Hindawi
Complexity
Volume 2019, Article ID 8176489, 11 pages
https://doi.org/10.1155/2019/8176489

Research Article

Passive Initialization Method Based on Motion Characteristics for Monocular SLAM

Yu Yang, Jing Xiong, Xiaoyu She, Chang Liu, ChengWei Yang, and Jie Li

School of Mechatronical Engineering, Beijing Institute of Technology, 5th South Zhongguancun Street, Beijing, China

Correspondence should be addressed to ChengWei Yang; yangchengwei2009@126.com

Received 12 October 2018; Accepted 23 January 2019; Published 5 February 2019

Guest Editor: Jungong Han

Copyright © 2019 Yu Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Visual SLAM techniques have proven to be effective methods for estimating robust position and attitude in the field of robotics. However, current monocular SLAM algorithms cannot guarantee timeliness of system startup due to the problematic initialization time and the low success rates. This paper introduces a rectilinear platform motion hypothesis and thereby converts the estimation problem into a verification problem to achieve fast monocular SLAM initialization. The proposed method is simulation tested on a fixed-wing UAV. Tests show that the proposed method can produce faster initialization of visual SLAM and that the advantages are more profound on systems with sparse image features.

1. Introduction

In recent years, with the application of Graph-based Optimization [1] and Bundle Adjustment (BA) [2] in Visual Simultaneous Localization and Mapping (vSLAM) and the emergence of excellent open-source libraries [3, 4], vSLAM systems are increasingly used in autonomous motion platforms. Exceptional open-source vSLAM systems also help popularize vSLAM techniques. Presently, vSLAM systems have been applied to UAV autonomous navigation [5, 6] and obstacle avoidance [7, 8] problems in GPS-denied environments. However, current vSLAM systems usually take a long time to initialize [9], posing difficulties for real-world engineering problems.

Currently, there exist many powerful vSLAM methods, such as PTAM [10], ORB-SLAM [11, 12] SVO [13, 14], and semidirect LSD-SLAM [15] and DSO [16]. Their initialization methods are summarized in Table 1.

Generally speaking, feature-based vSLAM techniques rely on epipolar geometry constraints or homography constraints [17]; they obtain the **R** and **t** corresponding to the minimum Reprojection Error with RANSAC or Least Squares methods. As for direct methods, they are usually initialized through randomized approaches, as exact point-to-point mappings cannot be obtained directly, leading to noncomputable **R** and **t**.

As can be seen from the above method, most of the classical monocular vision SLAM method does not consider the motion characteristics of the platform during the initialization phase. However, the basic equations of the SLAM system are composed of equations of motion and observation equations. Most of the current research focuses on the observation equations. This paper believes that the reasonable introduction of motion hypothesis can effectively improve the robustness of observations, especially in the initialization phase.

PTAM's initialization works with the hypothesis that captured images are mainly composed of flat surfaces; initial camera motion **R** and **t** are then computed with homography matrix (**H**) accordingly. ORB-SLAM algorithms are effective extensions of PTAM that compute computing essential matrix (**E**) and H simultaneously; the final initialization method is then selected by comparing the respective scores. LSD-SLAM and DSO, as direct methods, cannot compute **R** and **t** through Reprojection. Therefore, they initialize through random variables. When camera motions cover enough distance, initialization will be effectuated by locking into specific depths. SVO's initialization is similar to that of PTAM, except that SVO integrates an additional assumption that the motion direction is perpendicular to the photographed plane, as SVO is originally designed for rotor UAV use.

Table 1: Summary of initialization methods.

	Method	Initialization	Main Approach	Category
1	PTAM	H	Feature-based	SLAM
2	ORB-SLAM	H+E	Feature-based	SLAM
3	LSD-SLAM	Rand	Direct	SLAM
4	DSO	Rand	Direct	VO
5	SVO	H	Feature-based+Direct	VO

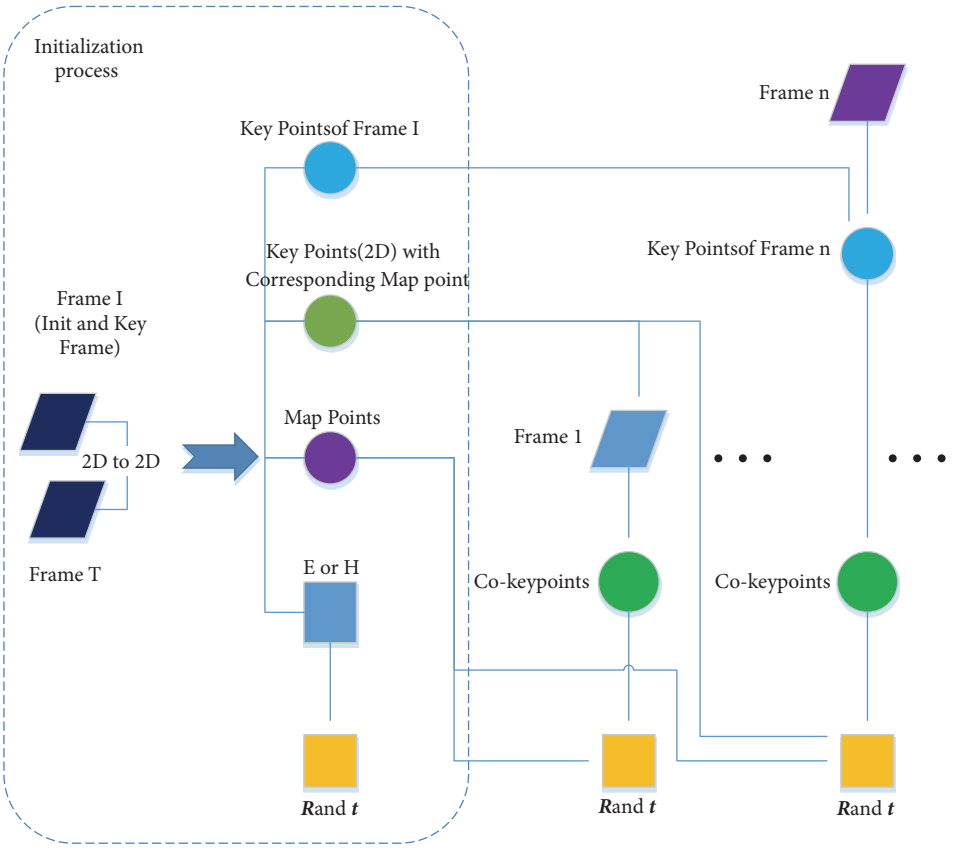


Figure 1: Classic feature-based initialization workflows.

These five classic methods are renowned in the field of monocular SLAM/VO, each possessing unique strengths. They have been successfully applied in their respective environments with satisfactory performance. The initialization workflows of the feature-based algorithms are summarized in Figure 1.

Theoretically, the initialization process depicted herein can initialize any movement except pure rotation. Firstly, corresponding points from separate frames are identified through feature-based or optical flow methods. These point mappings are then utilized along with monocular-camera imaging characteristics in computing **H** or **E** under the epipolar geometry frame. **H** or **E** is then decomposed to produce **R**, t, and finally the initial map points with the additional assumption that the mapped points contain no actual movement. This concludes the traditional initialization process, where the frames can be adjacent or nonadjacent, and the decomposition utilizes RANSAC, Eight-Point Method, or Bundle Adjustment. Subsequent processes will use the initial **R**, **t** and map points (3D) for the chain processes maintaining the monocular SLAM system. Due to the scale uncertainty of the monocular visual SLAM system, no initialization method can produce real-world distance of the map points; the dimensionless depths are provided instead. The initial map points (3D) play an important role for subsequent frames. The indirect 3Dto2D correspondences between pixel points and map points (3D), together with the geocalculated DLT/P3P [18]/EPnP [19]/UPnP [20] or the optimized BA, are used to determine the subsequent frames' positions and orientations relative to the preceding key frame. Frame I is an initialization frame as well as a key frame. As the camera moves on, the number of indirect 3Dto2D correspondences that can be established will gradually decrease, leading to probable failure of the aforementioned chain processes. It is then necessary to consider inserting new key frames to replenish the map points (3D) needed for the chain processes. Complete SLAM

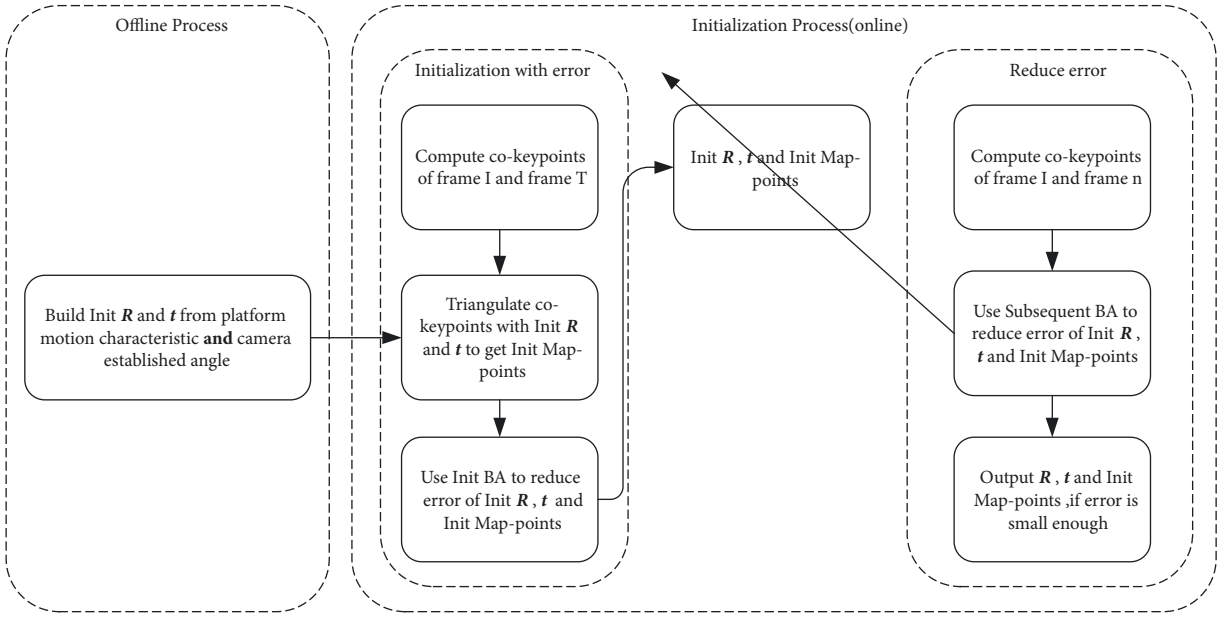


Figure 2: Proposed method flowchart.

algorithms also involve another important process termed loop closure, which will not be discussed further, as it is not much related to the present paper.

It can be seen from above that the **E** or **H** is obtained from point correspondences is the initial enabler of the entire monocular SLAM system. However, when point correspondences are insufficient or too inaccurate, the obtained **E** or **H** may contain large errors, affecting the accuracy of the map points (3D), and thus compromising the subsequent processes. Current methods are based on limited errors of **R** and **t**. Therefore, the actual implementations contain much strict computation on **E** or **H**, leading to low success rates in many cases. When the monocular SLAM systems are applied to fixed-wing unmanned aerial vehicles (UAVs), the initialization success rates are even more worrying [5, 6].

In this paper, we add a generalized motion characteristic hypothesis in the initialization process to transform the solution of camera motion R and t into the error elimination problem during the initialization process. In this way, the success rate of initialization is increased. In view of the error caused by the hypothesis, this paper reduces the error by multiframe optimization method, thus improving the accuracy of the initialization process.

1.1. Contribution. Firstly, this paper proposes the platform motion characteristics, which represents the motion state of the platform in most of the time. Secondly, this paper introduces the platform motion characteristics into the initialization phase of monocular vision SLAM and avoids the solution of the essential matrix and the homography matrix by optimization. Finally, this paper uses the subsequent BA to convert the initialization from a transient process to a convergent process of several consecutive frames.

2. Monocular SLAM Initialization Method Based on Platform Motion Characteristics and Optimization

2.1. Overview. The present paper proposes a monocular SLAM initialization method based on platform motion characteristics and optimization, the flowchart of which is shown in Figure 2.

The proposed method contains an offline process and an online process. In the offline process, initial motions **R** and **t** are computed with platform motion characteristics of the camera installation mode. In the online process, firstly, a set of Frame I and Frame T are used to detect and match the feature points, and then initial map points are generated under the initial motion hypothesis, whose initial errors are eliminated by Init BA. Finally, the subsequent BA is performed with the matched feature points of Frame I and Frame n, further reducing the errors of **R**, **t**, and the map points. The initialization is considered to have succeeded when the errors converge.

2.2. Initial Motion Hypothesis Combining Platform Motion Characteristics and Camera Installation. The proposed method utilizes an initial motion hypothesis to initialize the system. Cars on ground generally run along straight lines, while aerial vehicles usually fly at a fixed angle of attack. This is a very broad description, as ground vehicles may turn, and aircraft may roll. General motion characteristics can be expressed with

Require: Frame I:Initialization Frame
Frame T:Target Frame
T_{mc}:Camera Motion Hypothesis
Ensure: Initialized Map Points $\mathbf{X}_{init}$,$\mathbf{R}_{init}$,$\mathbf{t}_{init}$
(1) Perform feature point matching for Frame I and Frame T, resulting in matched points $\mathbf{x_I}$ and $\mathbf{x_T}$.
(2) Triangulate $\mathbf{x_I}$ and $\mathbf{x_T}$ using hypothesis T_{mc} to generate map points $\mathbf{X}_{init}$
(3) Optimize $\mathbf{X}_{init}$, $\mathbf{R}_{init}$, $\mathbf{t}_{init}$ with Init BA and update accordingly.
(4) **return** $\mathbf{X}_{init}$ $\mathbf{R}_{init}$ $\mathbf{t}_{init}$

Algorithm 1: Monocular SLAM initialization with initial motion hypothesis.

$$T_m = \begin{bmatrix} R_m & t_m \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_m \cos\phi_m & \sin\psi_m \sin\theta_m \cos\phi_m - \cos\psi_m \sin\phi_m & \cos\psi_m \sin\theta_m \cos\phi_m + \sin\psi_m \sin\phi_m & x_m \\ \cos\theta_m \sin\phi_m & \sin\psi_m \sin\theta_m \sin\phi_m + \cos\psi_m \cos\phi_m & \cos\psi_m \sin\theta_m \sin\phi_m - \sin\psi_m \cos\phi_m & y_m \\ -\sin\theta_m & \sin\phi_m \cos\theta_m & \cos\psi_m \cos\theta_m & z_m \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Equation (1) is a 6-DOF rectilinear description of any motion platform. For the monocular vSLAM initialization, the rectilinear hypothesis of the platform motion needs to be expressed in the coordinate system of the camera. The require conversion is derived from the mounting characteristics of the camera and is expressed with

$$T_{mc} = T_m T_{vc} \tag{2}$$

T_{vc} in (2) is the transformation matrix of the camera coordinate system with respect to the platform coordinate system. This matrix can be obtained from the camera installation characteristic. A general form of T_{vc} is given in

$$T_{vc} = \begin{bmatrix} R_{vc} & t_{vc} \\ 0 & 1 \end{bmatrix} \tag{3}$$

Substitute T_m and T_vc in (2) by (1) and (3), respectively, the camera motion model under the aforementioned rectilinear hypothesis is obtained.

$$T_{mc} = \begin{bmatrix} R_m R_{vc} & R_m t_{vc} + t_m \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{mc} & t_{mc} \\ 0 & 1 \end{bmatrix} \tag{4}$$

2.3. Monocular SLAM Initialization with Initial Motion Hypothesis. The established camera motion model is then used in the implementation of the passive initialization process. In conventional initialization methods, **R** and **t** are obtained from the decomposing of the strictly computed **E** or **H**. The proposed method replaces the decomposition results **R** and **t** with $\mathbf{R}_{mc}$ and $\mathbf{t}_{mc}$ and thereby triangulates the feature points to obtain the map points and finally utilizes Init BA to reduce the errors (Algorithm 1).

The Init BA mentioned above minimalizes the Reprojection Error for the feature points of Frame I and Frame T. Let $x_1^n \in \mathbb{R}^2$ and $x_2^n \in \mathbb{R}^2$ be the coordinates of the matched feature points in Frame I and Frame T, respectively. The previously established $\mathbf{T}_{cm}$ is used to triangulate x_1^n and x_2^n to obtain map points $X_m^n \in \mathbb{R}^3$.

$$\begin{bmatrix} -\mathbf{I} & x_1 & 0 & 0 \\ 0 & 0 & -\mathbf{I} & x_2 \end{bmatrix} \begin{pmatrix} K\begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \\ K\begin{bmatrix} \mathbf{R_{mc}} & \mathbf{t_{mc}} \end{bmatrix} \end{pmatrix} \begin{pmatrix} X_m \\ 1 \end{pmatrix} = 0 \tag{5}$$

$\mathbf{T}_{cm}$ introduces the rectilinear hypothesis; therefore, it is necessary to restrain the errors in the map points' coordinates.

$$\{\mathbf{X}_{init}, \mathbf{R}_{init}, \mathbf{t}_{init}\} = \underset{\mathbf{X}_m, \mathbf{R}_{cm}, \mathbf{t}_{cm}}{\arg\min} \sum_{j=1}^{N} \left(\left\| \frac{1}{\lambda_1^j} K X_m^j - \left[x_1^j, 1\right]^T \right\|^2 + \left| \frac{1}{\lambda_2^j} K \left(R_{cm} X_m^j + t_{cm}\right) - \left[x_2^j, 1\right]^T \right\|^2 \right) \tag{6}$$

The map points can be optimized once by (6), which reduces the error caused by the hypothetical model. Due to the quality of feature point matching, only Init BA cannot make the error of X, R, and t small enough, so the method introduces the subsequent BA to further reduce the error.

Equation (6) describes the Init BA optimization of the map points. Due to the quality of the matched feature points, Init BA alone cannot reduce the errors of **X**, **R**, and **t** to an acceptable margin. The propose method utilizes the subsequent BA to further reduce the errors.

2.4. Error Reduction with Subsequent BA. Limited by the number and distribution of matching feature points, the errors contained in $\mathbf{X}_{init}^i$, $\mathbf{R}_{init}$, and $\mathbf{t}_{init}$ cannot be evaluated, so this paper introduces subsequent BA to achieve further error suppression and initialization accuracy evaluation. The main idea of subsequent BA is to optimize X, Rinit, and tinit with each subsequent frame and then decide whether to continue the subsequent optimization by judging its convergence (Algorithm 2).

Require: Frame I: Initialization Frame
Frame n: Subsequent Frame
$\mathbf{x_I}$: Feature points of Frame I
$\mathbf{x^1}, \mathbf{x^2} \ldots \mathbf{x^{n-1}}$: Feature points of previous frames corresponding to $\mathbf{x_I}$
$\mathbf{R}_1, \mathbf{R}_2 \ldots \mathbf{R}_{n-1}$: Optimized rotation matrices of previous frames
$\mathbf{t}_1, \mathbf{t}_2 \ldots \mathbf{t}_{n-1}$: Optimized translation vectors of previous frames
$\mathbf{X}_{init}$: Initialized Map Points
Ensure: Map points $\mathbf{X}_{init}$ $\mathbf{R}_n$ $\mathbf{t}_n$ and Initialization Evaluator v.
(1) Extract feature points of Frame n and match with $\mathbf{x_I}$ to obtain matched feature points $\mathbf{x^n}$.
(2) Optimize the Reprojection Error with Subsequent BA and thereby obtain optimized $\mathbf{X}_{init}$, $\mathbf{R}_n$, $\mathbf{t}_n$.
(3) Calculate current evaluator value v_n.
(4) **if** $v_n \approx v_{n-1}$ **then**
(5) Stop iteration, and return $\mathbf{X}_{init}$, $\mathbf{R}_n$, $\mathbf{t}_n$ as initialization results.
(6) **end if**
(7) **if** n is too big **then**
(8) Stop iteration, and report failure of initialization.
(9) **end if**

Algorithm 2: Error reduction with subsequent BA.

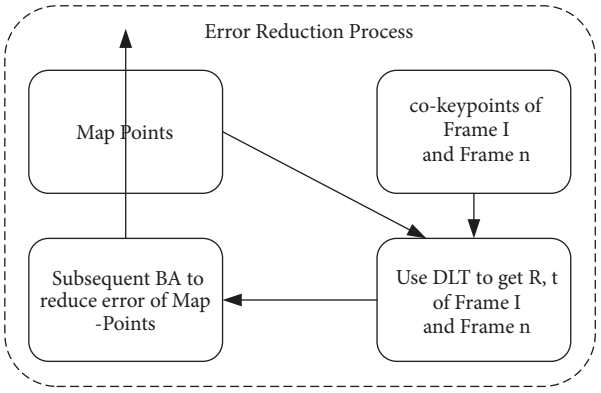


Figure 3: Error reduction process.

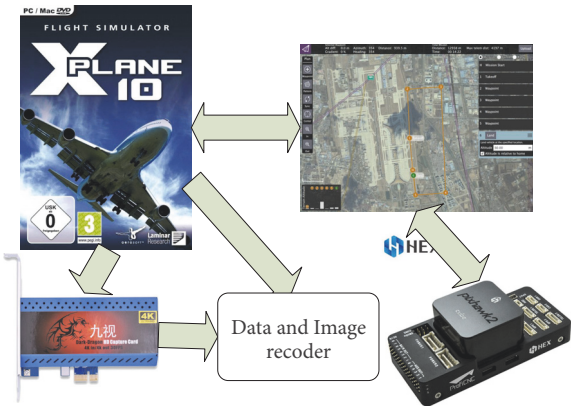


Figure 4: Simulation system.

The errors contained in $\mathbf{X}^i_{init}$, $\mathbf{R}_{init}$, and $\mathbf{t}_{init}$ cannot be evaluated through Init BA, due to the scale and distribution of the matched feature points. Subsequent BA is thus utilized for initial error evaluation and further error reduction. The main idea of subsequent BA is to optimize $\mathbf{X}^i_{init}$, $\mathbf{R}_{init}$, and $\mathbf{t}_{init}$ with each subsequent frame. Convergence evaluation is performed to determine when to stop the subsequent optimization.

For each frame of the subsequent input, the coordinate X_{init} of the map points is optimized as shown in the process of Figure 3. When it converges, the error elimination process is considered to be ended, and the subsequent BA process in the figure is as shown in (7).

For each subsequent frame, X_{init} is optimized with the error reduction process shown in Figure 3 and

$$\begin{aligned} &\left\{\mathbf{X}^i, \mathbf{R}_l, \mathbf{t}_l \mid \mathbf{X}^i \in \mathbf{X}_{init},\ l \in \mathbf{N}_f\right\} \\ &\quad = \arg\min_{\mathbf{X}^i, \mathbf{R}_l, \mathbf{t}_l} \sum_{k \in \mathbf{N}_f} \sum_{j \in \mathbf{X}_{init}} \rho\left(E\left(k, j\right)\right) \\ &E\left(k, j\right) = \left\|\mathbf{x}^j - p\left(\mathbf{R}_k \mathbf{X}^j + \mathbf{t}_k\right)\right\|_2^2 \end{aligned} \tag{7}$$

It can be seen from (7) that the scale of optimization gets larger with continuous input of subsequent frames, which ensures to some extent the reliability of the optimized results. The optimized map points are viewed as initialization results, providing input for subsequent chain processes. The quality of initialization is evaluated with

$$\begin{aligned} v &= \overline{\mathbf{E}}_o \\ \mathbf{E_o} &= \left\{e < \overline{\mathbf{E}_{all}} \mid e \in \mathbf{E}_{all}\right\} \end{aligned} \tag{8}$$

where $\mathbf{E}_{all}$ is the sum of the Reprojection Errors of all map points in all frames participating in the optimization.

3. Simulation

3.1. Simulation System. In order to better reproduce vehicle motion characteristics, the present study builds a hardware-in-the-loop (HIL) simulation system, as illustrated in Figure 4. It consists of four parts, namely, the Xplane10 flight simulation software, the Pixhawk2 flight controller, the

Table 2: Self-evaluation performance indicators.

	Indicator Name	Unit	Alias
1	Number of Convergence Frames	frame	NCF
2	Initial Error	deg	IE
3	Convergence Error	deg	CE
4	Average Number of Convergence Frames	frame	ANCF
5	Average Initial Error	deg	AIE
6	Average Convergence Error	deg	ACE

Table 3: Comparative-evaluation performance indicators.

	Indicator Name	Unit	Alias
1	Success Rate of Initialization	percentage	SRI
2	Average Error of Initialization	deg	AEI

QGroundControl software, and the data logger. Xplane10 and QGroundControl run on PC (CPU: Intel i7-7700K 4.20GHz, graphics card: NVidia GTX 1080 8G, memory: 32GB). The Pixhawk2 controller is linked to PC via a USB port.

Xplane10 plays the most important role in the entire simulation system, providing aircraft models and simulation images. The Pixhawk2 controller performs autonomous control of the fixed-wing aircraft in Xplane10, with QGroundControl acting as the data relay. Specifically, Xplane10 sends the aircraft states to QGroundControl through local loopback UDP; QGroundControl forwards the aircraft data to Pixhawk2 via the USB port using the Mavlink protocol; Pixhawk2 sends out the control commands through the same protocols. The data logger records the uplink-downlink data through UDP and the first-person view (FPV) simulation images through the video capture card.

Data transmitted in the simulation system can be roughly classified as periodic data and sporadic data. Periodic data includes the control commands and the aircraft states. Sporadic data includes the start signal, the waypoint-planning instruction, etc. Through meticulous testing, the frequency of the periodic commands is set at 65HZ, and the image sampling frequency is set at 25HZ.

3.2. Performance Indicators. Reasonable and balanced performance indicators are needed to evaluate the initialization methods. This paper proposes two groups of performance indicators, for self-evaluation (Table 2) and comparative evaluation (Table 3), respectively.

This study holds that the convergence frame number and the initial error are key indicators to assess the proposed passive initialization algorithm. As the initial error is only affected by the aircraft state at the initial time and the rectilinear motion hypothesis, the convergence frame number is a stronger indicator of the usability of the proposed method.

The optimized map points and pose information are only usable after convergence. Since error rotation e_R is quite unintuitive, for ease of understanding, this paper decomposes e_R into e_{pitch}, e_{roll}, e_{yaw} to facilitate the evaluation of performance. The difference in length between $\mathbf{t}_{init}$ and $\mathbf{t}_T$ (true value of $\mathbf{t}_{init}$) is not considered due to the depth uncertainty of monocular vSLAM initializations; only the difference in angle between $\mathbf{t}_{init}$ and $\mathbf{t}_T$ is considered.

3.3. Test Design. In order to thoroughly test the proposed initialization method, we devise a simple self-evaluation test and an advanced comparative-evaluation test. The self-evaluation test measures the inherent capabilities of the new method, while the comparative-evaluation test runs the competing algorithms on different terrains. The test scenarios include: taxing, climbing, level flight, BTT turn, diving, and landing.

3.4. Self-Evaluation Test and Result Analysis. The test results of the algorithm in this paper are shown in Figure 6, and the convergence curve of the algorithm is given, where Figure 6(a) gives the convergence curve for initializing in the running state. It can be seen that in this state, the initial error is small, because the state of motion of the aircraft is very close to the motion assumption in the slipping state, and the error is within 1° even if the error is not eliminated. Figure 6(b) gives the convergence curve for the initialization of the aircraft at the moment of takeoff. It can be seen that there is a large error in the motion state of the aircraft and the motion assumption at this time. Due to the characteristics of the fixed-wing aircraft, it is mostly in a level flight during the cruise flight. In this state, the motion of the aircraft is similar to the motion assumption. Therefore, several special states are selected in the test, including the climbing to level flight (Figure 6(d)), level flight to BTT turn (Figure 6(e)), level flight to dive (Figure 6(f)), etc. Thus Figure 6 gives the convergence of the algorithm in each typical state during a complete flight, which does not reflect the ability of the algorithm in the whole process.

Figures 7 and 8 and Table 4 summarize the convergence-related initialization performance at different poses throughout one complete flight. Figure 7 shows the convergence time statistics of the algorithm in the whole flight process. Figure 8 indicates the error distribution of the algorithm under different thresholds. Table 4 gives the exact values of Figures 7 and 8.

3.5. Comparative-Evaluation Test and Result Analysis. This paper selects ORB-SLAM2 and DSO as the competing classical algorithms for the comparative-evaluation test. In order to better reflect their performance, this study runs the all methods on plain terrain (Figure 5) and mountainous

Table 4: Convergence statistics.

	δ_s	ANCF	AIE	ACE
1	1°	1.49	95.4%	0.83
2	0.7°	1.65	83.5%	0.58
3	0.5°	2.81	78.1%	0.41
4	0.3°	3.94	67.5%	0.25
5	0.1°	5.63	63.3%	0.08

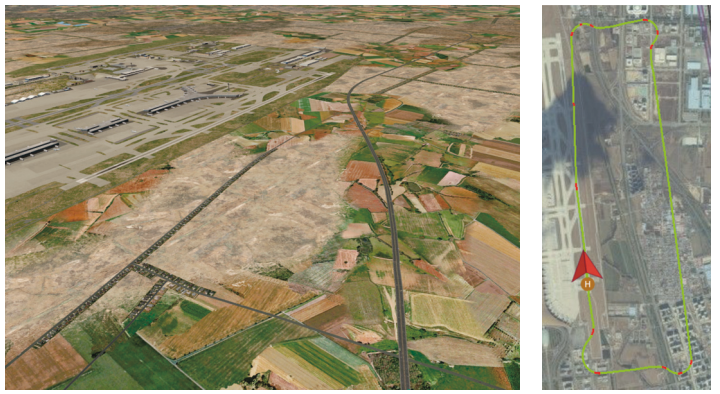


Figure 5: Simulation scenarios on plain terrain.

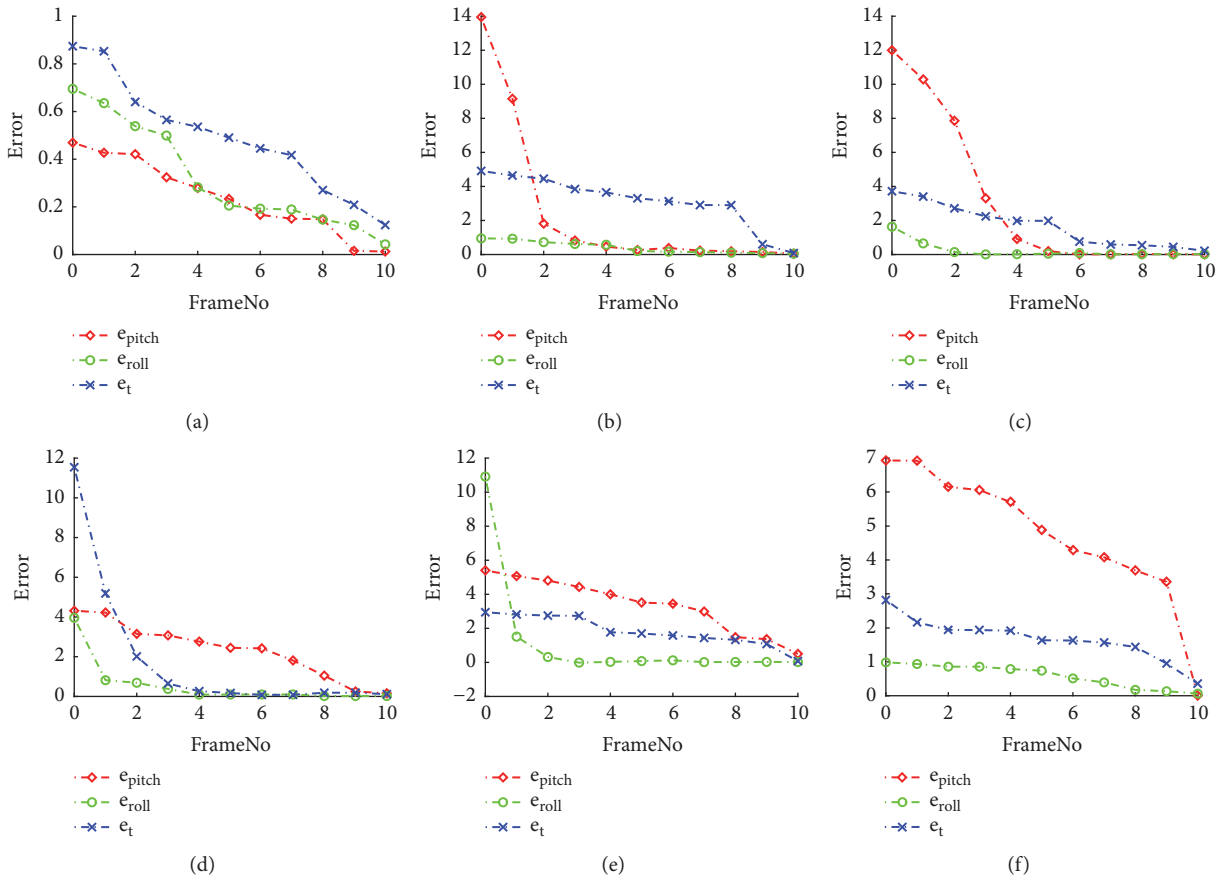


Figure 6: Error time histories.

Table 5: Initialization results under different terrains.

	Method	Terrain	SRI	AEI
1	ours (δ_s = 1)	Plain	95.4%	0.83
2	ours (δ_s = 0.7)	Plain	83.5%	0.58
3	ours (δ_s = 0.5)	Plain	78.1%	0.41
4	ours (δ_s = 0.3)	Plain	67.5%	0.25
5	ours (δ_s = 0.1)	Plain	63.3%	0.08
6	ORB-SLAM2	Plain	8.2%	1.67
7	DSO	Plain	12.2%	1.01
8	ours (δ_s = 1)	Mountainous	23.1%	0.93
9	ours (δ_s = 0.7)	Mountainous	17.1%	0.65
10	ours (δ_s = 0.5)	Mountainous	14.9%	0.44
11	ours (δ_s = 0.3)	Mountainous	11.1%	0.24
12	ours (δ_s = 0.1)	Mountainous	9.1%	0.08
13	ORB-SLAM2	Mountainous	5.4%	1.78
14	DSO	Mountainous	10.4%	1.38

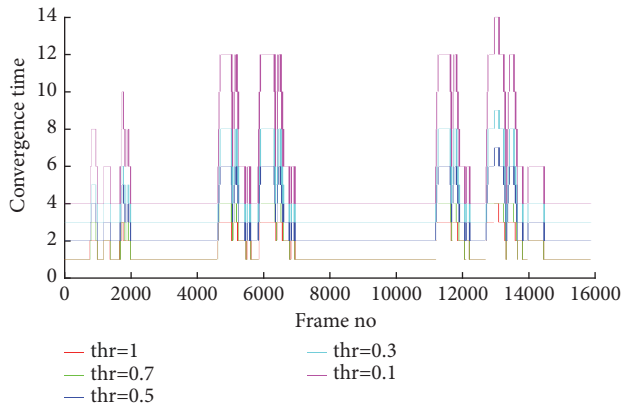


Figure 7: Convergence time histories.

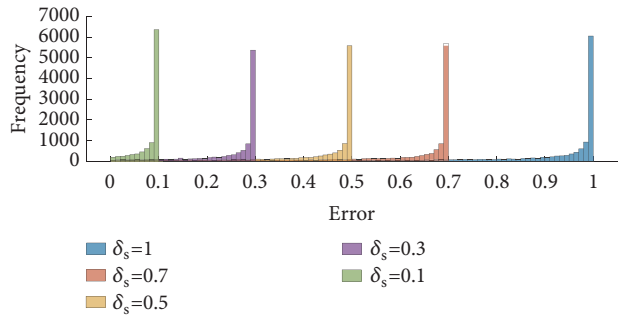


Figure 8: Converged error statistics.

terrain (Figure 9). Considering the stochastic nature of ORB-SLAM2, the study conducts five comparative-evaluation subtests on each terrain. The best subtest results are viewed as the illustrative test results.

Figure 10 gives the initialization results of the three algorithms in two terrains. δ_s in proposed method is set to 0.5 during test. It can be seen that SRI of the proposed method in both plain and hilly terrain is greater than that of ORB-SLAM2 or DSO. Considering the effect of δ_s on proposed method, SRI under different δ_s is compared, as in the Table 5

In addition to the comparative-evaluation performance indicators introduced in Table 3, this paper also compares the number of matched feature points(ANMFP) needed by ORB-SLAM2, DSO and the proposed method, respectively, to effectuate successful initialization (Table 6).

It can be seen from Table 6 that the ANMFP value of the proposed algorithm is between 50 and 70, while the ANMFP of ORB-SLAM2 is above 200. The DSO algorithm requires a larger ANMFP, because it uses a direct method framework. It can be seen that the number of feature points required by the proposed algorithm is much smaller than that of ORB-SLAM2 and DSO. The reason for this result is determined by the basic structure of the algorithm in this paper. The algorithm does not directly calculate the **H** or **E** by relying on the correspondence between the feature points of two adjacent frames, but continuously optimizes the initial pose by using the feature point correspondences that can be continuously observed in successive frames. That is to say, for this method, there is no need to have so many feature points in the initial frame. This method could get an acceptable initial attitude as long as enough points can be continuously observed in successive frames. This also explains from another side why the algorithm can achieve a higher SRI. Therefore, the method in this paper can achieve better results from little feature points when dealing with sparse image features.

4. Conclusion

In this paper, we propose a rectilinear hypothesis of platform motion and thereby derive a passive initialization method for monocular SLAM. Init BA and subsequent BA are utilized in reducing the errors between the actual motion and that of the proposed hypothesis. A simulated fixed-wing aircraft is selected as the test platform for the proposed method. Results show that the success rate of monocular SLAM Initialization

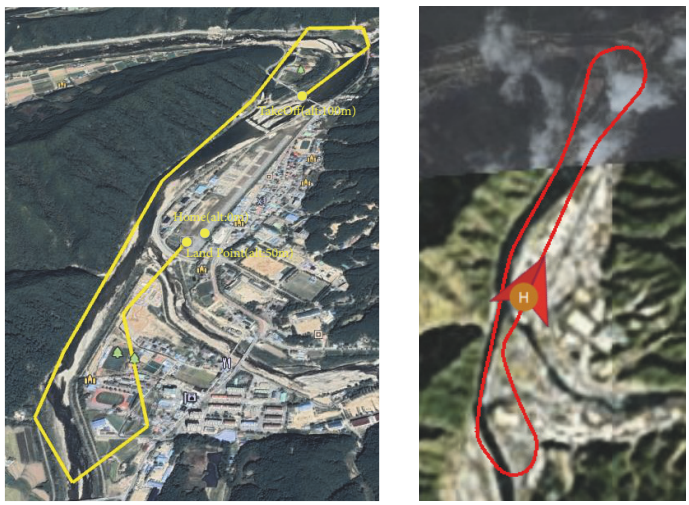


Figure 9: Simulation scenarios on mountainous terrain.

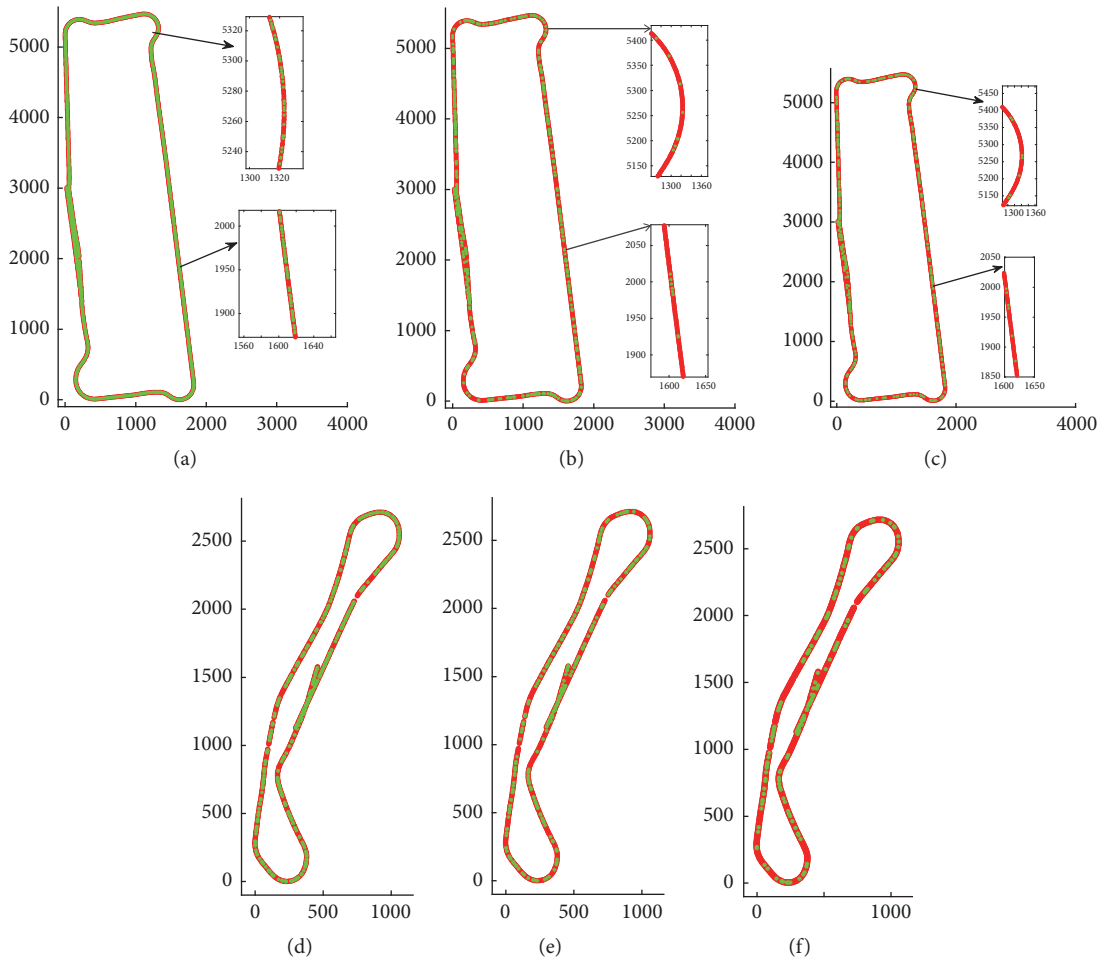


Figure 10: Initialization results comparison (green: success; red: failed), (a) plain (ours), (b) plain (ORB-SLAM2), (c) plain (DSO), (d) mountainous (ours), (e) mountainous (ORB-SLAM2), and (f) mountainous (DSO).

Table 6: Number of matched feature points needed.

	method	Terrain	ANMFP
1	ours ($\delta_s = 1$)	Plain	57.1
2	ours ($\delta_s = 0.7$)	Plain	58.5
3	ours ($\delta_s = 0.5$)	Plain	56.1
4	ours ($\delta_s = 0.3$)	Plain	55.2
5	ours ($\delta_s = 0.1$)	Plain	63.3
6	ORB-SLAM2	Plain	297.3
7	DSO	Plain	1907.1
8	ours ($\delta_s = 1$)	Mountainous	78.7
9	ours ($\delta_s = 0.7$)	Mountainous	68.2
10	ours ($\delta_s = 0.5$)	Mountainous	74.9
11	ours ($\delta_s = 0.3$)	Mountainous	66.2
12	ours ($\delta_s = 0.1$)	Mountainous	71.1
13	ORB-SLAM2	Mountainous	254.8
14	DSO	Mountainous	1953.8

is greatly improved compared with that of ORB-SLAM2. However, this method is only effective on platforms with strong motion characteristics and cannot be used indiscriminately on platforms characterized by randomized motions, such as humans and animals. At present, the method has yet to be tested in real-world environments, which will be rectified in future works.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The research received no external funding. Among the authors, Yu Yang, Jing Xiong, and Xiaoyu She are graduate students of Beijing Institute of Technology. The research was performed as part of their education. Authors Jie Li, Chengwei Yang, and Chang Liu are employed by Beijing Institute of Technology. They only played a supervising role in the research.

Conflicts of Interest

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

References

[1] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[2] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, ""Bundle adjustment" a modern synthesis," in *Proceedings of the Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372, Springer, Corfu, Greece, 1999.

[3] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: a general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 3607–3613, IEEE, Shanghai, China, May 2011.

[4] S. Agarwal and K. Mierle, "Ceres solver," http://ceres-solver.org.

[5] Y. Lin, F. Gao, T. Qin et al., "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.

[6] F. Nex and F. Remondino, "UAV for 3D mapping applications: a review," *Applied Geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[7] S. Lin, M. A. Garratt, and A. J. Lambert, "Monocular vision-based real-time target recognition and tracking for autonomously landing an UAV in a cluttered shipboard environment," *Autonomous Robots*, vol. 41, no. 4, pp. 881–901, 2017.

[8] D. Scaramuzza, M. C. Achtelik, L. Doitsidis et al., "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robotics and Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[9] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg, "Global localization from monocular SLAM on a mobile phone," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 531–539, 2014.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07)*, pp. 225–234, Nara, Japan, November 2007.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[12] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: fast semi-direct monocular visual odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 15–22, IEEE, Hong Kong, June 2014.

[14] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation, ICRA 2014*, pp. 2609–2616, IEEE, June 2014.

[15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *Proceedings of the Computer Vision –ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8690, pp. 834–849, Springer International Publishing, 2014.

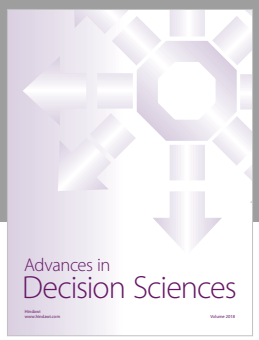
[16] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

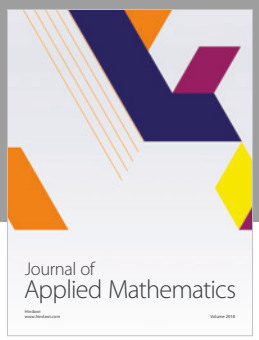
[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.

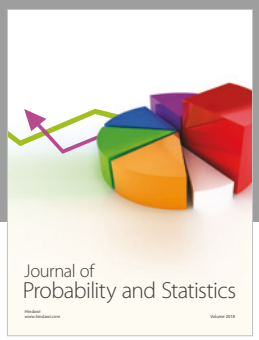
[18] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

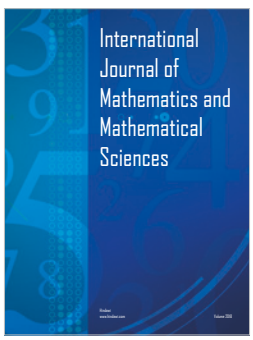
[19] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: an accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

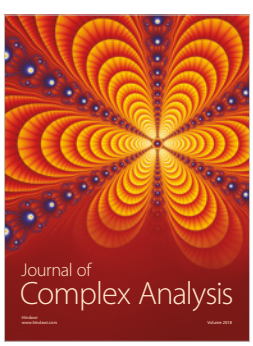
[20] L. Kneip, H. Li, and Y. Seo, "UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability," in *Proceedings of the European Conference on Computer Vision*, vol. 8689, pp. 127–142, Springer, 2014.

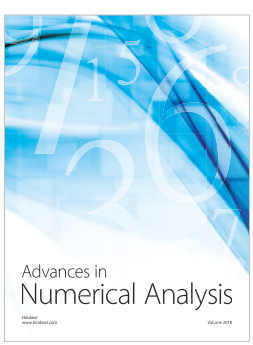


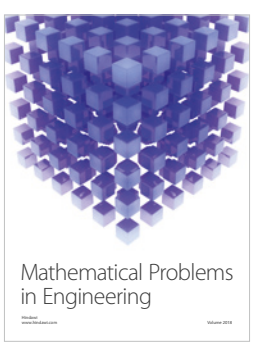


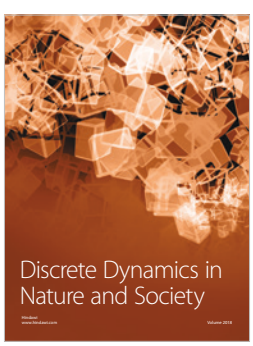


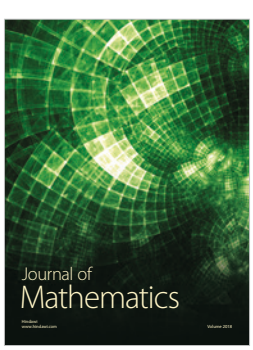


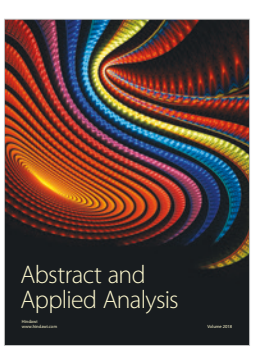


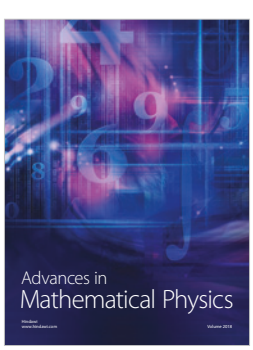












Hindawi

Submit your manuscripts at
www.hindawi.com