Expediting the Convergence of Global Localization of UAVs through Forward-Facing Camera Observation

Zhenyu Li 1,2, Xiangyuan Jiang 1,*, Sile Ma 1, Xiaojing Ma 1, Zhenyi Lv 2, Hongliang Ding 2, Haiyan Ji 2 and Zheng Sun 2

1 Institute of Marine Science and Technology, Shandong University, Qingdao 266237, China; 202221077@mail.sdu.edu.cn (Z.L.); masile@sdu.edu.cn (S.M.); maxiaojing@sdu.edu.cn (X.M.)
2 Shenzhen Keweitai Enterprise Co., Ltd., Shenzhen 518126, China; lzy139029@keweitai.com (Z.L.); danny.ding@keweitai.com (H.D.); haiyan.ji@keweitai.com (H.J.); zheng.sun@keweitai.com (Z.S.)

* Correspondence: xyjiang@sdu.edu.cn

Abstract: In scenarios where the global navigation satellite system is unavailable, unmanned aerial vehicles (UAVs) can employ visual algorithms to process aerial images. These images are integrated with satellite maps and digital elevation models (DEMs) to achieve global localization. To address the localization challenge in unfamiliar areas devoid of prior data, an iterative computation-based localization framework is commonly used. This framework iteratively refines its calculations using multiple observations from a downward-facing camera to determine an accurate global location. To improve the rate of convergence for localization, we introduced an innovative observation model. We derived a terrain descriptor from the images captured by a forward-facing camera and integrated it as supplementary observation into a point-mass filter (PMF) framework to enhance the confidence of the observation likelihood distribution. Furthermore, within this framework, the methods for the truncation of the convolution kernel and that of the probability distribution were developed, thereby enhancing the computational efficiency and convergence rate, respectively. The performance of the algorithm was evaluated using real UAV flight sequences, a satellite map, and a DEM in an area measuring 7.7 km × 8 km. The results demonstrate that this method significantly accelerates the localization convergence during both takeoff and ascent phases as well as during cruise flight. Additionally, it increases localization accuracy and robustness in complex environments, such as areas with uneven terrain and ambiguous scenes. The method is applicable to the localization of UAVs in large-scale unknown scenarios, thereby enhancing the flight safety and mission execution capabilities of UAVs.

Keywords: UAV; localization; convergence; forward-facing camera; truncation; PMF; large scale

1. Introduction

Typically, an unmanned aerial vehicle (UAV) relies on navigation information from a global navigation satellite system (GNSS) to achieve global localization during flight. However, UAVs are exceptionally susceptible to communication interference that can disrupt GNSS signals, thereby posing a significant challenge to flight safety. Consequently, UAV localization in GNSS-denied environments has garnered considerable interest among researchers. Drawing inspiration from bionics [1], UAVs can utilize visual information and landmarks for localization and navigation in environments where GNSS is unavailable. Localization encompasses both relative and global methods. Visual–inertial odometry is commonly used for relative localization [2–4] but is plagued by significant drift issues [5]. In contrast, global localization provides latitude and longitude information, which aids in effectively mitigating the drift in relative localization and facilitates precise long-distance navigation.

Global localization depends on the scale of the map database used. After acquiring initial GNSS data or manually configuring prior global localization information, UAVs
correlate aerial images with a small-scale map database during the localization process to determine their global locations. However, in unknown areas or under severe communication constraints, UAVs may lack access to prior global localization information. Accurately and rapidly determining global localization without prior information or with only weak prior information remains a challenging problem due to potential increases in database matching scale, matching uncertainty, and computational cost.

The primary challenge addressed in this study is the initial global location estimation in unknown areas, often referred to as the “start-up robot problem” [6]. Theoretically, a viable and widely used approach involves the iterative Bayesian filtering framework, which employs either the recursive Monte Carlo localization (MCL) method [7] or the point-mass filter (PMF) method [8,9]. MCL, also referred to as particle filtering, represents samples within the state space through a certain number of particles. Each particle is assigned a weight in accordance with the likelihood probability distribution, and the probability distribution of location is computed based on the processing of the particles. Compared with MCL, PMF has differences in the representation and calculation mode of the state space and probability distribution. It discretizes the entire state space into grids with a certain resolution. Meanwhile, it presents and calculates the probability distribution of location based on the grids. The two approaches do not have strict constraints on the initial position of the robot, and both achieve efficient and robust localization through iterative prediction–observation steps. The two approaches possess their respective advantages. Typically, the computational cost of MCL is relatively lower, whereas PMF demonstrates greater robustness [8]. Predictions are typically derived from relative motion data obtained using visual–inertial odometry, while observations are acquired by matching aerial images captured by a downward-facing camera to a satellite map or digital elevation model (DEM). Upon the convergence of these iterations, it can be deduced that the UAV has acquired a reliable global location. The robustness of the observations is crucial for the successful convergence of the localization process [10]. In this study, we employed an iterative Bayesian filtering framework focusing on improving the observation component. We hypothesize that incorporating additional reliable observations will significantly improve the convergence rate of the localization iterations. To this end, we used a forward-facing camera and developed robust descriptors and matching methods based on its observation model, as opposed to the more common downward-facing cameras. The rationale for employing a forward-facing camera is discussed as follows:

- Unlike a downward-facing camera that has a limited observational range, a forward-facing camera mounted on a UAV can capture extensive scene information ahead, even at lower flight altitudes;
- The forward-facing camera is utilized not only for localization but also for detecting obstacles, thereby improving flight safety;
- In some rescue and inspection application scenarios, the forward-facing camera can be used simultaneously for localization as well as for the detection and identification of objects. This will enhance the task execution ability under GNSS denial circumstances.

However, challenges arise when matching aerial images from a forward-facing camera with satellite maps due to differences in data collection equipment and perspectives, similar to those encountered with downward-facing cameras. Additionally, discrepancies in lighting, color, and other factors between satellite maps and aerial images, coupled with the inherent blurriness in natural scenes (e.g., forests and deserts), lead to significant feature disparities. Thus, robust feature extraction that can handle these differences and scene ambiguities during matching is crucial when using observations from a forward-facing camera.

In this study, we introduce a method to accelerate the convergence rate of localization iterations using a forward-facing camera on a UAV. This approach employs a DEM, which is impervious to seasonal and lighting changes, as the reference database. It involves extracting features from images captured by the forward-facing camera, mapping these features to a specially designed descriptor, and then quantifying the observation probability.
distribution by matching these descriptors. The observation probability distributions are integrated into a PMF framework, which enhances the convergence rate, accuracy, and robustness of the localization process. Additionally, we have developed two truncation techniques within the PMF framework to reduce computational complexity and further improve the localization convergence rate. Notably, our proposed method does not provide a complete localization solution but rather supplements the general localization approach. This is particularly significant for enhancing global localization performance in environments where GNSS are unavailable. The main contributions of this study are outlined as follows:

1. We proposed a method for constructing a robust 2.5D grid feature descriptor using two consecutive frames captured by a forward-facing camera to construct a two-view structure from motion (SFM) model. The 3D points derived from the SFM were projected onto the 2.5D grid feature descriptor, with varying weights assigned to each grid based on odometry noise and the 3D points. This descriptor offers terrain information over a large area and provides more comprehensive observations than those obtained using a downward-facing camera;
2. We define global localization as a state estimation problem based on the PMF. We quantified the matching information between the described features and the DEM as a probability distribution, integrating it into the PMF framework as an additional observation. We developed two truncation methods to enhance computational efficiency and accelerate the convergence rate of localization. The first method truncated the convolution kernel size for state prediction, and the other truncated the posterior state probability distribution using a sliding window. These approaches aim to improve the computational efficiency and expedite the convergence ratio of localization;
3. The experimental results from real flight sequences indicate that our method can accelerate the convergence rate of localization during the takeoff, ascent, and cruise flight stages. Additionally, the method effectively enhances localization accuracy and robustness in complex scenarios such as uneven terrain and ambiguous scenes;
4. The proposed method demonstrates high scalability and the potential for seamless integration into any Bayesian filtering state estimation framework. Moreover, the observational model can be extended to include multiple pinhole or fisheye cameras, which further enhance the localization performance.

The structure of this paper is organized as follows: Section 2 presents research related to our work. Section 3 introduces the methodology of this study, and Section 4 substantiates the performance of the method through comparative experiments. Section 5 presents the detailed results and discussion. Section 6 provides the conclusions and future prospects.

2. Related Works

The global localization of UAVs using visual methods is a significant research area within computer vision. Recent comprehensive reviews [11,12] outlined diverse research methodologies and approaches to tackle this challenge. Several methods presuppose that UAVs have already obtained initial global locations or restrict matching to specific map areas [13–16]. Our research primarily addresses the challenges of initial global localization of UAVs in uncharted territories, often referred to as the start-up robot problem. A common feature of these studies is that UAVs lack precise prior global localization information and rely instead on offline maps or elevation data for coarse-to-fine global localization.

Several studies have investigated scene-matching techniques between aerial images and large-scale satellite maps to address global localization issues. The effectiveness of scene matching hinges on extracting features that are invariant to factors such as acquisition time, lighting conditions, and perspectives. Conventional scene-matching methods include global feature-based techniques such as normalized cross-correlation (NCC) [17] and mutual information (MI) [14], as well as local feature-based approaches such as scale-invariant feature transform (SIFT) and speeded-up robust features (SURF). However, these methods face challenges owing to significant differences in image illumination and features [18].
recent years, advancements in deep learning have enhanced the representation of local or global features through neural networks [19–22], which improved the robustness of feature matching. Furthermore, similar studies have framed the global localization problem as a cross-view image retrieval issue [23–25], focusing on feature representation and matching across various perspectives and lighting conditions. Although most previous research has focused on the extraction and matching of image features, variations in image characteristics can lead to failures in scene matching or image retrieval. Therefore, relying solely on these methods for global localization in large-scale scenarios lacks robustness.

Several studies have advanced the field of scene matching and image retrieval by incorporating the dynamics of visual–inertial odometry and using iterative computational frameworks to estimate global locations across extensive areas. These enhancements have significantly improved the robustness of localization systems, enabling UAVs to achieve precise and stable localization across various environments. Choi et al. [26] introduced a UAV localization method utilizing a building ratio map (BRM), which involves extracting building regions via semantic segmentation, calculating the building proportion in image frames, and constructing a rotation-invariant global feature descriptor. They also developed an iterative localization framework with defined convergence conditions. In a 2.2 km × 2.6 km area without a known initial location, their method achieved localization convergence with a root-mean-square error of only 12 m after 27 iterations. Mantelli et al. [27] developed an observation model using the abBRIEF descriptor, which incorporates the CIE color space and an image-denoising strategy to enhance robustness against lighting variations. This model was integrated into the MCL iterative localization framework and tested through extensive flight experiments in three different regions. The results showed that localization could converge within 30 to 50 iterations, even without initial coordinates, with a minimum average location error of 17 m throughout the flight.

Kinnari et al. [22] created a convolutional neural network model that was robust to appearance variations due to seasonal changes. In further research [28], they proposed a global localization method for large-scale maps that utilized seasonally invariant features. This method matches the aerial images taken by a downward-facing camera with a satellite map and quantifies them as the observation probability distribution under a PMF. The method significantly improves localization accuracy in scenarios with extensive scene blurring. Localization experiments conducted on a 100 km² map exhibited that this method could converge to an average localization error of 12.6 to 18.7 m within 23.2 to 44.4 iterations, representing a highly advanced approach in the current field.

We conducted a comprehensive analysis of the studies mentioned. To ensure consistency with the observational perspective of satellite maps, all aerial images used in the research were captured using a downward-facing camera or were transformed into a downward perspective through geometric transformations. From a biological perspective, we contend that the data derived from these observations are somewhat limited. Moreover, these studies exclusively utilized satellite maps and failed to incorporate detailed terrain data, such as elevation information. Given these considerations, the problem of global localization for visually based UAVs remains unresolved, indicating potential for further optimization and improvement.

3. Method

3.1. Overview of the Method

The complete robot localization problem entails estimating the 6-DoF state, which includes both 3-DoF position and 3-DoF orientation. To simplify the problem by reducing the dimensionality of the state space, it is assumed that the yaw angle and altitude are determinable from magnetometer and barometer measurements, respectively, whereas the pitch and roll angles are derived from IMU measurements. Therefore, we focused only on the 2-DoF localization problem. We define the state variables of the UAV in the global coordinate system as follows:
\[ \chi_t = [x_t, y_t]^T \]  

where \( x \) and \( y \) represent the longitude and latitude, respectively, and \( t \) represents the time index.

Given the initial uncertainty and potential for ambiguous features in the environment, the problem of global localization in unknown areas is typically formulated as a nonparametric Bayesian filtering problem [29]. This involves iteratively computing the posterior probability distribution of the state based on control information, observation data, and the map. In this framework, odometry-derived observation information is used to predict the probability distribution of the state at time \( t \) based on its distribution at time \( t - 1 \):

\[ \chi_t = \chi_{t-1} + u_{t-1} + w \]  

Specifically, the location displacement from time \( t - 1 \) to time \( t \) is denoted as \( u_{t-1} \), with its associated noise represented by \( w \). During the observation process, a similarity calculation is employed, yielding the following observations:

\[ Z_t = h(\chi_t) + v \]

where \( h(\chi_t) \) represents a function that maps the current location of the UAV to a similarity value observation, \( h(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R} \), and the observation noise is denoted by \( v \).

Similarity is calculated using techniques such as image feature matching or point-cloud matching. Bayesian filtering then updates the posterior probability distribution of the system state by computing the distribution based on \( h(\chi_t) \). Particle filters (also known as MCL) and PMF are nonparametric Bayesian filtering solutions commonly used for robot localization [29]. Kinnarne et al. [28] discussed the advantages and disadvantages of both methods for global localization, ultimately favoring PMF. Similarly, we opted to employ PMF, which calculates the probability distribution on a discrete grid rather than using particles, ensuring consistent performance, constant computation cost, and constant memory requirements. Utilizing a low-dimensional state space significantly reduced the computational demands of the PMF calculation process. The proposed method was integrated into a comprehensive visual localization framework based on PMF, as depicted in Figure 1. In PMF, the probability distribution of the global location is discretized into a grid map with \( N \) uniformly distributed grids. We define \( \delta \) as the side length of a single grid cell and \( k \) as the grid index.

The proposed method is divided into three main components. The first component (Section 3.2) details the construction of a 2.5D grid feature descriptor using a two-view SFM model; the second component (Section 3.3) outlines the similarity matching process and its transformation into a likelihood measure; and the third component (Section 3.4) discusses the fusion of multi-source observations and the enhancement of convergence performance. The focus is on developing the observational model and principles to improve convergence performance. Consequently, the visual–inertial odometry and image map matching, illustrated in Figure 1, were implemented using established and conventional methods. These techniques were solely utilized to provide state prediction information and individual observations for the PMF model without exploring specific methodological details.
Figure 1. Block diagram of the proposal. The first part (Section 3.2) describes the method of constructing a 2.5D descriptor based on two-view SFM; the second part (Section 3.3) outlines the matching procedure and the approach for obtaining the likelihood distribution; and the third part (Section 3.4) introduces the technique of fusion observations and accelerated iteration calculation based on PMF.

3.2. Methodology for Constructing Descriptor

3.2.1. Methodology for Constructing 3D Point Cloud Using Two-View SFM

The SFM method facilitates sparse point-cloud reconstruction by capturing multiple images from different poses using a single camera and performing feature matching between these images [30]. Previous research [31] has shown that feature matching is effective even under challenging lighting conditions, indicating that the SFM method possesses a certain level of robustness against changes in illumination. Furthermore, the reconstructed point cloud captures terrain information, which is generally temporally stable and minimally affected by seasonal variations. This stability suggests that matching the 3D point cloud reconstructed by SFM with a DEM is theoretically feasible.

The two-view SFM method is an efficient and practical technique for constructing 3D point clouds. This method involves using two consecutive keyframes captured by a forward-facing camera, as illustrated in Figure 2. To effectively capture these keyframes, the flight movements of the UAV must satisfy specific horizontal or vertical displacement criteria. During the two-view SFM process, feature points are detected and initially matched across the two frames. Subsequently, geometric constraints for the matching pairs of features are established based on the epipolar constraint model.

\[ p_2^T K^{-T} t^\hat{R} K^{-1} p_1 = 0 \]  

(4)

Specifically, \( t \) represents a 3-DoF translation vector and \( t^\hat{R} \) denotes the skew-symmetric matrix of \( t \). \( K \) denotes the intrinsic parameters of the camera and \( R \) represents the rotation matrix. The RANSAC method [32] is employed to extract a set of inlier points. The singular value decomposition (SVD) method is then used to solve for rotation and translation based on Equation (4), followed by the reconstruction of the 3D coordinates of the points through triangulation. However, a 3D point set reconstructed from two frames exhibits
scale ambiguity, which is a limitation inherently linked to the lack of degrees of freedom in the epipolar constraint model [30].

\[
\begin{align*}
\mathbf{t} &\equiv \begin{bmatrix} t_X \, & t_Y \, & t_Z \end{bmatrix}^T \in \mathbb{R}^3, \\
\mathbf{R} &\equiv \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \\
\mathbf{K} &\equiv \begin{bmatrix} k_{11} & k_{12} & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \\
\mathbf{R}_0 &\equiv \mathbf{I}_3,
\end{align*}
\]

\( \mathbf{K} \) denotes the intrinsic parameters of the camera and \( \mathbf{R} \) represents the rotation matrix. The RANSAC method [32] is employed to extract a set of inlier points. The singular value decomposition (SVD) method is then used to solve for rotation and translation based on Equation (4), followed by the reconstruction of the 3D coordinates of the points through triangulation. However, a 3D point set reconstructed from two frames exhibits scale ambiguity, which is a limitation inherently linked to the lack of degrees of freedom in the epipolar constraint model [30].

**Figure 2.** Schematic of two-view SFM. (a) Two-view SFM process during the cruise flight stage, where the UAV maintains a constant altitude; (b) two-view SFM process during the ascend stage, where the UAV maintains a constant horizontal location.

The flight process is categorized into two stages: a vertical takeoff and ascent stage and a horizontal cruise stage. To realize the application of two-view SFM, when capturing the two images, the position of the camera needs to have a certain amount of translation, and this translation needs to be restored to the real scale. During the takeoff and ascend stages, the translation of the UAV mainly results from the change in altitude. Barometers and 1D laser range finders are capable of providing absolute altitude information and aboveground level information, respectively; therefore, the translation can be calculated through the measurement values at different altitudes. During the cruise flight stage, the translation of the UAV mainly results from the change in the horizontal position. By utilizing the sequence of video frames from the downward-facing camera and the distance-to-ground measurement from the 1D laser range finder, visual odometry with real scale can be established; thereby, we can obtain the change in the horizontal position. We defined a scale factor, denoted as \( s \). During various flight stages, the scale factor was computed based on different translations. Subsequently, the 3D point coordinates were multiplied by the scale factor to restore the real-scale coordinates. The calculation of the scale factor is as follows:

\[
s = \begin{cases} 
\frac{\Delta h}{|t_Z|}, & \text{During takeoff and ascend stage} \\
\frac{\sqrt{u_X^2 + u_Y^2}}{\sqrt{t_X^2 + t_Y^2}}, & \text{During cruise flight stage} 
\end{cases}
\]

where \( t_X, t_Y, \) and \( t_Z \) represent the 3 elements in the translation vector \( \mathbf{t} \). \( u_X \) and \( u_Y \) denote the horizontal displacement of the odometry, where \( \Delta h \) corresponds to the change in altitude.
3.2.2. Mapping of 3D Points to Descriptor

This approach also introduces a 2.5D descriptor based on terrain features, represented as an \( m \times m \) grid map where the scale of each grid aligns with that of the PMF. The 3D points reconstructed by SFM are transformed and mapped onto the corresponding grids of the descriptor through the coordinate transformation matrix. Meanwhile, the height of each 3D point is added to the altitude of the UAV to acquire the absolute altitude of each 3D point. Based on this, we calculate the average value of the absolute altitude of the 3D points within a single grid of the descriptor, and this value is stored in the grid as the elevation value. The descriptor utilizes the NED (North, East, Down) coordinate system, a local coordinate system centered on the current location of the UAV. Accordingly, the center grid of the descriptor corresponds to the location of the UAV. Upon acquiring a new keyframe, a two-view SFM is conducted, followed by the transformation of each 3D point from the camera coordinate system into the NED coordinate system.

\[
P_{i}^{\text{ned}} = R_{c}^{\text{ned}} P_{i}^{c} \tag{6}
\]

We derived the elevation information for each 3D point by incorporating the altitude of the UAV.

\[
\text{Alt}_i = \text{Alt}_{\text{UAV}} + P_{i}^{\text{ned}}(z) \tag{7}
\]

where \( i \) represents the index of a 3D point and \( R_{c}^{\text{ned}} \) denotes the transformation matrix from the camera coordinate system to the NED coordinate system, which can be derived from the pitch, roll, and yaw angles of the gimbal mounted on the UAV. \( P_{c}^{c} \) and \( P_{i}^{\text{ned}} \) represent the 3D point coordinates in the camera and NED coordinate systems, respectively, whereas \( P_{i}^{\text{ned}}(z) \) denotes the relative height of the 3D point in the NED coordinate system. In addition, \( \text{Alt}_{\text{UAV}} \) and \( \text{Alt}_i \) correspond to the altitude of the UAV and each 3D point, respectively. Using the N and E coordinates of the 3D point in the NED coordinate system along with the grid scale, each 3D point was mapped to its corresponding grid on the descriptor. Concurrently, the elevation of each 3D point was accumulated within each grid. After processing all 3D points, the average elevation for the points within each grid of the descriptor was calculated to determine the elevation value for each grid. Note that grids containing fewer points than a specified threshold were marked as invalid and excluded from further matching calculations.

3.3. Methodology for Matching and Calculating Likelihood

3.3.1. Expression of the Similarity between the Descriptor and the Patch

To prepare for descriptor matching with the DEM, the DEM was converted into a 2.5D grid map, referred to as the 2.5D DEM map. Figure 3a displays an example of a 2.5D DEM map derived from the DEM. The total number of grid cells \( N \) and the scale of each grid \( \delta \) in the 2.5D DEM map were aligned with those in the PMF, with each grid containing average elevation data for its respective area. The descriptor was then matched with the 2.5D DEM map to assess the likelihood of observation. This section elaborates on the methods used for matching and calculating likelihood.

As detailed in Section 3.2, the grid size in the descriptor matches that of the 2.5D DEM map, and the orientation of their coordinate systems is aligned. This alignment facilitates direct matching using a template matching method. A sliding window technique is employed to scan the 2.5D DEM map, selecting a patch of grid size \( m \times m \) centered on the current grid for comparison with the descriptor to evaluate their similarity. This method calculates the observational likelihood for each grid in the 2.5D DEM map, representing the PDF of observational likelihood.

Reference [33] discusses various similarity calculation methods. Inspired by the SAD algorithm, we developed a weighted method for similarity calculation that accounts for the uncertainties in odometry and DEM elevation data. Different weights are assigned to grids within the descriptor that are flagged as valid. The similarity between the descriptor
and the patch corresponding to the $k$-th grid in the 2.5D DEM map is calculated using the following formula:

$$(S_p)_k = \sum_{j=1}^{(M)_k} (w_j)_k (S_j)_k$$

(8)

where $M$ denotes the number of effective grids in the descriptor; $j$ corresponds to the grid index within the descriptor; $w_j$ represents the weight of an individual grid; and $S_j$ represents the similarity between a single grid in the descriptor and a single grid in a patch. The specific configurations for $w_j$ and $S_j$ are stated as follows.

**3.3.2. Design of $w_j$**

The 3D point location in the NED coordinate system is influenced by noise from various sources, including odometry, gimbal angles, and barometers. To illustrate this, we analyzed the noise in the horizontal location of a 3D point during a cruising flight by projecting the 3D point and the camera onto a horizontal plane, as depicted in Figure 4a. The variables Camera$(t - 1)$ and Camera$(t)$ represent the horizontal locations of the camera at two consecutive keyframe acquisition times, with the horizontal distance between them determined by the horizontal displacement of the odometry. In this case, please note that we are not using $\chi_i$ in this context, as Camera$(t)$ is three-dimensional and will be utilized for modeling the elevation noise in subsequent sections.

In Figure 4a, the noise in the horizontal location of the 3D point consists of angular noise from the gimbal’s yaw angle and horizontal displacement noise from the odometry. For simplicity, both noise sources are assumed to follow a Gaussian distribution with a mean of zero and standard deviations denoted as $\sigma_{odom}$ and $\sigma_{yaw}$, respectively. We also

---

**Figure 3.** (a) A 2.5D DEM map generated from DEM; (b) schematic of mapping, matching, and similarity computation. First, 3D points are mapped to the 2.5D descriptor, and the map patch corresponding to grid $k$ in the 2.5D DEM map is matched to the descriptor, then the similarity $(S_p)_k$ is calculated and the likelihood probability distribution $p(z_i(k) | \chi_i(k))$ is obtained.
modeled the horizontal location noise of the 3D point as a Gaussian distribution, with its standard deviation approximated by the following equation:

\[
\sigma_h \approx \sqrt{(D_h \tan(\sigma_{\text{yaw}}))^2 + \left(\frac{D_h \sigma_{\text{odom}}}{|C|}\right)^2}
\]  

(9)

where \(D_h\) represents the horizontal distance from 3D point to Camera\((t)\), and \(|C|\) denotes the displacement magnitude of odometry. Based on Equation (9), the greater the horizontal distance of a point from Camera\((t)\), the higher the noise in its horizontal location, and the standard deviation of the noise is directly proportional to the horizontal distance of the point from Camera\((t)\).

Figure 4. Modeling of horizontal location noise and elevation noise for 3D points. The orange dashed line indicates the boundary of the influence of angular noise on the position of a 3D point. (a) Modeling of horizontal location noise for 3D points: noise introduced by the yaw angle of the camera (left) and odometry (right). (b) Modeling of elevation noise for 3D points: noise introduced by the pitch angle of the camera (left) and barometric measurements (right). The gray area represents the distribution of the noise.

Considering this property, the significance of an individual grid cell in computing the similarity between descriptor and patch should diminish as the distance from the grid cell to the center of the descriptor increases. Based on Section 3.2, upon calculating the mean value, each grid could be represented by a point \(p\) located at the center of the grid. Similarly, we can utilize Equation (9) to characterize the standard deviation of horizontal location noise for the point \(p\), where \((\sigma_h)_j\) is defined as the standard deviation of the horizontal
location of one single grid, with $j$ corresponding to the grid index in accordance with Equation (8).

Considering that the probability density integral of point $p$’s horizontal location within a single grid cell reflects the likelihood of its true horizontal location being within that grid cell and noting that this value decreases as the point’s distance from the descriptor’s center increases, we conclude that this likelihood value indicates the contribution of the grid in computing similarity. Therefore, we use the probability density integral of point $p$’s horizontal location within a single grid cell to represent the weight:

$$w_j = \int_{-\delta/2}^{\delta/2} \int_{-\delta/2}^{\delta/2} \frac{1}{2\pi(\sigma_h)^2} e^{-\frac{x^2 + y^2}{2(\sigma_h)^2}} \, dx \, dy$$

(10)

3.3.3. Design of $S_j$

The similarity is computed by quantifying the difference in elevation between the individual grid cell in the descriptor and that in the patch. A smaller elevation difference indicates a higher similarity. Consistent with previous discussions, the elevation of each grid is represented by the elevation of the point $p$.

To demonstrate this concept, we analyze the elevation noise of 3D points during a cruising flight by projecting the 3D point and the camera onto a vertical plane, as depicted in Figure 4b.

The elevation noise at point $p$ is attributed to angular noise from the gimbal pitch angle and altitude noise from the barometer. Both types of noise are modeled as Gaussian distributions with a mean of zero and standard deviations denoted by $\sigma_{\text{pitch}}$ and $\sigma_{\text{baro}}$, respectively. The elevation noise at point $p$ is also modeled as a Gaussian distribution, with its standard deviation approximated as follows:

$$\sigma_e \approx \sqrt{(D \tan(\sigma_{\text{pitch}}))^2 + \sigma_{\text{baro}}^2}$$

(11)

where $D$ represents the spatial distance from a 3D point to Camera$(t)$. In addition to the elevation of the point $p$, there is noise in the elevation of the patch, which can be modeled as a Gaussian distribution with a mean of 0 and a standard deviation of $\sigma_{\text{map}}$. Considering both noise sources, the similarity calculation treats the elevation difference between individual grid cells in the descriptor and the patch as a Gaussian distribution with a mean of zero and a standard deviation that satisfies:

$$\sigma'_e = \sqrt{\sigma_e^2 + \sigma_{\text{map}}^2}$$

(12)

Let $(\sigma'_e)_j$ represent the standard deviation of the elevation for an individual grid, where the corresponding grid index $j$ is consistent with that in Equation (8). We directly employed the instantaneous probability as a similarity:

$$S_j = \frac{1}{\sqrt{2\pi(\sigma'_e)_j^2}} e^{-\frac{|E|^2}{2(\sigma'_e)_j^2}}$$

(13)

Specifically, $|E|$ represents the absolute value of the elevation difference between an individual grid cell in the descriptor and that in the patch. In the similarity calculation, it is a numerical value greater than zero, where a higher similarity indicates a greater likelihood of observation. Thus, the similarity between the descriptor and the patch can be directly used as the observational information for the $k$-th grid, i.e., $z_t(k) = (S_p)_k$. Figure 3b
illustrates the mapping, matching, and similarity computation process. The likelihood in
the PMF is derived by applying a normalization step to the similarity calculation results.

\[
p(z_t(k)|\chi_t(k)) = \frac{(S_p)_k}{\sum_{n=1}^{N}(S_p)_n}
\]  \hspace{1cm} (14)

### 3.4. Methodology for Multi-Observation Fusion and Truncation

#### 3.4.1. Multi-Observation Fusion Method Based on PMF

The PMF employs a Bayesian filtering technique to address challenges in terrain
navigation. PMF calculates the probability distribution of a UAV location and iteratively
updates this distribution with each new observation. It discretizes the state space into grids
and performs numerical approximations of the probability distributions and integrals using
Bayesian filtering. The fundamental recursive computational process based on discrete
grids operates as follows:

\[
p(\chi_t(k)|Z_{t-1}) = \sum_{n=1}^{N} p(w(\chi_t(k) - u_{t-1} - \chi_{t-1}(n)))p(\chi_{t-1}(n)|Z_{t-1}) \delta^2 \hspace{1cm} (15)
\]

\[
a_t = \sum_{n=1}^{N} p(z_t(n)|\chi_t(n))p(\chi_t(n)|Z_{t-1}) \delta^2 \hspace{1cm} (16)
\]

\[
p(\chi_t(k)|Z_t) = a_t^{-1} p(z_t(k)|\chi_t(k))p(\chi_t(k)|Z_{t-1}) \hspace{1cm} (17)
\]

\[
\hat{\chi}_t = \sum_{n=1}^{N} \chi_t(n)p(\chi_t(n)|Z_t) \delta^2 \hspace{1cm} (18)
\]

\[
P_t = \sum_{n=1}^{N} (\chi_t(n) - \hat{\chi}_t)(\chi_t(n) - \hat{\chi}_t)^T \sum_{n=1}^{N} p(\chi_t(n)|Z_t) \delta^2 \hspace{1cm} (19)
\]

Specifically, \(Z_t = z_{12}\) represents historical observations. Equation (15) represents the
state prediction process, where the relative displacement and displacement noise from
visual odometry are considered to derive the distributions of \(u_{t-1}\) and \(p_w\). Equation (17)
represents the posterior probability update, where \(a_t\) denotes the normalization parameter.
\(\hat{\chi}_t\) and \(P_t\) denote the mean and covariance of the final output state, respectively. In the
above equation, \(p(z_t|\chi_t)\) represents only the likelihood of a single observation, such as the
observation likelihood derived from the matching results between the downward-facing
camera image and the 2D satellite map. Assuming that each set of observation data includes
distinct inputs from both the downward-facing and forward-facing cameras and assuming
their independence, the confidence in these observations can be enhanced through fusion.
Within the Bayesian framework, the fused observation likelihood \(p(z_t|\chi_t)\) is proportional
to the product of the individual observation likelihoods.

\[
p(z_t|\chi_t) \propto p(z_t^d|\chi_t)p(z_t^f|\chi_t) \hspace{1cm} (20)
\]

where \(p(z_t^d|\chi_t)\) and \(p(z_t^f|\chi_t)\) represent the likelihood observations obtained from the
downward-facing and forward-facing cameras, respectively. Thus, Equations (16) and (17)
are transformed into the following:

\[
a_t = \sum_{n=1}^{N} p(z_t^d(n)|\chi_t(n))p(z_t^f(n)|\chi_t(n))p(\chi_t(n)|Z_{t-1}) \delta^2 \hspace{1cm} (21)
\]

\[
p(\chi_t(k)|Z_t) = a_t^{-1} p(z_t^d(k)|\chi_t(k))p(z_t^f(k)|\chi_t(k))p(\chi_t(k)|Z_{t-1}) \hspace{1cm} (22)
\]

#### 3.4.2. Truncation Method

We improved the computational efficiency and convergence rate of PMF using two
truncation mechanisms.
(a) Convolution kernel truncation

To demonstrate the computational complexity of the PMF, we considered a 2-dimensional PMF with a specified grid count \( N \). During the computation of the state prediction probability distribution in Equation (15), each grid undergoes a convolution calculation that requires time \( N \). The time complexity of this step is denoted by \( T(N) = O(N^2) \), representing the most computationally demanding phase of the PMF. To improve efficiency, we implement truncation based on the assumption that odometry noise follows a Gaussian distribution. This Gaussian distribution possesses a specific radius, indicating that during the state prediction process of a single grid (in Equation (15)), our focus can be restricted to the grids within a specific range, and the impact of the grids beyond this range can be disregarded. Based on the Gaussian distribution parameters of the odometry noise, an appropriate convolution kernel size can be determined, and the influence of the grids outside the convolution kernel can be ignored, thereby reducing the computational costs. It should be noted that the configuration of the size of the convolution kernel is highly critical. An overly small convolution kernel will result in incorrect state predictions, whereas a convolution kernel that is excessively large may not notably enhance the computational efficiency. Assuming the standard deviation of the odometry displacement noise is \( \sigma_{\text{odom}} \), we truncate the convolution kernel to a radius of \( 3\sigma_{\text{odom}} \), as depicted in Figure 5a. After truncation, the number of grids involved in the convolution decreases to \( 2\left(\frac{3\sigma_{\text{odom}}}{\delta} + 1\right)^2 \).

The time complexity of the state prediction can be expressed as follows.

\[
T(N) = O\left(\frac{2\left(3\sigma_{\text{odom}}/\delta + 1\right)^2}{\delta}N\right)
\]

For a large-scale map, it is reasonable to assume that \( \frac{2\left(3\sigma_{\text{odom}}/\delta + 1\right)^2}{\delta} \) is significantly smaller than the \( N \). Theoretically, truncating the convolution kernel can result in a substantial improvement in the computational efficiency of state prediction.

(b) Sliding Window-Based probability truncation

To expedite the convergence rate of the iterative process, we developed a method that truncates low-probability grids using a sliding window technique.

In PMF, both state prediction and posterior update processes influence the probability distribution. There are some grids whose probabilities remain extremely low throughout multiple iterations, and their existence forms a restriction on the convergence rate of covariance. Timely and accurate removal of low-probability grids can expedite the convergence of the state and reduce computational overhead. However, considering the observational noise and potential for ambiguous scenes, it is crucial not to truncate these grids too hastily, as this could lead to non-convergence. We introduced a sliding window truncation mechanism, illustrated in Figure 5b, where a sliding buffer of size \( s \) stores the posterior probability distribution of the most recent \( s \) frames. After updating the posterior probability distribution, a decision-making strategy is employed to truncate probabilities below a predefined threshold:

\[
p(\chi_{t}(k)|Z_t) = \begin{cases} 
0, & \forall t' \in [t-s,t], p(\chi_{t'}(k)|Z_{t'}) < \varepsilon \\
p(\chi_{t}(k)|Z_t), & \text{otherwise}
\end{cases}
\]

where \( \varepsilon \) denotes the user-defined truncation threshold. After truncation, the probability density distribution is renormalized to maintain the integrity of the cumulative distribution function. This truncation approach, which considers multiple probability density distributions [8], is more robust than methods relying on a single distribution, thus reducing the risk of accidental convergence failure. It should be noted that the quantity of sliding windows is of crucial significance. An insufficient number of sliding windows will result in incorrect truncation, whereas an excessive number of sliding windows will pose difficulties
for the grid in fulfilling the truncation conditions. Furthermore, the design of the truncation threshold is equally crucial. An overly small threshold will make it arduous for the grid to meet the truncation conditions. In contrast, a threshold that is excessively large will lead to incorrect truncation and impact the accuracy upon convergence.

**Figure 5.** Illustrations of truncation methods: (a) convolution kernel truncation; (b) sliding window-based probability truncation.

### 4. Experiments

#### 4.1. Experimental Setting

The experimental platform consists of two main components: a data acquisition platform and an algorithm-processing platform, as illustrated in Figure 6. The data acquisition platform includes a UAV flight platform and two photoelectric stabilization gimbal platforms. The UAV flight platform employs the KWT-X4L industrial UAV platform, which is equipped with a GNSS module and an attitude and heading reference system (AHRS) for recording positioning and orientation system (POS) information, thereby providing accurate ground-truth locations. We mounted two photoelectric stabilization gimbal platforms on the UAV flight platform. The two photoelectric stabilization gimbal platforms are configured to capture images from downward- and forward-facing orientations, respectively. Additionally, the downward-facing gimbal is outfitted with a 1D laser range finder to measure the aboveground level (AGL). The algorithm-processing platform utilizes a ThinkPad Neo14 laptop equipped with an i5-12500H CPU and 16 GB of RAM. The software was developed in C++ and operated on the Windows 11 operating system using Visual Studio with the OpenCV library. This platform establishes a network connection with the data acquisition platform and performs algorithm validation by processing images, videos, and POS information collected offline. During the algorithm calculation process, the images captured by the forward-facing camera and the downward-facing camera are utilized to compute the observation likelihood of two distinct viewpoints. The videos recorded by the downward-facing camera, in combination with the laser-ranging information collected by the downward-facing gimbal, are jointly utilized to acquire the odometry information. Within the localization framework, aside from the main thread of PMF, we devised three sub-threads to calculate the forward-facing camera observation, downward-facing camera observation, and odometry information, respectively:

- **Forward-facing camera observation thread:** Acquire the images captured by the forward-facing camera at time \( t - 1 \) and time \( t \) and generate descriptors and compute the observation likelihood \( p(z_t | \chi_t) \) via the approaches presented in Sections 3.2 and 3.3 of this article;
• Downward-facing camera observation thread: Acquire the image captured by the downward-facing camera at time $t$ and compute the observation likelihood $p(z_t^d | \chi_t)$ based on a prevalent image-matching methodology;

• Odometry thread: Calculate the odometry information $u_{t-1}$ between time $t - 1$ and time $t$ based on a prevalent visual odometry methodology.

**Figure 6.** (a) Data acquisition platform; (b) algorithm-processing platform.

At first, a 7.7 km × 8 km area in the suburbs of Jinan City, China, was selected. The MAPWORLD GIS satellite map and DEM map of the area were acquired through offline download on the ground station system. Subsequently, a 10 km flight path featuring 75 waypoints was planned on the map. Each waypoint incorporates GPS longitude and latitude as well as flight altitude and also includes control information such as flight speed, UAV yaw, and gimbal attitude. Once the waypoints were set up, we controlled the UAV to take off and ensured that it automatically executed the flight path. At each waypoint, the flight control system automatically commands the two gimbals to capture images and POS information; furthermore, the downward gimbal consistently records videos and laser distance data during the flight, and the videos and data are utilized as samples for the calculation of visual odometry. Eventually, upon the accomplishment of the flight mission, we obtained the images, videos, and POS via the ground station system and verified our algorithm on the algorithm-processing platform. To guarantee the field of view coverage of the downward camera, the altitude of our flight route is fixed at 620 m above sea level (approximately 200–400 m relative to the ground). In order to ensure a sufficient field of view coverage of the downward-facing camera, the waypoints were planned at an altitude of 620 m above sea level (approximately 200–400 m relative to ground level). To comprehensively validate the performance of the algorithm in diverse terrain types and surface features, the flight path covers diverse terrain types, including plains, undulating hills, and mountains, and includes a variety of surface features. Figure 7a presents the screenshot of waypoint planning on the ground station system. Figure 7b,c display specific images captured during the flight.
were made by integrating barometric altitude and laser distance measurements. These windows was set to three.

level (approximately 200–400 m relative to the ground). In order to ensure a su pport of 620 m above sea level (approximately 200–400 m relative to ground level). To com-prehensively validate the performance of the algorithm in diverse terrain types and sur-vice coverage of the downward-facing camera, the waypoints were planned at an altit-

duation mode type, the use of a truncation operation, and the incorporation of observation
data from the forward-facing camera during the takeoff and ascent phases, as detailed in Table 1. Each method employed the same PMF framework for backend processes with \( \delta = 20 \text{ m} \), \( N = 1,440,000 \). The PMF state prediction utilizes the ORB-SLAM2 method [34] to construct odometry, which is then scale-corrected using a 1D laser range finder [35].

Given the odometry’s accumulated error characteristics, we set the per-meter travel error at 0.1 m (\( \frac{\sigma_{\text{odm}}}{N} = 0.1 \)), which informed the calculation of the convolution kernel truncation radius. The probability truncation threshold \( \varepsilon \) was set at \( \frac{91}{N} \), and the number of sliding windows was set to three.

During observations with the downward-facing camera, image scaling adjustments were made by integrating barometric altitude and laser distance measurements. These adjusted images were then matched to a 2D satellite map using the NCC method [17]. For observations using the forward-facing camera, a descriptor size of 2 km \( \times \) 2 km was used. Feature extraction and matching were conducted on keyframes from two adjacent frames using the SIFT method, followed by 3D reconstruction (Figure 8 displays the results of image matching and 3D reconstruction for adjacent keyframes). Subsequently, the 3D points were mapped onto the descriptor and matched to the DEM.

We emphasize that, during the downward-facing camera observation process, considering the changes in environmental lighting, surface features, and terrain elevation, it is adequate for conducting an observational comparative experiment, although the NCC method may be suboptimal. The noise parameters were determined based on the data characteristics of each sensor and the elevation data. The yaw angle noise \( \sigma_{\text{yaw}} \) was configured as 3°, barometer altitude noise \( \sigma_{\text{baro}} \) was configured as 15 m, gimbal pitch angle noise \( \sigma_{\text{pitch}} \) was configured as 0.5°, and DEM elevation noise \( \sigma_{\text{map}} \) was configured as 20 m.

Figure 7. (a) Screenshot of waypoint planning on ground station system; (b) sample of aerial images captured by a downward-facing camera; (c) sample of aerial images captured by a forward-facing camera.

4.2. Experimental Procedure

To rigorously validate the improvements in convergence ratio, robustness, localization accuracy, and computational efficiency offered by the proposed method, we initially established a comparative framework. This framework involved six different comparative methods, each operating under varied conditions based on three variables: the observation mode type, the use of a truncation operation, and the incorporation of observation data from the forward-facing camera during the takeoff and ascent phases, as detailed in Table 1. Each method employed the same PMF framework for backend processes with \( \delta = 20 \text{ m} \), \( N = 1,440,000 \). The PMF state prediction utilizes the ORB-SLAM2 method [34] to construct odometry, which is then scale-corrected using a 1D laser range finder [35].

Given the odometry’s accumulated error characteristics, we set the per-meter travel error at 0.1 m (\( \frac{\sigma_{\text{odm}}}{N} = 0.1 \)), which informed the calculation of the convolution kernel truncation radius. The probability truncation threshold \( \varepsilon \) was set at \( \frac{91}{N} \), and the number of sliding windows was set to three.

During observations with the downward-facing camera, image scaling adjustments were made by integrating barometric altitude and laser distance measurements. These adjusted images were then matched to a 2D satellite map using the NCC method [17]. For observations using the forward-facing camera, a descriptor size of 2 km \( \times \) 2 km was used. Feature extraction and matching were conducted on keyframes from two adjacent frames using the SIFT method, followed by 3D reconstruction (Figure 8 displays the results of image matching and 3D reconstruction for adjacent keyframes). Subsequently, the 3D points were mapped onto the descriptor and matched to the DEM.

We emphasize that, during the downward-facing camera observation process, considering the changes in environmental lighting, surface features, and terrain elevation, it is adequate for conducting an observational comparative experiment, although the NCC method may be suboptimal. The noise parameters were determined based on the data characteristics of each sensor and the elevation data. The yaw angle noise \( \sigma_{\text{yaw}} \) was configured as 3°, barometer altitude noise \( \sigma_{\text{baro}} \) was configured as 15 m, gimbal pitch angle noise \( \sigma_{\text{pitch}} \) was configured as 0.5°, and DEM elevation noise \( \sigma_{\text{map}} \) was configured as 20 m.
The flight path can be segmented into four stages based on prediction step in the PMF was conducted to evaluate the performance enhancement of localization to some extent. Additionally, an average runtime analysis of the state prediction step in the PMF was conducted to evaluate the performance enhancement of the convolution kernel truncation method. The results are presented in Table 1. A more intuitive visual representation of the entire localization process is provided through charts and map trajectories, as depicted in Figures 9–11.
Figure 9. Visualization of the location trajectory, with 0 denoting the start-point and 74 representing the end-point in each graph. The actual flight path and the results of each experiment are represented by green and red curves, respectively. Figures (a–f) depict the localization trajectories for SAMPLE-00 to SAMPLE-05.

Figure 10. Statistical graphs of localization error.
5. Results and Discussion

Figure 9 illustrates the actual flight path and the results of each experiment, represented by green and red curves, respectively. The analysis and discussion are presented as follows.

5.1. Influence of Type of Observation Mode

It can be discerned from Table 1 that the second observation mode (SAMPLE-02 to SAMPLE-05) demonstrated a higher convergence rate in localization and a smaller average localization error after convergence compared to the first observation mode (SAMPLE-00 to SAMPLE-01). This suggests that the second mode offers greater precision and robustness in localization. Without the truncation operation, the number of iterations required for convergence in SAMPLE-02 and SAMPLE-03 decreased from 67 to 19, in comparison to SAMPLE-00, while for ALE, it decreased from 165.4 m to 37.6 m and 37.4 m, respectively. With the truncation operation, the number of iterations for convergence in SAMPLE-04 and SAMPLE-05 decreased to 14 and 12, respectively, compared to 30 in SAMPLE-01, while for ALE, it decreased to 34.4 m and 34.4 m, respectively, compared to 55.4 m in SAMPLE-01. Additionally, as depicted in Figure 10, during the fourth stage of flight, the localization

<table>
<thead>
<tr>
<th>Experiment Index</th>
<th>Variable-1</th>
<th>Variable-2</th>
<th>Variable-3</th>
<th>ITE</th>
<th>ALE (m)</th>
<th>ALStd (m)</th>
<th>RT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE-00</td>
<td>Type 1</td>
<td>No</td>
<td>–</td>
<td>67</td>
<td>165.4</td>
<td>321.9</td>
<td>13.2</td>
</tr>
<tr>
<td>SAMPLE-01</td>
<td>Type 1</td>
<td>Yes</td>
<td>–</td>
<td>30</td>
<td>55.4</td>
<td>90.7</td>
<td>0.078</td>
</tr>
<tr>
<td>SAMPLE-02</td>
<td>Type 2</td>
<td>No</td>
<td>No</td>
<td>19</td>
<td>37.6</td>
<td>107.2</td>
<td>13.1</td>
</tr>
<tr>
<td>SAMPLE-03</td>
<td>Type 2</td>
<td>No</td>
<td>Yes</td>
<td>19</td>
<td>37.4</td>
<td>105.4</td>
<td>13.5</td>
</tr>
<tr>
<td>SAMPLE-04</td>
<td>Type 2</td>
<td>Yes</td>
<td>No</td>
<td>14</td>
<td>34.4</td>
<td>77.9</td>
<td>0.072</td>
</tr>
<tr>
<td>SAMPLE-05</td>
<td>Type 2</td>
<td>Yes</td>
<td>Yes</td>
<td>12</td>
<td>34.4</td>
<td>79.9</td>
<td>0.069</td>
</tr>
</tbody>
</table>
error for the first observation mode significantly increases due to terrain undulations and scene ambiguity. In contrast, the localization error for the second observation mode remains relatively small and stable. This stability is attributed to the enhanced constraints on the posterior probability distribution provided by observations from the forward-facing camera, thereby improving the accuracy and robustness of the localization.

5.2. Influence of Sliding Window Probability Truncation

The inclusion of sliding window probability truncation accelerates the convergence rate of localization across both observation modes. It can be discerned from Table 1 that for the first observation mode, the number of iterations required for convergence decreased from 67 in SAMPLE-00 to 30 in SAMPLE-01. For the second observation mode, the number of iterations decreased from 19 in SAMPLE-02 to 14 in SAMPLE-04 and from 19 to 12 when comparing SAMPLE-03 and SAMPLE-05. Fundamentally, this is attributed to the truncation of the low-confidence grids, which leads to a more concentrated probability distribution and thereby expedites the convergence of the covariance. Additionally, based on the ALE presented in Table 1, it was discovered that through the application of the sliding window truncation approach, the localization error upon convergence was decreased to a certain extent. This is owing to the fact that we chose the suitable number of sliding windows and the rational truncation threshold, thereby effectively guaranteeing that the influence brought by low-confidence grids can be precisely and promptly eliminated.

5.3. Influence of the Utilization of Observation from the Forward-Facing Camera during the Takeoff and Ascent Stages

A comparative analysis between SAMPLE-04 and SAMPLE-05 indicates that utilizing observations from the forward-facing camera during the takeoff and ascent stages reduces the number of iterations required for convergence from 14 to 12. Furthermore, a comparison between SAMPLE-02 and SAMPLE-03 shows that although the number of iterations for convergence remains the same, there is a notable decrease in localization error and standard deviation during the initial iterations (from iteration index 10 to 18), as illustrated in Figures 10 and 11. To quantify more intuitively, statistics on the average localization error and the average localization standard deviation of the first nine iterations prior to convergence have been conducted, as shown in Table 2. For SAMPLE-04 and SAMPLE-05, under the circumstance of the same number of iterations, the average localization error and the average localization standard deviation of SAMPLE-05 are significantly smaller. Compared with SAMPLE-04, the average localization error of SAMPLE-05 decreased by approximately 18.9%, and the average localization standard deviation decreased by approximately 16.8%. When comparing SAMPLE-03 with SAMPLE-02, the average localization error and the average localization standard deviation of SAMPLE-03 are also significantly smaller. In comparison with SAMPLE-02, the average localization error of SAMPLE-03 decreased by approximately 14.4%, and the average localization standard deviation declined by approximately 5.4%. This improvement suggests enhanced localization performance. Therefore, employing observational data during the takeoff and ascent stages can potentially expedite the convergence rate and improve the accuracy and robustness of localization.

Table 2. Average localization error and average localization standard deviation statistics throughout several iterative processes prior to convergence (SAMPLE-02 to SAMPLE-05).

<table>
<thead>
<tr>
<th>Experiment Index</th>
<th>Average Localization Error (m) (9 Iterations Prior to Convergence)</th>
<th>Average Localization Standard Deviation (m) (9 Iterations Prior to Convergence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE-02</td>
<td>771.0</td>
<td>1336.6</td>
</tr>
<tr>
<td>SAMPLE-03</td>
<td>660.2</td>
<td>1263.9</td>
</tr>
<tr>
<td>SAMPLE-04</td>
<td>1139.4</td>
<td>1480.3</td>
</tr>
<tr>
<td>SAMPLE-05</td>
<td>924.0</td>
<td>1231.8</td>
</tr>
</tbody>
</table>
5.4. Influence of Truncating the Convolution Kernel

Truncating the convolution kernel during the prediction step of the PMF results in a significant reduction in runtime. As indicated by the runtime statistics in Table 1, runtime decreases from approximately 13.1 to 13.5 s to between 0.069 and 0.078 s; that is, the runtime has been reduced by 99.5%. In the experiments we conducted, we determined the truncation size based on the standard deviation of the odometer noise and the scale. It is notable that, considering the displacement of the odometer at two adjacent instants, the size of the convolution kernel is variable, and its average size is approximately $7 \times 7$. This size is considerably lower than the number of grids of PMF, and the theoretical reduction rate of the actual runtime should be 99.9%. Nevertheless, we have observed that there exists a disparity between the reduction rate of the actual runtime and the theoretical value. Currently, in-depth exploration has yet to be conducted on this phenomenon. It is speculated that the reason lies in the differences in the underlying optimization strategies for loop computing of the CPU. Furthermore, the improvement in computational efficiency brought by truncating the convolution kernel is pronounced.

5.5. Consideration of the Noise Model

A concise discussion is conducted regarding the mathematical model of the observation noise. The Gaussian model is typically employed to model the noise of sensors and states. Its mathematical expression is distinct and intuitive, and the calculation is straightforward. The noise of the gimbal’s attitude originates from the white noise and bias of the IMU sensor, among which the white noise is approximately Gaussian distributed; furthermore, within the state estimator of the gimbal, the noise of each state is also modeled as Gaussian distribution. Hence, it is precise and rational to depict the attitude noise using the Gaussian model. However, concerning the observation noises of odometry, barometer, and DEM elevation information, although they approximately exhibit the characteristics of Gaussian distribution in numerous cases, under certain circumstances (e.g., barometers encounter unstable airflow), the magnitude of noise will increase to a certain extent, and the noise exhibits more complex distribution characteristics. If the conventional Gaussian model is employed at this point, it will lead to modeling errors. Even considering these circumstances, in this paper, the Gaussian model is still used to model the observation noise. During the experimental process, we accommodate the circumstances where the noise is more pronounced, and the noise distribution is more complex by setting a larger Gaussian kernel radius. Enlarging the radius of the Gaussian kernel is a strategy from an engineering perspective. It not only retains the intuitive and computationally convenient advantages of the Gaussian model but also enhances adaptability in adverse situations.

5.6. Summary of Discussion

Based on the analysis and discussion above, we propose that for a general PMF-based localization framework, integrating the forward-facing camera observation model can significantly enhance the convergence rate, accuracy, and robustness of localization during the cruise flight stage. Additionally, this integration may lead to improvements in localization during the takeoff and ascent stages. The test scenarios, which included diverse terrains and surface features, demonstrated the effectiveness of the proposed method comprehensively. Moreover, the proposed methods of convolution kernel truncation and probabilistic truncation can reduce runtime and accelerate the convergence rate of localization, respectively.

6. Conclusions

This study introduces a novel approach designed to accelerate the convergence rate of localization, specifically targeting the challenge of global localization in environments where GNSS signals are unavailable. The proposed method consists of two main components: the development of an observation model utilizing a forward-facing camera and the implementation of two truncation methods. Experimental results conducted on a
7.7 km × 8 km map illustrate that incorporating the observation model from the proposed forward-facing camera into a PMF framework significantly enhances the rate of localization convergence, as well as improves accuracy and robustness compared to methods that rely solely on observations from downward-facing cameras. Additionally, from an engineering standpoint, the two truncation methods introduced in this study contribute to improved computational efficiency and further accelerate the localization convergence rate.

The observation model presented in this research demonstrates considerable extensibility and can be theoretically applied to any UAV localization framework that employs Bayesian filtering. Furthermore, the two truncation methods are merely an initial attempt to enhance the performance of the PMF framework that contains only two states. For the PMF framework containing more states, we believe that it will bring more significant performance improvement. This method holds substantial potential for practical applications, especially for UAV localization in mountainous regions, effectively addressing the limitations of localization techniques that depend exclusively on downward-facing cameras. Given the limited FoV of a forward-facing camera, there remains considerable scope for further performance enhancements. Future research will focus on developing an omnidirectional observation model using fisheye cameras to further enhance the localization performance.

**Author Contributions:** Conceptualization, Z.L. (Zhenyu Li); methodology, Z.L. (Zhenyu Li) and X.J.; software, Z.L. (Zhenyu Li), H.D. and H.J.; validation, Z.L. (Zhenyu Li) and Z.S.; formal analysis, X.J. and X.M.; investigation, X.J.; resources, S.M. and Z.L. (Zhenyi Lv); data curation, X.M.; writing—original draft preparation, Z.L. (Zhenyu Li); writing—review and editing, X.J.; visualization, H.J.; supervision, X.J.; project administration, X.M.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been funded by the Key Research and Development Program of Shandong Province (2021CXGC011304) and the Industry, University and Research Innovation Fund of China University (2021ZYA1204).

**Data Availability Statement:** Data are available on reasonable request from the authors.

**Acknowledgments:** The authors would like to express their appreciation to Shandong University for the financial support and guidance in theoretical research, as well as to Shenzhen Keweitai Company for providing the UAV platform and offering technical assistance.

**Conflicts of Interest:** Authors Zhenyu Li, Zhenyi Lv, Hongliang Ding, Haiyan Ji, Zheng Sun were employed by the company Shenzhen Keweitai Enterprise Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**References**

1. Holland, R.A. The role of visual landmarks in the avian familiar area map. *J. Exp. Biol.* 2003, 206, 1773–1778. [CrossRef] [PubMed]


20. Cao, Y.; Ren, K.; Chen, Q. Template matching based on convolution neural network for UAV visual localization. *Optik* 2023, 283, 170920. [CrossRef]


29. Thrun, S. Probabilistic robotics. *Commun. ACM* 2002, 45, 52–57. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.