2024.3481547.

Wu, X., F. Huang, Y. Wang, and H. Jiang. 2022. A VINS Combined with Dynamic Object Detection for Autonomous Driving Vehicles. IEEE Access 10(July): 91127–91136. doi: 10.1109/ACCESS.2022.3202161.

Zhang, C., T. Huang, R. Zhang, and X. Yi. 2021. PLD-SLAM: A new RGB-D SLAM method with point and line features for indoor dynamic scene. ISPRS Int. J. Geo-Information 10(3). doi: 10.3390/ijgi10030163.

Zhang, L., L. Wei, P. Shen, W. Wei, G. Zhu, et al. 2018. Semantic SLAM based on object detection and improved octomap. IEEE Access 6: 75545–75559. doi: 10.1109/ACCESS.2018.2873617.