

Article

RTAPM: A Robust Top-View Absolute Positioning Method with Visual–Inertial Assisted Joint Optimization

Pengfei Tong ¹, Xuerong Yang ^{1,*}, Xuanzhi Peng ² and Longfei Wang ^{1,3}

¹ The School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen 518106, China; tongpf3@mail2.sysu.edu.cn (P.T.); wanglf37@mail2.sysu.edu.cn (L.W.)

² The School of College of Engineering, Shantou University, Shantou 515063, China; 22jzpeng@stu.edu.cn

³ JiangHuai Advanced Technology Center, Hefei 230000, China

* Correspondence: yangxr23@mail.sysu.edu.cn

Abstract: In challenging environments such as disaster aid or forest rescue, unmanned aerial vehicles (UAVs) have been hampered by inconsistent or even denied global navigation satellite system (GNSS) signals, resulting in UAVs becoming incapable of operating normally. Currently, there is no unmanned aerial vehicle (UAV) positioning method that is capable of substituting or temporarily replacing GNSS positioning. This study proposes a reliable UAV top-down absolute positioning method (RTAPM) based on a monocular RGB camera that employs joint optimization and visual–inertial assistance. The proposed method employs a bird’s-eye view monocular RGB camera to estimate the UAV’s moving position. By comparing real-time aerial images with pre-existing satellite images of the flight area, utilizing components such as template geo-registration, UAV motion constraints, point–line image matching, and joint state estimation, a method is provided to substitute satellites and obtain short-term absolute positioning information of UAVs in challenging and dynamic environments. Based on two open-source datasets and real-time flight experimental tests, the method proposed in this study has significant advantages in positioning accuracy and system robustness over existing typical UAV absolute positioning methods, and it can temporarily replace GNSS for application in challenging environments such as disaster aid or forest rescue.



Academic Editors: Hector Gutierrez and Yang Liu

Received: 2 December 2024

Revised: 2 January 2025

Accepted: 4 January 2025

Published: 7 January 2025

Citation: Tong, P.; Yang, X.; Peng, X.; Wang, L. RTAPM: A Robust Top-View Absolute Positioning Method with Visual–Inertial Assisted Joint Optimization. *Drones* **2025**, *9*, 37. <https://doi.org/10.3390/drones9010037>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: point–line image matching; motion constraints; joint state estimation; satellite signals denied; absolute positioning

1. Introduction

In recent years, vision and vision-assisted localization [1,2] have emerged as the most essential solutions for the replacement or supplementation of global navigation satellite system (GNSS) positioning. Relative vision localization [3] and absolute vision localization [2] are the two primary methods for localizing unmanned aerial vehicles (UAVs) based on vision. Frame-to-frame positioning is another term for the relative vision positioning technique, while frame-to-reference position is another term for the absolute vision positioning method. The primary obstacles to vision-based localization methods are the complexity of comprehending and discriminating these data, as well as the analysis of big data in real time.

The primary challenge of relative vision positioning [2] is the resolution of error accumulation, specifically time drift. Drift is the cumulative error that occurs when recursive estimation is used to produce new estimates. The accuracy of the current estimation will be influenced by the error in the previous estimate if the current estimate depends on it.

Relevant researchers have conducted a significant amount of research on the fundamental characteristics of cumulative error in the context of relative visual positioning. For instance, visual–inertial odometry [4–6] closely integrates with inertial data. This method can address UAVs' short-term positioning demands in limited scenes. However, these methods are unable to address the fundamental issue of error accumulation, necessitating further targeted research. We employed absolute vision positioning as an approach to eliminate drift. In order to complete the UAV self-positioning, the absolute vision positioning method typically relies on a pre-collected dataset, known as reference data, which undergoes accurate geo-reference and correction to compare its similarity to the current frame. The reference data may consist of loosely organized satellite image sets or merged satellite images, most of which have undergone orthorectification [7]. Nowadays, the increasing availability of free maps (such as Google Earth™) and geographic information systems (GIS) (ArcGIS™) facilitates the rapid advancement of absolute positioning methods. The pre-flight UAV image dataset is yet another method of acquiring reference data. When acquiring images, it is necessary to record the airborne GNSS position information [8–11]. The absolute visual positioning method relies on the reliability of GNSS during data collection, whereas real-time positioning does not require GNSS.

The main contributions of the paper can be summarized as four parts:

- (1) We proposed an automatic geo-referencing algorithm for remote sensing imagery. Remote sensing images collected by Google Maps were written into geographical information employing automatic registration procedures, completing the automatic production of template images.
- (2) A fast point–line matching approach was proposed. We estimated the UAV velocity in real time and employed motion constraints to minimize the map's matching range. FAST methods were employed to extract anchor points on the line segments generated by the EDlines approach, and a fast map matching and search method suitable for large-scale outdoor environments was established by means of a limited amount of anchor points. This method enhances the usability of map matching on platforms with limited computer capabilities.
- (3) A global state joint estimation approach was proposed. This study combined visual–inertial odometry with error accumulation and map matching absolute positioning, and it employed the global state joint estimate approach to substantially minimize the error accumulation of the relative visual positioning method in the long-term positioning process.
- (4) An experimental platform was constructed, and a bird's-eye view absolute positioning solution was developed. The flight control served as an intermediate node, while the onboard camera was employed to capture aerial views in real time. The UAV's onboard CPU processed any positioning issues in real time and sent the results to the ground station in real time via the Mavros communication of the data transmission module to be employed by other application modules.

This paper is divided into the following sections. First, an overview of visual absolute positioning based on UAV is introduced in Section 2. Then, Section 3 explains the principle of RTAPM, which is a robust top-view absolute positioning method. Afterwards, the design and implementation of the experimental system are described in Section 4, and Section 5 describes the experiment and results. Finally, the experimental conclusions and future research work are discussed in Section 6.

2. Absolute Localization: An Overview

Vision-based localization and navigation methods have been the subject of extensive theoretical and experimental investigation by numerous academics [2]. Three primary

subjects of exploration include visual odometry, feature matching-based UAV location technology, and template matching-based UAV localization. As shown in Figure 1, the absolute visual positioning and navigation process of a UAV based on prior satellite images is described.

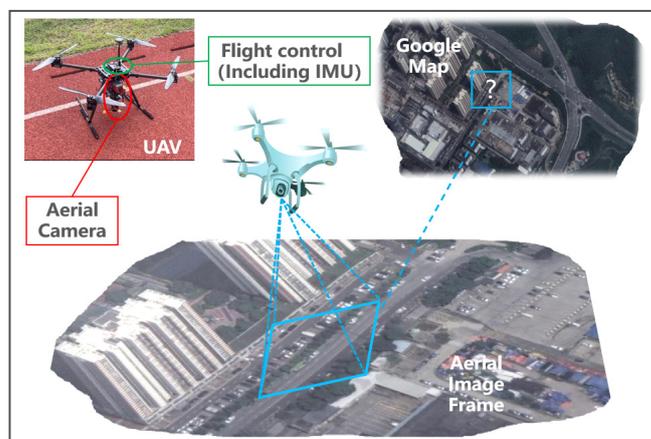


Figure 1. Schematic diagram of the absolute visual positioning and navigation of UAV based on prior satellite images [2].

2.1. UAV Localization Based on Template Matching

In the field of image matching and image registration, template matching is often referred to as direct or intensive matching. Many researchers utilized template matching to suggest various approaches in the research on absolute visual positioning. The positioning process has been described as a template matching problem, and it can be performed by searching for the template in the reference map based on the UAV's current view. The template matching-based UAV locating approach compares two image patches and establishes their similarity measurements, employing image patch comparison operators such as the sum of square differences. The high computational cost for calculating similarity is the primary disadvantage of template matching.

Dalen et al. [12] proposed using normalized cross-correlation to estimate the UAV's absolute position. Dense images between UAV images and Bing Maps™ reference images are aligned using the normalized cross-correlation that is added to the probability density function in the particle filter architecture. The aim of this work was to add an absolute location estimate to the simultaneous localization and mapping (SLAM) navigation system [13,14]. The estimated location and variance are incorporated into the extended Kalman filter (EKF) measurement update of the SLAM navigation system when the variance threshold and continuous estimation are completely close. An information theoretic measure of the interdependency between two signals is known as mutual information. For the purpose of UAV location, mutual information is employed to quantify how much information two images share [15,16]. When analyzing local and global differences between images, mutual information is more accurate than sum of squares and normalized cross-correlation, but it is computationally more challenging. Wan et al. [17] proposed a positioning approach based on illumination invariant phase correlation [18]. Phase correlation is a template matching method based on Fourier translation features. Research and demonstration have shown that the phase correlation process is fundamentally indifferent to light fluctuations brought about by the sun's position. Patel [19] presented a positioning strategy based on Yol's work [20]. In [21], a location approach was proposed based on the normalized information distance estimated using mutual information similarity measurements. Unlike mutual information, normalized information distance remains unaffected by image

overlap. As a result, normalized information distance is more commonly utilized for geographical location.

2.2. UAV Localization Based on Feature Matching

Feature matching, also known as indirect matching, has the potential to effectively substitute template matching in UAV visual positioning applications. Feature matching and location include two phases, which are feature point detection and descriptor extraction. The angle detectors, such as the well-known Harris and FAST [22] detectors, are commonly employed for feature point detection. The aim of feature point detection is finding salient spots that are readily discernible in two completely distinct detection rounds on separate photos in the same area. The images utilized for feature point detection may differ substantially in illumination, proportion, rotation, and viewpoint. Descriptor extraction is the process of extracting feature vectors from the area around feature points. The gradient histogram based on SIFT features [23] and the binary test based on Brief features [24] represent two common feature matching approaches. The goal of feature matching is to generate a descriptor that employs metrics like Euclidean distance or Markov distance to match several feature points.

The works of Seema et al. [25] and Saranya et al. [26] are comparable. UAV images are registered to the global reference map by comparing normalized cross-correlation features with a combination of random sample consensus and SURF. Shan et al. [27] provided a positioning system that incorporated directional gradient histograms, particle filtering, and optical flow. The reference data for this method are obtained by applying the directional gradient histogram to all feasible points of the global reference map. Chiu et al. [28] proposed a system for positioning and navigation in a global positioning system (GPS) denial environment that combines inertial measurement unit (IMU) and geographic image registration data. Mantelli et al. [29] designed a 4-DOF absolute positioning system with satellite imagery. The system employs a down-looking monocular RGB camera, assuming that the roll and pitch angles are near zero, and matches the UAV image to the satellite map employing Brief's abBRIEF descriptor. Masselli et al. [30] proposed a strategy combining terrain classification and particle filtering. This approach classifies terrain patches into four arbitrary categories: grass, bushes, roads, and buildings. This method just employs the visual method to estimate position and does not involve additional IMU data. The experiment was carried out in an outside environment of 100 m, and the average position estimation error was 9.5 m.

2.3. Visual Odometry

Visual odometry is a positioning approach that compares the current frame observed by the UAV to previous frames to determine the difference in self-motion. In general, the pose estimation problem is solved using operational flow analysis [31]. Visual odometry adds the estimated differential pose vector to the prior pose estimation to obtain a new pose estimate. As a result, visual odometry can be used to estimate position only based on current and historical observations.

Visual odometry is not limited to the relative visual positioning method but can additionally be employed in an absolute visual positioning system developed with pre-flight data. In this technique, visual odometry is employed to collect data registered by geo-graphical location and establish a database that is valuable for future relocation. The absolute positioning approach of visual odometry is distinct from the other methods. Except for Goforth and Lucey [32], current approaches lack the use of continuous frame comparison. In existing visual odometry methods [33], the UAV image must be compared with the pose graph, not merely the image features, so it faces completely distinct challenges,

promoting the development of the absolute visual positioning method of visual odometry. The research of Goforth and Lucey appears to be more equivalent to the depth learning method. Warren et al. [11] presented a positioning strategy. After GPS positioning failed, the UAV could return to its initial area. The authors updated the VT&R path tracking algorithm for the operation of unmanned vehicles. Once the UAV obtained stable and continuous GPS location, it employed visual odometry and GPS navigation to generate a relative pose graph. The visual odometry system employed SURF features to minimize pose outliers and the maximum likelihood estimation sample consensus to obtain the transformation relative to the last key frame.

In summary, when the satellite positioning system refuses, existing approaches mostly employ imagery and the airborne reference image to match the scene in order to acquire the UAV's absolute position. However, as the reference image and the information collected by the UAV in real time are different in height, time, and perspective, the corresponding relationship of features in the image is destroyed, which makes it difficult to complete the accurate positioning of the UAV in the real complex environment. As a result, a highly reliable matching system must be explored with the goal of guaranteeing correct UAV positioning in a real-world challenging environment.

3. RTAPM: A Robust Top-View Absolute Positioning Method

The matching positioning based on satellite maps attempts to address the problem of being unable to obtain a reliable absolute position due to noisy GNSS signals or being denied. In a complex challenge environment, this approach can replace the GNSS system to obtain short-term absolute geographical location data for UAVs when GNSS signals are unavailable or positioning has a significant divergence. As shown in Figure 2, the UAV absolute positioning method based on the existing image only involves the use of an airborne down-looking monocular RGB camera to estimate the flight position of the UAV through image projection matching between the real-time aerial image and the existing satellite image or mapping the base map in the flight area during flight, resulting in UAV positioning and map measurement updates. The absolute positioning technology of UAVs based on mapping base maps does not rely on GNSS, which can significantly improve the autonomous combat capabilities of UAVs in a conflict environment.

The RTAPM method proposed in this paper is an absolute positioning technology for UAVs that relies on surveying and mapping base maps. It locates and extracts significant points and line features from up-to-date surveying and mapping base maps, generates a key feature library, develops observation models employing real UAV collected data, and achieves absolute positioning by matching the observation model to the feature library. At the same time, the framework integrates a real-time visual-inertial odometry system and employs the visual optical flow approach to enhance the system's positioning efficiency. In contrast, the motion constraint strategy is employed in the map matching stage to minimize the range to be matched as much as feasible while maintaining accuracy, and the least point and line feature matching is employed for efficient matching. Finally, joint state filtering is utilized to attain high-precision fusion of the results of map matching and visual-inertial odometry while taking consistency into account, resulting in the UAV's real-time geographical location.

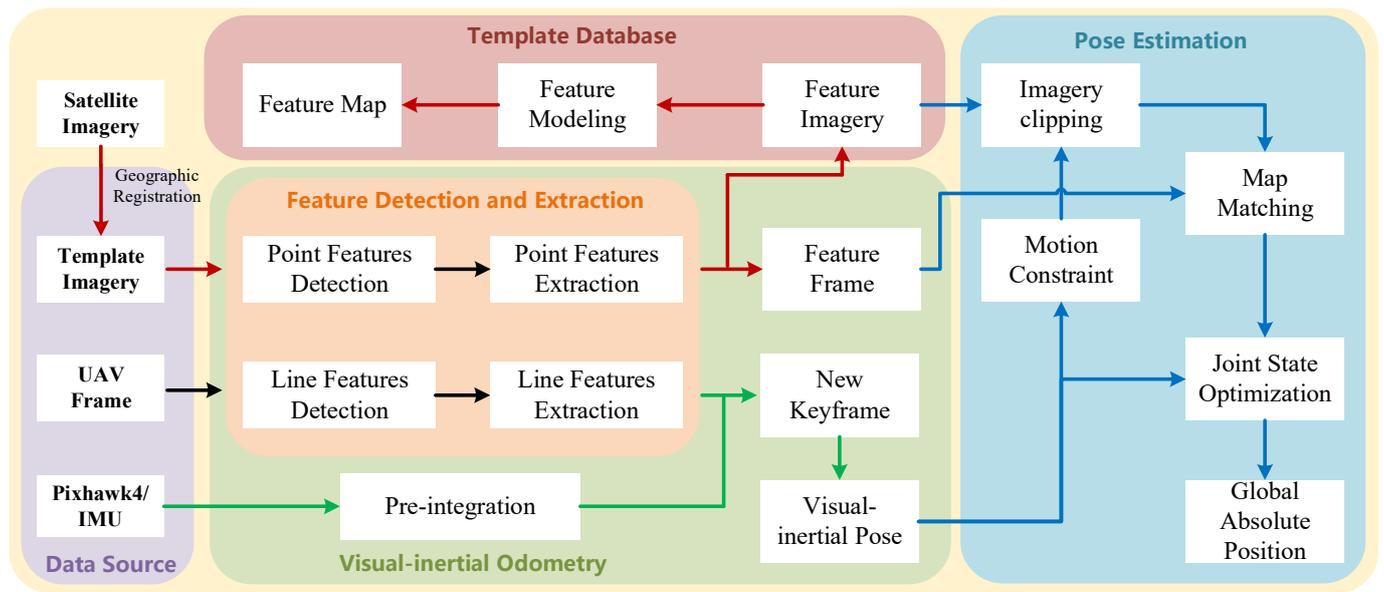


Figure 2. The framework of RTAPM. The green arrows represent the data flow of visual-inertial odometry; the red arrows represent map matching; and the blue arrows represent pose evaluation. The black arrows represent the common data flow for visual-inertial odometry and map matching.

3.1. Template Geographic Registration

The geographic registration template image is based on a Google Earth survey area image. The imagery matching method is used to write a variety of coordinate information, comprising geographical reference, so that the matching process can be completed with the goal of responding immediately to real-time location during matching and positioning.

Algorithm 1 illustrates the template geographic registration method. First, select a download source based on the survey area's known longitude and latitude coordinates, image zoom level, and image style, and then collect the Google image tiles that require zoom level from the Google image. The downloaded image tiles suffer from ontological coordinates, which are required to be unified into the same coordinate system. The image tiles are then fused together. The desired coordinate scheme is chosen via a series of appropriate coordinate transformations, and the world coordinates of the four corners of the template image are filled in, which enables the matching and positioning to be quickly retrieved based on the pixel values when the matching is completed. Finally, the image is saved as the chosen geographic image.

Tiles primarily provide open-source images. For collecting tiles in an area with accurate longitude and latitude, as well as the longitude and latitude corresponding to pixel locations on tiles, longitude and latitude coordinates typically have to be converted to tile coordinates and pixel coordinates, respectively. The fundamental differences between tile coordinate systems are the distinct earth ranges intercepted by projection and the different tile coordinate starting points. For the world map projected into a plane by Mercator across different map resolutions, the world map is divided into pixels as map units, where each map unit is referred to as a map tile. Figure 3 demonstrates the specific conversion.

Algorithm 1 Template Geographic Registration

Input: Template boundary longitude $long_b$, latitude lat_b and zoom level l .

Output: Registered template $T_{long,lat}$.

- 1: **if** $long_b$ and lat_b are legal **then**
 - 2: Obtain template boundary tiles coordinates (x_b^{tile}, y_b^{tile}) from $(long_b, lat_b)$ by Equation (10).
 - 3: **if** l exists in (x_b^{tile}, y_b^{tile}) **then**
 - 4: Obtain download source *urls*.
 - 5: Obtain tile image total numbers n .
 - 6: $num = 0$
 - 7: **for** tile i in n **do**
 - 8: $num = num + 1$
 - 9: **if** $num \geq 2$ **then**
 - 10: Merge tile images from 1 to num .
 - 11: Obtain merged tiles corners pixel coordinates $(x_{corner}^{pixel}, y_{corner}^{pixel})$ by Equation (11).
 - 12: Obtain merged tiles corners longitude and latitude $(long_{corner}, lat_{corner})$ by Equation (3).
 - 13: Geographic information writing image.
 - 14: $num = num - 1$
 - 15: **end if**
 - 16: **end for**
 - 17: Save template image.
 - 18: **else**
 - 19: Decrease the zoom level l .
 - 20: **end if**
 - 21: **end if**
 - 22: Obtain registered template $T_{long,lat}$.
-

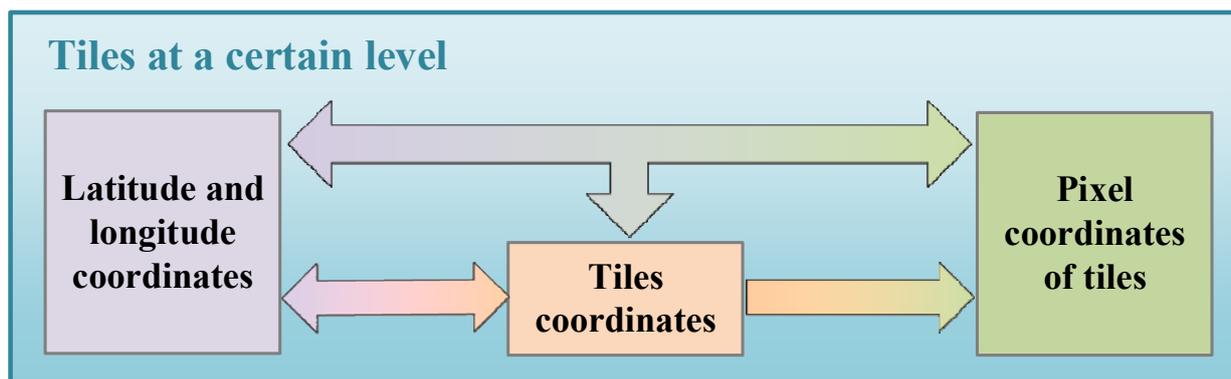


Figure 3. Coordinate transformation in the template geographic registration.

Latitude and longitude coordinates $(long_b, lat_b)$ to tile coordinates (x_b^{tile}, y_b^{tile}) ,

$$\begin{cases} x_b^{tile} = \frac{long_b + 180}{360} * 2^l \\ y_b^{tile} = \left(\frac{1}{2} - \frac{\ln(\tan(lat_b * \pi / 180) + \sec(lat_b * \pi / 180))}{2\pi} \right) * 2^l \end{cases} \quad (1)$$

Latitude and longitude coordinates $(long_{corner}^{tile}, lat_{corner}^{tile})$ to pixel coordinates $(x_{corner}^{pixel}, y_{corner}^{pixel})$,

$$\begin{cases} x_{corner}^{pixel} = \frac{long_{corner}^{tile} + 180}{360} * 2^l * 256 \% 256 \\ y_{corner}^{pixel} = \left(1 - \frac{\ln(\tan(\frac{long_{corner}^{tile} * \pi / 180} + \sec(\frac{long_{corner}^{tile} * \pi / 180}))}{2\pi})\right) * 2^l * 256 \% 256 \end{cases} \quad (2)$$

Pixel coordinates of tiles $(x_{corner}^{pixel}, y_{corner}^{pixel})$ to longitude and latitude coordinates $(long_{corner}, lat_{corner})$,

$$\begin{cases} long_{corner} = \frac{x_{corner}^{tile} + \frac{x_{corner}^{pixel}}{256}}{2^l} * 360 - 180 \\ lat_{corner} = \arctan\left(\sinh\left(\pi - 2\pi \frac{y_{corner}^{tile} + \frac{y_{corner}^{pixel}}{256}}{2^l}\right)\right) * \frac{180}{\pi} \end{cases} \quad (3)$$

where l represents the zoom level of tiles.

3.2. Motion Constraints

The primary aim of the UAV motion constraint is to establish the UAV's real-time flight speed. Based on the previously known position, a more reasonable estimate is developed for the UAV's position at the next time. The main objective of this step is to minimize the size of satellite image matching while enhancing matching calculation efficiency. Figure 4 illustrates a schematic depiction of the solution, with the green dotted box representing the UAV's most likely position at the next period.

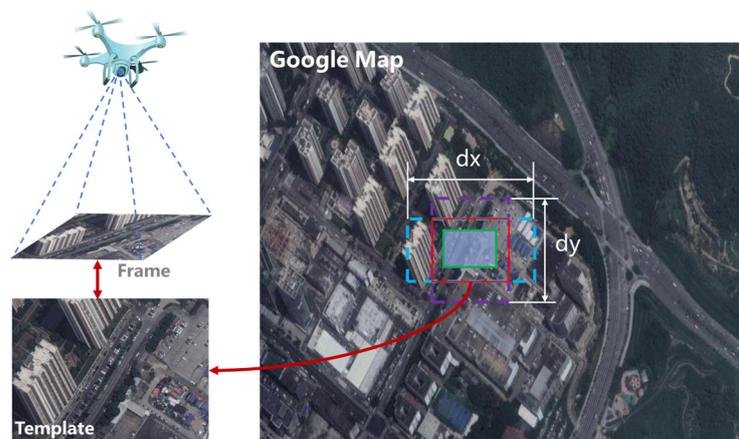


Figure 4. The diagram illustrates the solution to the motion constraints. The purple and blue dashed boxes represent the potential positions of the subsequent frame of the image based on a single-direction evaluation. The red solid line represents the potential position based on a combined evaluation of the two directions, while the green solid line represents the anticipated actual position.

The error state Kalman filter (ESKF) [34] is capable of accurately predicting the movement state of UAVs. The alteration in position and speed of a fixed-wing UAV during flight is primarily owing to forward motion; consequently, only scalar speed is evaluated. Figure 4 demonstrates that the matching template image corresponding to the real-time frame captured by the camera is the overlapping part of purple and blue dotted lines in the Google reference map. The two dotted boxes represent the various positions of UAVs obtained through moving in two directions (heading and sideways) at the same time.

Assuming that the UAV's present position, as established through filtering, is (x_t, y_t) and its velocity is (v_x, v_y) , the template image's estimated heading length and width are

dy and $\frac{dy}{2}$, respectively; the estimated lateral length and width are dx and $\frac{dx}{2}$, respectively. The geometric relationship can be used to formulate dx and dy as follows:

$$\begin{cases} \frac{dx}{2} = v_y \cdot \Delta t \\ \frac{dy}{2} = v_x \cdot \Delta t \end{cases} \quad (4)$$

where Δt is the time interval between the two image frames that are required to match.

Therefore, the coordinates of the lower left corner (x_t^{ll}, y_t^{ll}) and the upper right corner (x_t^{ur}, y_t^{ur}) of the template image are expressed as follows:

$$\begin{cases} x_t^{ll} = x_t - \frac{dx}{4}, y_t^{ll} = y_t - \frac{dy}{4} \\ x_t^{ur} = x_t + \frac{dx}{4}, y_t^{ur} = y_t + \frac{dy}{4} \end{cases} \quad (5)$$

3.3. Geographic Image Point–Line Matching

Conventional map matching algorithms generally employ points that have larger gradient values that are positioned at corners as matching features. However, in real-world situations, the number of stable feature points that can potentially be recovered from an image is limited. We noticed that, when compared to feature points, straight line features of artificial facilities such as buildings in aerial images in large-scale environments have stronger stability and anti-interference capabilities; additionally, in typical scenes such as urban environments, the probability of straight line textures appearing is higher than feature points. Figure 5 demonstrates that in a large-scale urban context, straight line elements between two adjacent aerial images are significantly more prevalent than point features. As a result, employing straight lines as matching features could enhance both accuracy and efficiency.

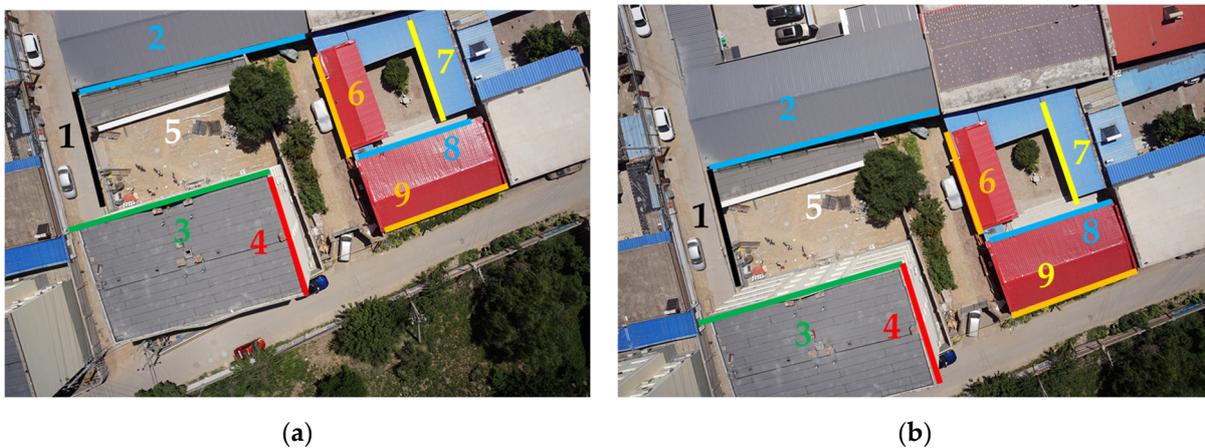


Figure 5. A schematic comparison of the straight line elements (nine straight line segments listed in different colors) in two adjacent aerial images of a landscape of cities. (a,b) are two adjacent aerial frames, and 1–9 represent same straight line segment features in different images.

Classical descriptor-based line matching algorithms (e.g., LBD [35], MSLD [36], LSD [37], EDLines [38]) model each line segment based on its local appearance and geometric constraints before matching it. However, these approaches have a considerable amount of computation and are challenging to employ for real-time work; additionally, the matching rate of success is mediocre. For continuous frame images, the offset of the line texture is modest. Based on this assumption, we may employ certain efficient feature point matching algorithms to achieve the line matching function. This work presents a line segment-based point matching approach to reduce the number of matching com-

putations required. Figure 6 illustrates the basic principle of anchor-based geographic image point-line matching. Firstly, Fast [20] is employed for determining corner points, followed by EDlines for recognizing line segments. Some significant points on the line segment have been picked as anchor points, and these points are used to symbolize the line during operation. The matching points are then counted, resulting in corner point and line segment tracking.

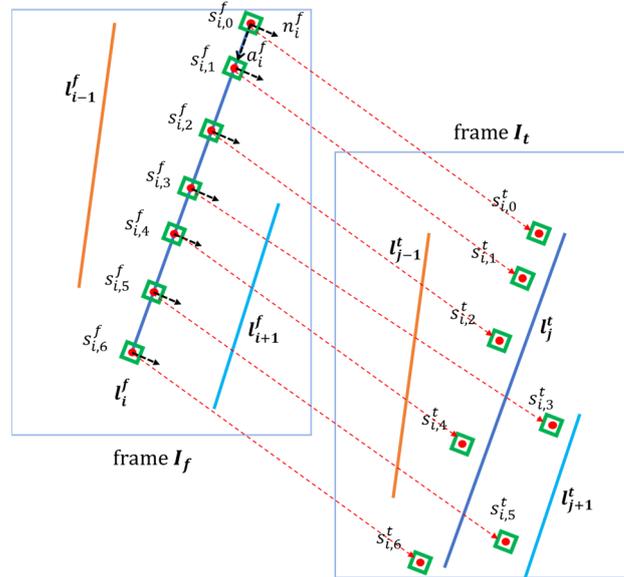


Figure 6. Schematic of anchor-based geographic image point-line matching. Frame I_f and Frame I_t are adjacent frames. The three straight lines indicate line features, while the red dots represent anchor points on the line segments. Employing the optical flow procedure, the anchor points are utilized as feature points to move on frame I_t in an attempt to find the corresponding anchor points to retrieve the line features.

Given two matched images I_f and I_t , we employ EDLines to detect straight lines in each image and establish a collection of straight line segments $L_f = \{l_i^f | l_i^f = (p_{i,0}^f, p_{i,1}^f), 0 \leq i \leq N_f\}$ in image I_f , where N_f is the number of lines. $p_{i,0}^f, p_{i,1}^f$ represent the 2D coordinates $p = [x, y]^T$ of the endpoints of the image's i -th line segment l_i^f . Similarly, Image I_t yields a set of straight line segments $I_t = \{l_j^t | l_j^t = (p_{j,0}^t, p_{j,1}^t), 0 \leq j \leq N_t\}$. For each line segment $l_i^f \in I_f$ on the image $I_t = \{l_j^t | l_j^t = (p_{j,0}^t, p_{j,1}^t), 0 \leq j \leq N_t\}$, the direction vector \mathbf{a}_i^f and normal vector \mathbf{n}_i^f are expressed.

$$\mathbf{a}_i^f = [a_{i,0}^f, a_{i,1}^f]^T = \frac{p_{i,1}^f - p_{i,0}^f}{|p_{i,1}^f - p_{i,0}^f|} \tag{6}$$

$$\mathbf{n}_i^f = [-a_{i,1}^f, a_{i,0}^f]^T \tag{7}$$

Utilizing FAST corner detection on any line segment $l_i^f \in I_f$ in image I_f , we subsequently employ the found corner points as anchor points. The anchor point set is $S_i^f = \{s_{i,j}^f, 0 \leq j \leq N_i\}$, and N_i is the number of line segment anchor points.

In visual-inertial odometry, for each anchor point on line segment $l_i^f \in I_f$ in image I_f , with $s_{i,j}^f$ as the starting position and \mathbf{n}_i^f as the search direction, a matching point is found on image I_t . $s_{i,j}^f$ is the anchor point on image I_f , $s_{i,j}^t$ is the corresponding matching

point on image I_t , $L_t = \{l_j^t | l_j^t = (p_{j,0}^t, p_{j,1}^t), 0 \leq j \leq N_t\}$ is the set of straight line segments in image I_t , and N_t is the number of lines. We calculate the distance $d_{i,j}$ between each matching point $s_{i,j}^t$ and the straight line segments l_j^t in image I_t .

$$d_{i,j} = \left\| \min \left(\max \left(\frac{-(p_{j,1} - p_{j,0})^T (p_{j,0} - s_{i,j})}{(p_{j,1} - p_{j,0})^T (p_{j,1} - p_{j,0})}, 0 \right), 1 \right) \cdot (p_{j,1} - p_{j,0}) + (p_{j,0} - s_{i,j}) \right\| \quad (8)$$

$$d_{\min} = \min_{\substack{0 \leq i \leq N_i \\ 0 \leq j \leq N_t}} d_{i,j} \quad (9)$$

where d_{\min} is the distance between all straight line segments l_j^t in image I_t , and d_{th} is the empirical threshold. When $d_{\min} < d_{th}$, point $s_{i,j}^t$ is considered to belong to line l_j^t .

When the ratio of anchor points falling on a straight line to the total anchor points meets a particular threshold, the two straight lines are considered to be equivalent.

Nevertheless, our matching strategy is different from visual-inertial odometry in geographical image matching. For obtaining the anchor point set $S_i^t = \{s_{i,j}^t | 0 \leq j \leq N_j\}$ on any line segment $l_i^t \in I_t$ in image I_t , we begin by performing the same procedure, where N_j is the number of anchor points on the line segment. Subsequently, we make use of the anchor points as feature points, and then we need to estimate the similarity of the two anchor point sets based on the lowest Euclidean distance d_2 .

$$d(S_i^f, S_i^t) = \sqrt{\sum_{j=1}^N (s_{i,j}^f - s_{i,j}^t)^2} \quad (10)$$

The final key point descriptors of the two pairs have to meet the following conditions.

$$\frac{\text{The nearest point } S_i^f \text{ from } S_i^t \text{ in the frame}}{\text{The next nearest point } S_j^f \text{ from } S_i^t \text{ in the frame}} < \text{Threshold} \quad (11)$$

3.4. Joint State Estimation

The approach estimates the multi-rotor UAV's global position and attitude employing a monocular RGB camera, an IMU (merely for UAV attitude), and a set of geo-referenced Google images.

Suppose $T_{W,k}$ is the conversion from UAV at the key frame to the global frame. The position of UAV in the world coordinate system is determined by $r_W^{k,W} = [x_W^{k,W}, y_W^{k,W}, z_W^{k,W}]^T$ and roll, pitch, and yaw $(\phi_{W,k}, \theta_{W,k}, \psi_{W,k})$, which can be determined from the rotation matrix $C_{W,k}$. Let $\uparrow^q = (I_1^q, I_2^q, \dots, I_K^q)$ be a sequence of real UAV query images from each key frame. We use a set of geo-referenced map images to locate each key frame image $\uparrow^m = (I_1^m, I_2^m, \dots, I_N^m)$. The global attitude of the map image is T_{W,m_s} .

$$T_{W,k} = \begin{bmatrix} C_{W,k} & r_W^{k,W} \\ 0^T & 1 \end{bmatrix} \quad (12)$$

The initial step in the estimate is to apply visual odometry to the UAV image. The adjusted gray image and the non-static UAV-to-sensor transformation are employed as inputs, with each frame calculated at 10 Hz. It is computed using the known translation between the three angles to generate a composite transformation $T_{f_s,f}$, which is then rotated into an ordinary camera frame. When yaw follows the UAV's heading, the roll and pitch axes remain globally stable in the gravity-aligned inertial frame.

For each frame of the image, features are extracted and SIFT descriptors are matched between frames to perform landmark triangulation. Features that cannot be triangulated through matching are triangulated through motion between consecutive frames. The descriptor in the latest image is matched with the last key frame to generate a 2D–3D point correspondence. The maximum likelihood estimation sample consensus (MLE-SAC) estimator is used to determine the uncertain full SE (3) incremental UAV attitude from the current frame to the last key frame $T_{f,k}, \sum f, k$. If the translation or rotation exceeds the threshold, or the number of interior points is below the minimum, new key frames are added. For each new key frame, windowing refinement (clump adjustment) is performed using Simultaneous Track Estimation and Mapping (STEAM). For each key frame image I_k^q , the goal is to determine the relative SE (3) pose between the query camera at k and the camera of the adjacent image T_{k_s, n_s} and then obtain the global attitude measurement value of the UAV.

$$T_{W,k} = T_{W, n_s} T_{k_s, n_s}^{-1} T_{k_s, k} \quad (13)$$

The state fusion is carried out by combining the $\mathbf{T}_{k,k-1}$ uncertainty relative transformation of visual odometry and the uncertain attitude measurement value of image registration $\mathbf{T}_{k,0}$. It is worth noting that $\mathbf{T}_{W,0}$ is the transformation from the local coordinate system to the global coordinate system, and it is constructed from the GNSS pose at the first key frame. Therefore, the filtering equation can be expressed as

$$\hat{\mathbf{P}} = \mathbf{Q}_k + \boldsymbol{\tau}_{k,k-1} \hat{\mathbf{P}}_{k-1} \boldsymbol{\tau}_{k,k-1}^T \quad (14)$$

$$\hat{\mathbf{T}}_{k,0} = \mathbf{T}_{k,k-1} \hat{\mathbf{T}}_{k-1,0} \quad (15)$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_k (\hat{\mathbf{P}}_k + \mathbf{R}_k)^{-1} \quad (16)$$

$$\hat{\mathbf{P}}_k = (1 - \mathbf{K}_k) \hat{\mathbf{P}}_k \quad (17)$$

$$\hat{\mathbf{T}}_{k,0} = \exp \left(\left(\mathbf{K}_k \ln \left(\mathbf{T}_{k,0} \hat{\mathbf{T}}_{k,0}^{-1} \right)^\vee \right)^\wedge \right) \hat{\mathbf{T}}_{k,0} \quad (18)$$

where $\boldsymbol{\tau}_{k,k-1}$ is the adjoint of $\mathbf{T}_{k,k-1}$, the prior uncertainty $\hat{\mathbf{P}}_k$ is the second-order approximation, \mathbf{K}_k is the Kalman gain, and $\ln(\cdot)^\vee$, $\exp(\cdot)^\wedge$ are SE(3) operators.

The proportion of visual odometry is computed by minimizing the difference between the incremental posteriori in the keyframe window and the translation of the visual odometry. For failure matching, the expected position and uncertainty are propagated, resulting in a posteriori global attitude for each key frame.

4. System Description

The RTAPM approach proposed in this study employs a monocular RGB camera and involves prior map information to provide real-time position. The compiled and encapsulated UAV positioning algorithm is universally compatible and can run independently of the positioning module. Figure 7 illustrates the framework of the RTAPM experimental system. The data from the flying control and camera are collected, utilizing ROS. Each module receives various sensor data via the rospy publish–subscribe procedure and processes and calculates it in its own module.

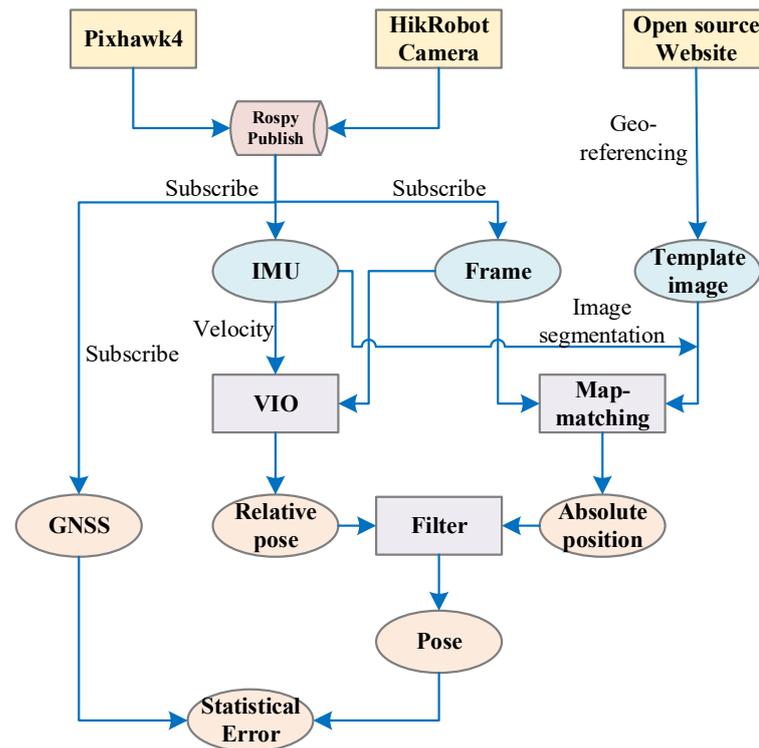


Figure 7. Framework of the RTAPM experimental system.

4.1. RTAPM Experimental Platform Design

For the development of the experimental test system, a widely common computer module, the NVIDIA Jetson Nano, was employed, which could provide up to 472 GFLOPS of computing power while consuming only 5 W of power. Jetson Nano runs Ubuntu 20.04 and features a quad-core Cortex-A57 CPU and the smallest Maxwell architecture GPU, with only 128 CUDA units, 4 GB LPDDR4 memory, and 16 GB of storage capacity.

Figure 8 shows the rotor UAV platform used for the experiment's flight tests. The rotor UAV is compatible with the down-looking 6 mm fixed-focus monocular RGB camera, the airborne core processing unit Jetson Nano processing board, and the Pixhawk 4 flight control built-in IMU module, which can be used to monitor the UAV's flight attitude during online positioning, while the positioning data of the real-time GPS-RTK module can be used as a reference value for visual matching positioning to verify the algorithm's accuracy.

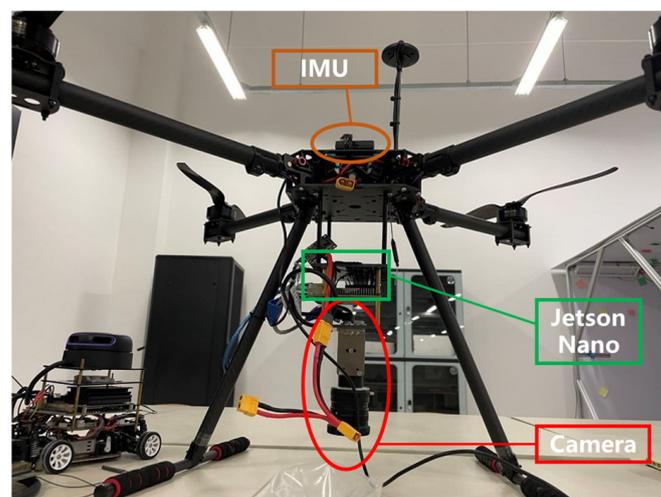


Figure 8. UAV experimental platform for RTAPM.

4.2. Data Communication and Transmission

For monitoring purposes of the UAV, the positioning module's real-time position needs to be transmitted to the ground station via flight control. The Jetson Nano airborne computer, Pixhawk flight control, and ground terminal computer are the equipment used for Mavros communications. The airborne computer and flight control are linked together by a USB data line, while the flight control and ground computer are associated by a wireless data transmission module, as illustrated in Figure 9.

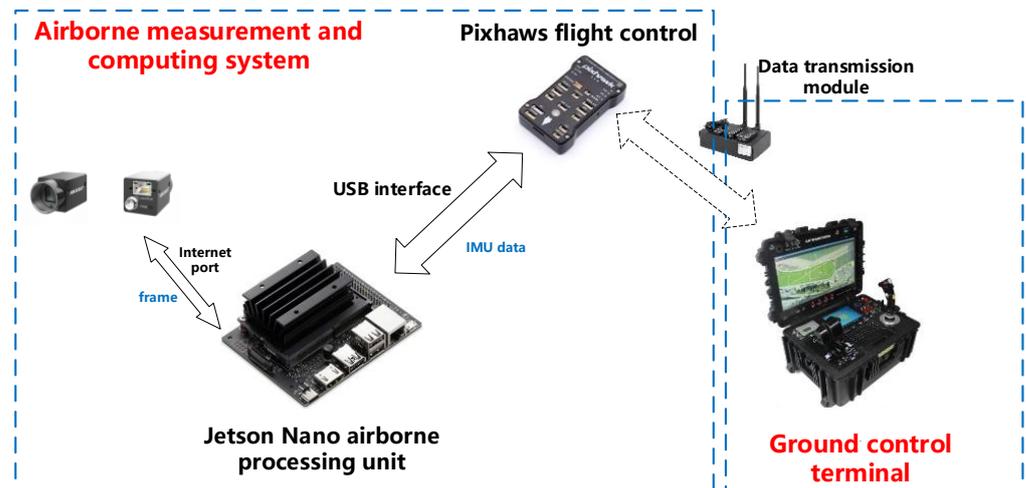


Figure 9. UAV experimental platform Mavros communication system structure.

5. Experiments and Results

5.1. Aerial Image Datasets

The UAV aerial image dataset employed in this study is divided into two parts: an open-source dataset and a real-world test. The aerial image datasets are mostly used for algorithm testing. The datasets were collected by a quadrotor or vertical UAV. During flight, the UAV was equipped with a nine-axis IMU and a monocular RGB camera, and it operated at a relatively constant height. At the same time, the dataset contained real-time flight position and attitude information provided by the GPS-RTK system and IMU. These pose data can be employed as truth values for evaluating the accuracy of our algorithm's real-world positioning. The main parameters of the aerial image dataset are shown in Table 1.

Table 1. Introduction to airborne image dataset.

Datasets	Camera Type	Flight Height (m)	Frame Rate (FPS)
Dataset1	Unknown	60 m	30
Dataset2	Unknown	200 m	15
Dataset3 *	Hikvision MV-CA050-10GM/GC	50 m	18

* Dataset3 is composed of images, IMU, and GNSS data collected by the UAV experimental platform for RTAPM in real-world flight tests.

5.2. Select Imaging Parameters and Compare Algorithms

The camera's sample frequency was relatively high. To lower the data processing capacity of the Jetson Nano computing module, we had to choose a suitable quantity of frames to extract the camera's real-time frames while maintaining positioning accuracy prior to positioning. We had to compare two groups: one for analyzing the effect of the frame number on the processing time of the processing board, and the other to calculate

the impact of an integer on positioning results. For evaluating the processing time and positioning error, we used 10 s for real-time frame samples. Considering the output frequency of our positioning results was 1 Hz, we estimated the average processing time within 1 s.

As shown in Figure 10, we selected one-third of the sample frequency for frame extraction by comparing datasets with different sampling frequencies without increasing the positioning error, minimizing calculation costs. In order to ensure that the prediction range provided by the motion constraint boosted the computational efficiency of the template image after clipping, we recorded each dataset’s operation time and compared it to that before clipping. Table 2 demonstrates that the motion constraint enhances positioning efficiency by approximately an order of magnitude.

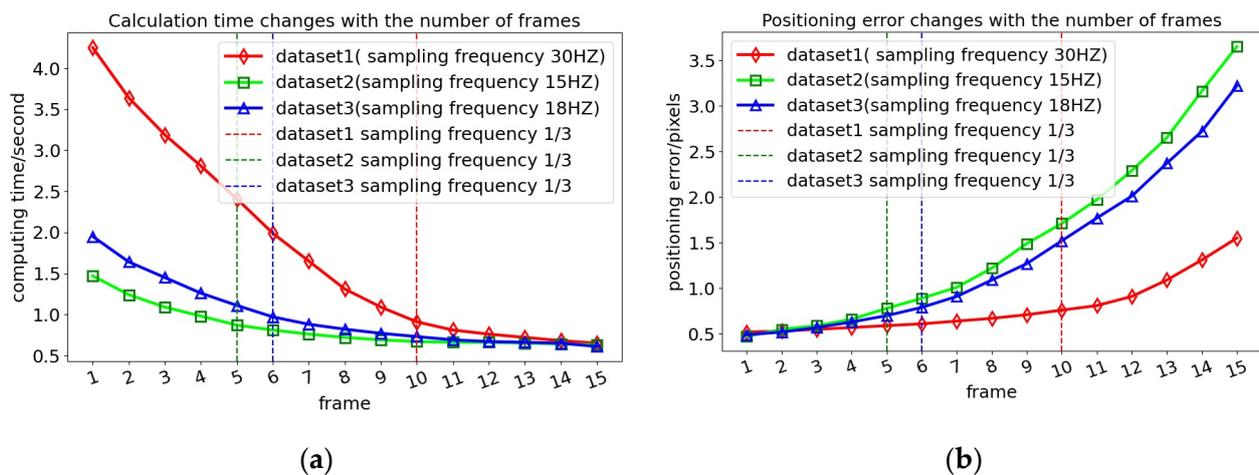


Figure 10. Relationship between image parameters and computing resources. (a) Relationship between image sampling frequency and computing time; (b) relationship between image sampling frequency and positioning error.

Table 2. Comparison of dataset calculation time before and after template image clipping.

Datasets	Each Positioning Time of Clip Template Image Without Motion Constraint (s)	Each Positioning Time of Clip Template Image with Motion Constraint (s)
Dataset1	1.59	0.303
Dataset2	1.66	0.291
Dataset3 *	1.45	0.323

* Dataset3 is composed of images, IMU, and GNSS data collected by the UAV experimental platform for RTAPM in real-world flight tests.

For the purpose of evaluating the effectiveness and dependability of the FAST combined with EDlines algorithm presented in this study in geographic imagery, the proposed approach is compared to several mainstream line matching methods in terms of feature extraction quantity and extraction time. Figure 11 and Table 3 show the experimental statistical results. Experimental results demonstrate that the proposed approach, while combined with FAST corner detection, productively eliminates a large number of redundant line segment features and employs only a few line segment features to provide more effective and high-precision matching while maintaining matching accuracy. As the amount of online features declines, it consequently decreases the running time necessary to extract those features.

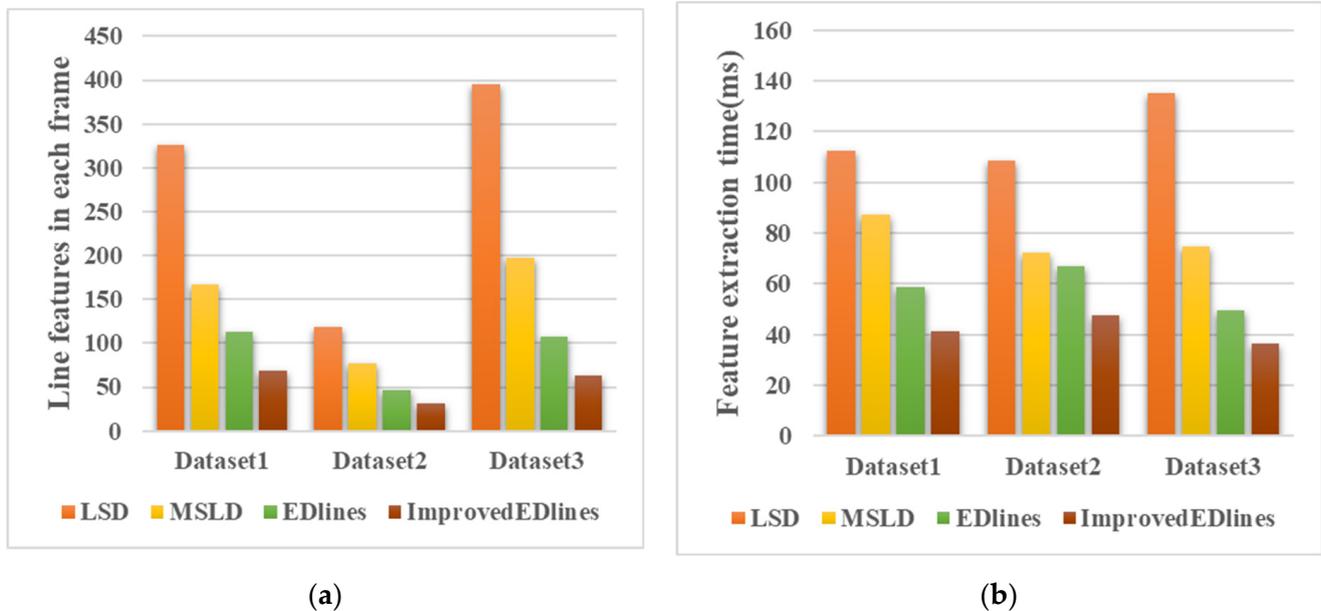


Figure 11. Comparing the number of point and line features extracted every frame, as well as the running time spent to extract features employing different approaches on various datasets. (a) The number of point and line features extracted from each frame; (b) the running time of feature extraction.

Table 3. The number of point and line features extracted from each frame and the running time of feature extraction.

Dataset	LSD		MSLD		EDlines		Improved EDlines	
	Line Features in Each Frame	Feature Extraction Time (ms)	Line Features in Each Frame	Feature Extraction Time (ms)	Line Features in Each Frame	Feature Extraction Time (ms)	Line Features in Each Frame	Feature Extraction Time (ms)
Dataset1	326	112.477	167	87.371	113	58.924	69	41.135
Dataset2	118	108.664	77	72.459	46	66.983	32	47.386
Dataset3 *	395	135.346	198	74.882	108	49.579	63	36.247

* Dataset3 is composed of images, IMU, and GNSS data collected by the UAV experimental platform for RTAPM in real-world flight tests.

5.3. Dataset Comparison and Real-World Experiment

Dataset 1: The data were collected at Pakistan's National University of Science and Technology. The flying height of the UAV and other sensors in this dataset was around 50–60 m during data collection, with a total collecting period of 45 s. The model and parameter details of the traveling camera remain unknown. The frame rate for collecting images was 30 FPS, and the total number of images was approximately 1352. The data were collected in an urban area environment. The flight geographical area comprised primarily roadways and buildings, with a large number of image texture features.

The image data in this dataset have a high sampling frequency and are without frame loss. In order to guarantee a stable and smooth data processing process, frame extraction was performed throughout data preprocessing, and the positioning results became available at 1 Hz. Figure 12 illustrate the calculation results of VIO, Geo-ref. + VIO, our approach (RTAPM), and the GPS-RTK position on the template image. The GPS-RTK position can be used as a reference trajectory for estimating the positioning errors for both methods. The positioning results from VIO, Geo-ref. + VIO, and the approach presented in this study, as well as the GPS trajectory, are symbolized as red solid lines, green solid lines, blue solid lines, and yellow circles, respectively.

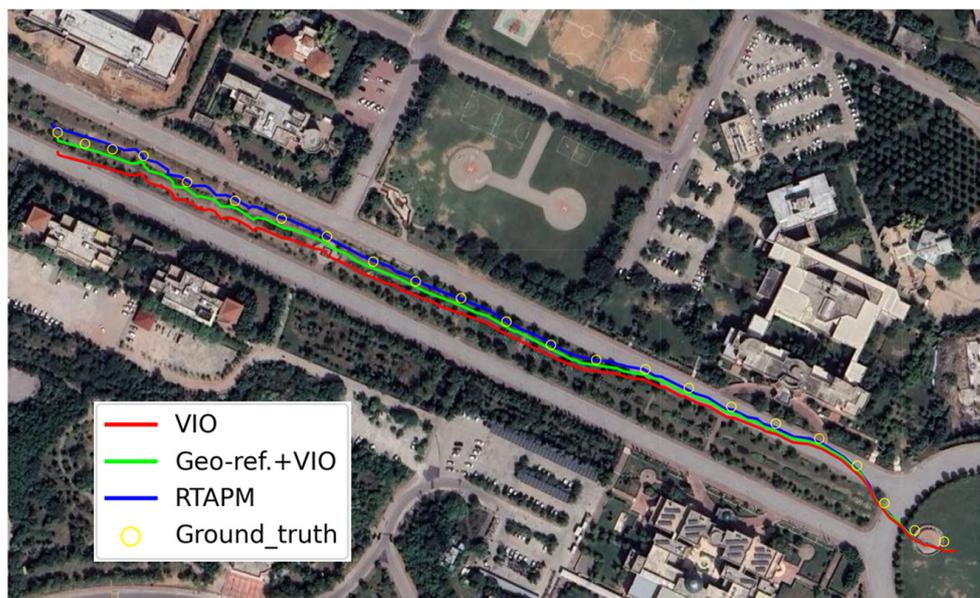


Figure 12. Comparison of the positioning trajectories of three algorithms on Dataset 1. The true trajectory is represented as a yellow circle, VIO's trajectory as a red solid line, Geo-ref. + VIO's trajectory as a green solid line, and RTAPM's trajectory as a blue solid line.

Figure 13a demonstrates the positioning estimation errors of VIO (red dashed line), Geo-ref. + VIO (green solid line), and RTAPM in the horizontal and vertical directions on various datasets. When utilizing the VIO system alone, positioning errors are cultivated over time, and error accumulation in the X direction occurs faster than in the Y direction. This might be since the feature similarity in the forward direction (X direction) is drastically higher than in the Y direction, and the positioning error in the Y direction is extremely similar and has always been high due to terrain features. The emergence of the absolute positioning approach has lowered the positioning error in the Geo-ref. + VIO strategy, but the overall error remains high. It is possible that the VIO system fails to obtain an accurate location in a short period of time; consequently, the data fusion procedure should be expanded. Compared to Geo-ref. + VIO, the approach proposed in this study reduces the positioning error to a very small range at the beginning of the joint state optimization, achieves accuracy optimization, and significantly eliminates VIO-induced position drift. Figure 13b shows the cumulative distribution of absolute position estimation error among three approaches. According to 3-sigma distribution statistics, the proposed method's 1-sigma and 3-sigma positioning accuracy are greater than 5.5 m and 6.5 m, respectively. Whether considering the overall error cumulative distribution or single point accuracy, the method proposed in this study outperforms the other two methods in terms of stability.

Dataset 2: The data underlying the experiment were collected by unmanned aerial vehicles in Rothenthurm, Switzerland. Figure 14 illustrates the map of the area. The flying height of the UAV along with the other sensors in this dataset was approximately 200 m, with a total capture time of 132 s. The model and parameter details of the traveling camera remain ambiguous. The frame rate of capturing images was 15 frames every second, yielding a total of around 1982 frames. Datasets were collected in urban areas. The majority of the flight area consisted of roads and grassy fields. The image texture features were sparse, and the recognition accuracy was low. The yellow circle represents the UAV's true flying path (GPS-RTK signal).

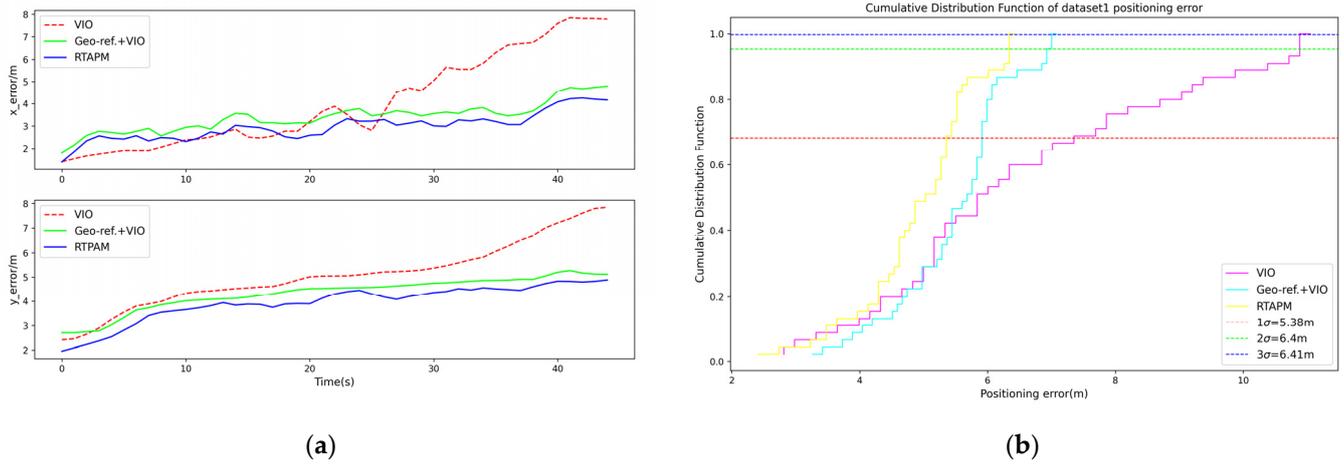


Figure 13. (a) Horizontal position estimation error in the X and Y directions (top and bottom); (b) cumulative distribution of positioning errors in Dataset 1.

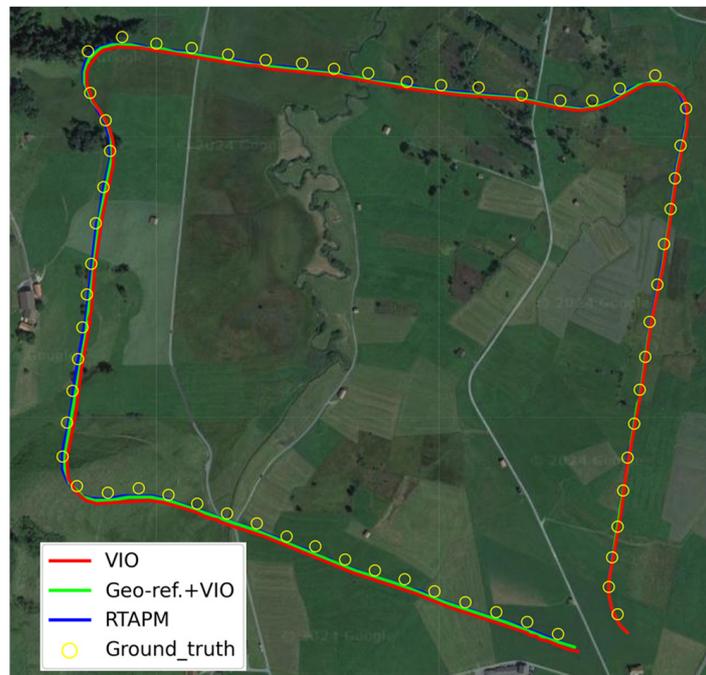


Figure 14. Comparison of the positioning trajectories of three algorithms on Dataset 2. The true trajectory is represented as a yellow circle, VIO’s trajectory as a red solid line, Geo-ref. + VIO’s trajectory as a green solid line, and RTAPM’s trajectory as a blue solid line.

Figure 15a demonstrate the positioning estimation errors of VIO (red dashed line), Geo-ref. + VIO (green solid line), and RTAPM in the horizontal and vertical directions on various datasets. Figure 15a demonstrates that following the UAV’s first turn, the drift error of VIO emerged instantly and gradually accumulated. After the third rotation, the cumulative error of VIO became apparent above 12 m; nevertheless, the way maps were employed was comparable to GNSS, which limited the accumulation of drift errors and maintained them within a stable range. Figure 15b demonstrate the cumulative distribution of absolute position estimation errors among the three methods. According to the 3-sigma distribution statistics, the proposed method’s 1-sigma and 3-sigma positioning accuracy exceeded 5.5 m and 6.0 m, respectively.

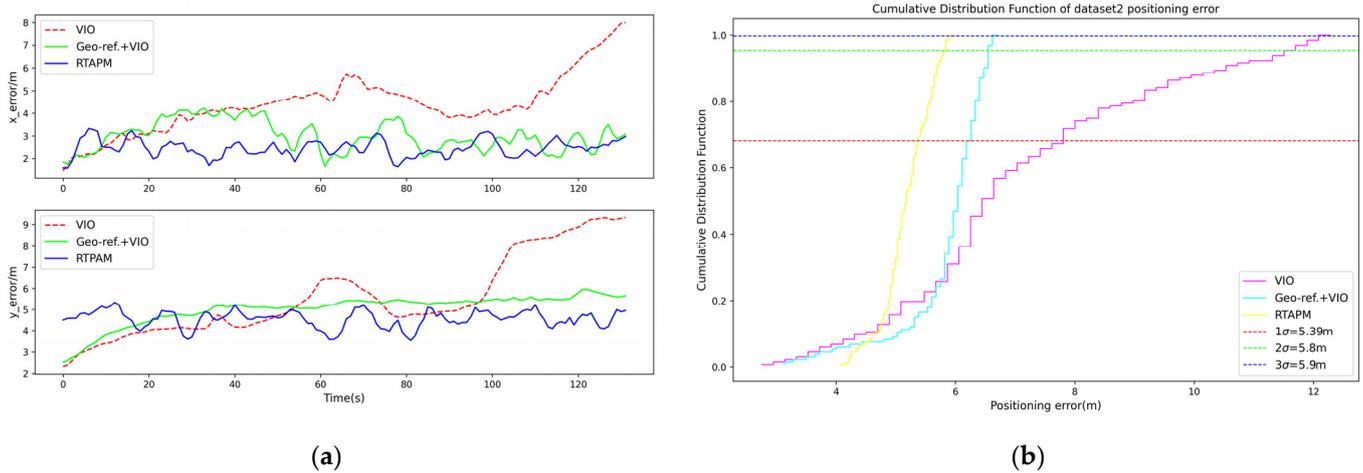


Figure 15. (a) Horizontal position estimation error in the X and Y directions (top and bottom); (b) cumulative distribution of positioning errors in dataset 2.

Dataset 3 (real-world flight experiment): After offline testing of datasets, we employed our own developed UAV hardware platform equipped with a Hikvision MV-CA050 camera and an Nvidia Jetson Nano processor to conduct a real-time flight online positioning test at Sun Yat-sen University's Shenzhen Campus, as well as transmitting the positioning results back to the ground station via data transmission. Figure 16 displays an image of the area, on which we overlaid the real-time flight trajectories.

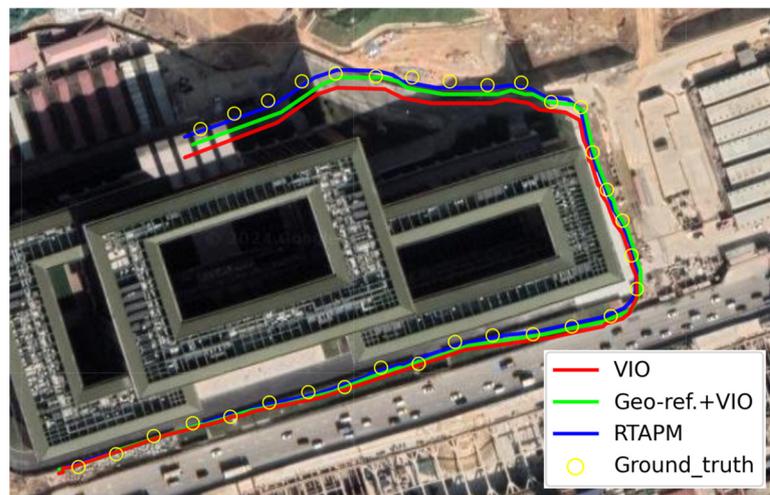


Figure 16. Comparison of the positioning trajectories of three algorithms on Dataset 3*. The true trajectory is represented as a yellow circle, VIO's trajectory as a red solid line, Geo-ref. + VIO's trajectory as a green solid line, and RTAPM's trajectory as a blue solid line.

Figure 17a demonstrate the positioning estimation errors of VIO (red dashed line), Geo-ref. + VIO (green solid line), and RTAPM in the horizontal and vertical directions on various datasets. Figure 17a shows that employing geographic reference effectively eliminated drift and minimized errors in the orthogonal direction. Geographic reference did not significantly decrease the error in the Y direction, and drift was effectively diminished. Figure 17b demonstrate the cumulative distribution of absolute position estimation errors among the three methods. According to the 3-sigma distribution statistics, the proposed method's 1-sigma and 3-sigma positioning accuracy exceeded 5.2 m and 5.6 m, respectively.

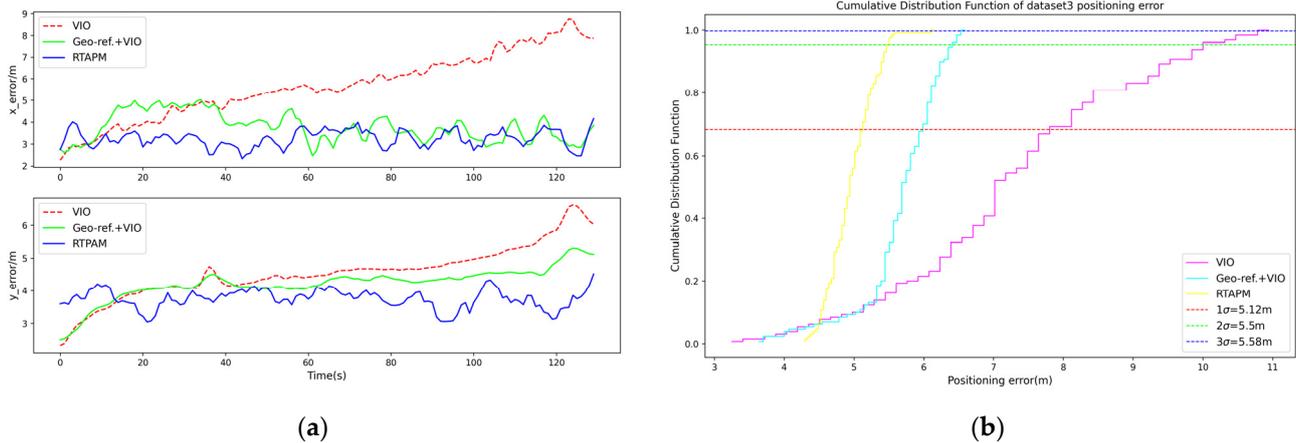


Figure 17. (a) Horizontal position estimation error in the X and Y directions (top and bottom); (b) cumulative distribution of positioning errors in Dataset 3 (real-world flight experiment).

Figure 18 demonstrates the positioning errors of the three approaches on three different datasets in the form of box plots. The graph illustrates instinctively that the VIO method’s positioning error distribution is appropriately discrete, with drift accumulation growing over time. Whereas the Geo-ref. + VIO approach decreases positioning error by a certain quantity through employing map matching, there are several discrete positioning results on datasets 2 and datasets 3, demonstrating that its positioning stability is barely enough. In comparison, the RTAPM approach productively solves this problem, with the positioning error becoming extremely clustered and relatively small. Figure 18b,c show a remarkable phenomenon. The Geo-ref. + VIO method’s positioning errors are separate outside the box plots (small circles), with minimal error levels. This demonstrates that the absolute positioning method is effective in reducing errors; however, because it is loosely integrated with VIO, the benefits of both are not reached. This highlights the worth of joint state optimization and the benefits of the strategy proposed in this study.

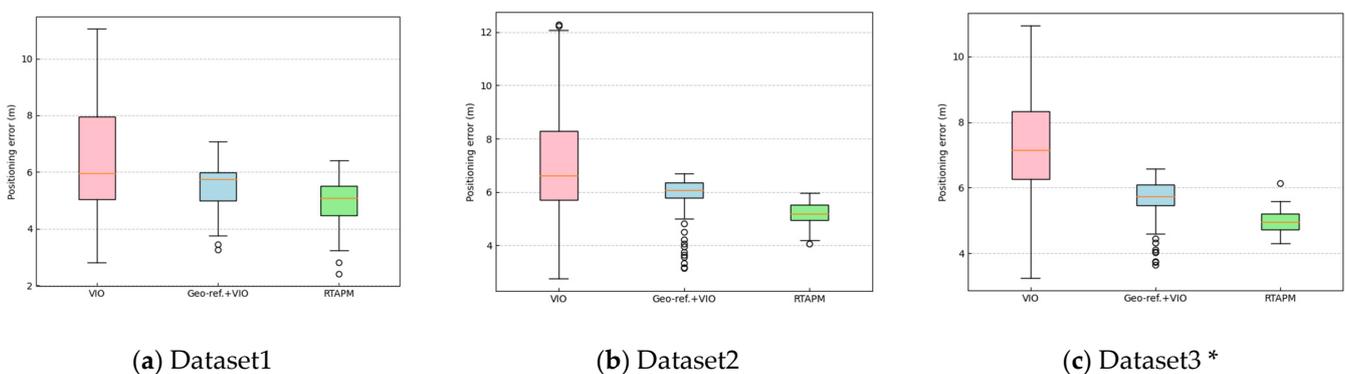


Figure 18. Box plots are employed to compare the three algorithms’ positioning error distributions within the three datasets. The box plot’s small black circles indicate error values that are either too small or too large and are not within the range of the mean. (a–c) show the statistical results for positioning errors on datasets 1, 2, and 3, respectively. Dataset3* is composed of images, IMU, and GNSS data collected by the UAV experimental plat-form for RTAPM in real-world flight tests.

To further analyze our experimental results, we performed a correlation analysis on the positioning results for the three experiments, calculated the essential accuracy evaluation indicators, and used graphs and tables for straightforward display. Figure 19 and Table 4 demonstrate the maximum, mean, and root mean square error of the three methods’ positioning errors on three different datasets. At the same time, Figure 20 and Table 5 illustrate the 3sigma distribution of the three algorithms’ positioning errors on

three distinct datasets. Numerous types of evaluation indicators demonstrate that the absolute positioning method proposed in this study exceeds the compared methods in many categories.

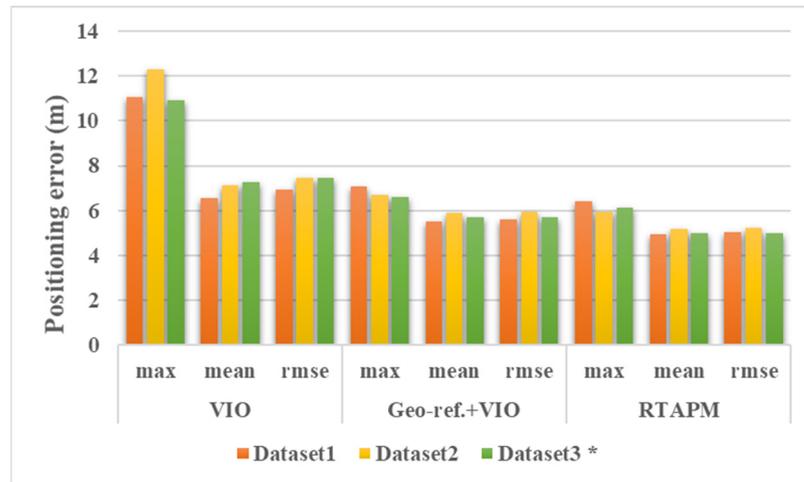


Figure 19. Comparison of the three algorithms’ positioning errors on three datasets (maximum error, mean error, and root mean square error). Dataset3* is composed of images, IMU, and GNSS data collected by the UAV experimental plat-form for RTAPM in real-world flight tests.

Table 4. Statistical errors (unit: m) (maximum, mean, root mean square error(rmse)) of various algorithms on different datasets.

Dataset	VIO			Geo-ref. + VIO			RTAPM		
	Max	Mean	Rmse	Max	Mean	Rmse	Max	Mean	Rmse
Dataset1	11.05	6.54	6.94	7.08	5.51	5.58	6.42	4.95	5.03
Dataset2	12.28	7.11	7.45	6.69	5.88	5.93	5.95	5.19	5.21
Dataset3*	10.94	7.28	7.47	6.59	5.69	5.72	6.14	4.97	4.99

* Dataset3 is composed of images, IMU and GNSS data collected by the UAV experimental platform for RTAPM in real-world flight tests.

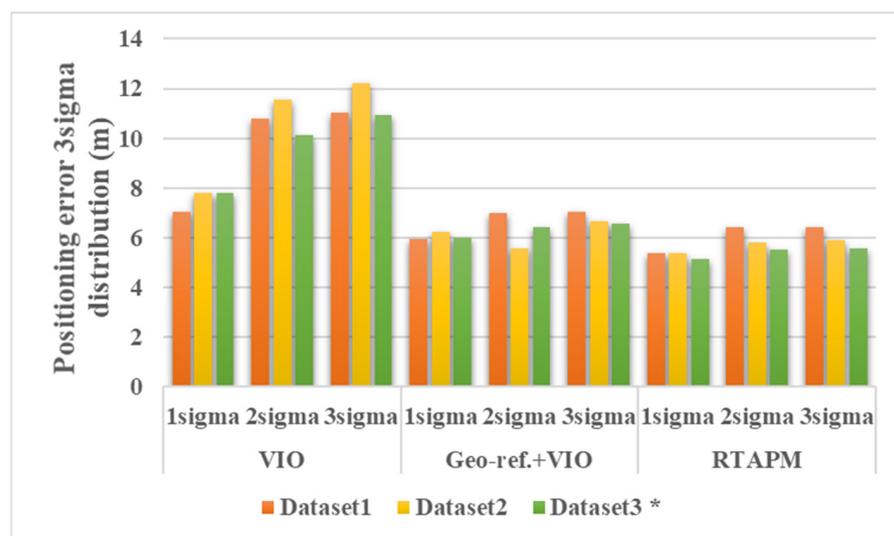


Figure 20. Comparison of the three approaches’ 3sigma distributions of positioning errors among three datasets. Dataset3* is composed of images, IMU, and GNSS data collected by the UAV experimental plat-form for RTAPM in real-world flight tests.

Table 5. Statistical errors (unit: m) distribution (1σ , 2σ , 3σ) of various algorithms on different datasets.

Dataset	VIO			Geo-ref. + VIO			RTAPM		
	1σ	2σ	3σ	1σ	2σ	3σ	1σ	2σ	3σ
Dataset1	7.06	10.77	11.03	5.93	6.97	7.02	5.38	6.40	6.41
Dataset2	7.81	11.53	12.23	6.24	5.59	6.68	5.39	5.80	5.90
Dataset3 *	7.81	10.13	10.94	5.98	6.43	6.58	5.12	5.50	5.58

* Dataset3 is composed of images, IMU, and GNSS data collected by the UAV experimental platform for RTAPM in real-world flight tests.

6. Conclusions

In this study, a map-based image matching positioning method compatible with UAVs was established to address the issue of employing satellite maps for visual positioning in the absence of GNSS satellite signals. The motion constraint module was then utilized to determine a limited map matching range, which significantly increased the speed of matching positions. The monocular RGB camera, IMU, and geo-referenced Google images were employed to optimize and estimate the UAV's global pose using point and line features. Experimental results show that the proposed approach is capable of positioning accuracy more than 6 m using only open-source Google images, which is not limited on terrain, and the technical implementation is straightforward. It is more robust and reliable than previous methods, significantly improving the accuracy and universality of UAV autonomous position. However, this approach contains several limitations, such as not properly taking into account the capture of data from visible light cameras under extreme lighting conditions (insufficient lighting, overexposure, etc.). Given the limitations and continuity, poor stability, and limited tasks of a single UAV's autonomous positioning, future work will employ a hierarchical multi-UAV cluster, combine multiple sensors to perform sensor position and attitude fusion estimation research, and implement UAV cluster collaborative positioning. In a UAV swarm, each UAV broadcasts its position in real time, obtains information from other UAVs, and improves its global positioning capability by using relative measures between UAVs.

Author Contributions: Conceptualization, P.T. and X.Y.; methodology, P.T. and X.P.; writing—original draft preparation, P.T.; writing—review and editing, X.Y.; supervision, X.Y. and L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Open Fund of Laboratory of Intelligent Control, grant number ICL-2023-0401.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the author.

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

References

- Sharma, T.; Acharya, B.; Daga, H.; Sharma, A.; Srinivasan, K.; Bajpai, V.; Chang, J.R. An insight on UAV/drone autonomous navigation methods and applications: A review. *Int. J. Soc. Humanist. Comput.* **2020**, *3*, 245–269. [\[CrossRef\]](#)
- Tong, P.; Yang, X.; Yang, Y.; Liu, W.; Wu, P. Multi-UAV Collaborative Absolute Vision Positioning and Navigation: A Survey and Discussion. *Drones* **2023**, *7*, 261. [\[CrossRef\]](#)
- Chen, S.; Yin, D.; Niu, Y. A survey of robot swarms' relative localization method. *Sensors* **2022**, *22*, 4424. [\[CrossRef\]](#) [\[PubMed\]](#)
- Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [\[CrossRef\]](#)
- Huai, Z.; Huang, G. Robocentric visual-inertial odometry. *Int. J. Robot. Res.* **2022**, *41*, 667–689. [\[CrossRef\]](#)
- Delaune, J.; Bayard, D.S.; Brockers, R. Range-visual-inertial odometry: Scale observability without excitation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2421–2428. [\[CrossRef\]](#)

7. Leprince, S.; Barbot, S.; Ayoub, F.; Avouac, J.P. Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1529–1558. [[CrossRef](#)]
8. Couturier, A.; Akhloufi, M.A. Relative visual localization (RVL) for UAV navigation. In *Degraded Environments: Sensing, Processing, and Display 2018*; SPIE: Bellingham, WA, USA, 2018; Volume 10642, pp. 213–226.
9. Couturier, A.; Akhloufi, M.A. UAV navigation in GPS-denied environment using particle filtered RVL. In *Situation Awareness in Degraded Environments 2019*; SPIE: Bellingham, WA, USA, 2019; Volume 11019, pp. 188–198.
10. Couturier, A.; Akhloufi, M. Conditional probabilistic relative visual localization for unmanned aerial vehicles. In Proceedings of the 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), London, ON, Canada, 30 August–2 September 2020; pp. 1–4.
11. Warren, M.; Greeff, M.; Patel, B.; Collier, J.; Schoellig, A.P.; Barfoot, T.D. There’s no place like home: Visual teach and repeat for emergency return of multirotor uavs during gps failure. *IEEE Robot. Autom. Lett.* **2018**, *4*, 161–168. [[CrossRef](#)]
12. Van Dalen, G.J.; Magree, D.P.; Johnson, E.N. Absolute localization using image alignment and particle filtering. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016; p. 647.
13. Magree, D.P.; Johnson, E.N. A monocular vision-aided inertial navigation system with improved numerical stability. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 5–9 January 2015; p. 97.
14. Gyagenda, N.; Hatilima, J.V.; Roth, H.; Zhmud, V. A review of GNSS-independent UAV navigation techniques. *Robot. Auton. Syst.* **2022**, *152*, 104069. [[CrossRef](#)]
15. Natal, J.; Ávila, I.; Tsukahara, V.B.; Pinheiro, M.; Maciel, C.D. Entropy: From thermodynamics to information processing. *Entropy* **2021**, *23*, 1340. [[CrossRef](#)]
16. Ali, A.; Naeem, S.; Anam, S.; Ahmed, M.M. Entropy in information theory from many perspectives and various mathematical models. *J. Appl. Emerg. Sci.* **2022**, *12*, 156–165.
17. Wan, X.; Liu, J.; Yan, H.; Morgan, G.L. Illumination-invariant image matching for autonomous UAV localisation based on optical sensing. *ISPRS J. Photogramm. Remote Sens.* **2016**, *119*, 198–213. [[CrossRef](#)]
18. Keller, Y.; Averbuch, A. A projection-based extension to phase correlation image alignment. *Signal Process.* **2007**, *87*, 124–133. [[CrossRef](#)]
19. Patel, B. Visual Localization for UAVs in Outdoor GPS-Denied Environments. Master’s Thesis, University of Toronto (Canada), Toronto, ON, Canada, 2019.
20. Yol, A.; Delabarre, B.; Dame, A.; Dartois, J.É.; Marchand, E. Vision-based absolute localization for unmanned aerial vehicles. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3429–3434.
21. Pascoe, G.; Maddern, W.P.; Newman, P. Robust direct visual localization using normalised information distance. In Proceedings of the British Machine Vision Conference, Oxford, UK, 21–24 November 2015; pp. 70–71.
22. Lu, C.; Qi, X.; Ding, K.; Yu, B. An improved FAST algorithm based on image edges for complex environment. *Sensors* **2022**, *22*, 7127. [[CrossRef](#)] [[PubMed](#)]
23. Tang, G.; Liu, Z.; Xiong, J. Distinctive image features from illumination and scale invariant keypoints. *Multimed. Tools Appl.* **2019**, *78*, 23415–23442. [[CrossRef](#)]
24. Fatima, B.; Ghafoor, A.; Ali, S.S.; Riaz, M.M. FAST, BRIEF and SIFT based image copy-move forgery detection technique. *Multimed. Tools Appl.* **2022**, *81*, 43805–43819. [[CrossRef](#)]
25. Seema, B.S.; Hemanth, K.; Naidu, V.P.S. Geo-registration of aerial images using RANSAC algorithm. In *NCTAESD-2014*; Vemana Institute of Technology: Bangalore, India, 2014; pp. 1–5.
26. Saranya, K.C.; Naidu, V.P.S.; Singhal, V.; Tanuja, B.M. Application of vision based techniques for UAV position estimation. In Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), Bangalore, India, 6–7 May 2016; pp. 1–5.
27. Shan, M.; Wang, F.; Lin, F.; Gao, Z.; Tang, Y.Z.; Chen, B.M. Google map aided visual navigation for UAVs in GPS-denied environment. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 114–119.
28. Chiu, H.P.; Das, A.; Miller, P.; Samarasekera, S.; Kumar, R. Precise vision-aided aerial navigation. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 688–695.
29. Mantelli, M.; Pittol, D.; Neuland, R.; Ribacki, A.; Maffei, R.; Jorge, V.; Prestes, E.; Kolberg, M. A novel measurement model based on abBRIEF for global localization of a UAV over satellite images. *Robot. Auton. Syst.* **2019**, *112*, 304–319. [[CrossRef](#)]
30. Masselli, A.; Hanten, R.; Zell, A. Localization of unmanned aerial vehicles using terrain classification from aerial images. In *Advances in Intelligent Systems and Computing*; Intelligent Autonomous Systems 13; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 831–842.

31. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
32. Goforth, H.; Lucey, S. GPS-denied UAV localization using pre-existing satellite imagery. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2974–2980.
33. Anderson, S.; Barfoot, T.D. Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3). In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 157–164.
34. Sola, J. Quaternion kinematics for the error-state Kalman filter. *arXiv* **2017**, arXiv:1711.02508.
35. Wang, J.X.; Wang, W.X.; Wang, C.Y.; Zhu, H.; He, W.Y.; Liu, S.Y. Line segment matching algorithm based on feature grouping and LBD descriptor. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 103–109. [[CrossRef](#)]
36. Jiang, S.; Jiang, C.; Jiang, W. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 230–251. [[CrossRef](#)]
37. Zheng, Y.S.; Jin, Y.W.; Dong, Y. Rail detection based on LSD and the least square curve fitting. *Int. J. Autom. Comput.* **2021**, *18*, 85–95. [[CrossRef](#)]
38. Zhao, S.; Zhang, T.; Wei, H. Fast visual inertial odometry with point–line features using adaptive EDLines algorithm. *Meas. Sci. Technol.* **2022**, *33*, 105401. [[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.