




# Visual odometry errors and fault distinction for integrity monitoring

Yuanwen Fu<sup>1</sup> · Shizhuang Wang<sup>1</sup> · Yawei Zhai<sup>1</sup> · Xingqun Zhan<sup>1</sup> 

Received: 19 July 2020 / Revised: 21 August 2020 / Accepted: 1 September 2020 / Published online: 14 September 2020  
© Shanghai Jiao Tong University 2020

## Abstract

Visual odometry (VO) has been widely used for many purposes in the past decade. However, the three assumptions of VO are not often met in the reality, and therefore, the uncertainty of VO output (the pose of agent) should be estimated for safety's sake, which can be done suitably by the integrity monitoring. To construct the integrity monitoring framework of VO, the first step is to establish the model of errors of measurements and calculate the fault rate, which has not been found in the literature to our knowledge. In response, this paper aims at establishing the model of errors of spatial points and calculating the fault rate in the stereo VO based on the feature point method. In this work, we describe the principle of stereo VO based on the feature point method in a deep and comprehensive way. The errors and faults of spatial points in stereo VO are defined, distinguished and classified in detail. The error propagation from pixel to spatial point is deduced, and the model of errors of spatial points is constructed. The KITTI odometry dataset is employed to evaluate the fault rate and standard deviation of errors of spatial points. And multiple sets of sensitive analyses are carried out to address the impact of RANdom SAMple Consensus (RANSAC) threshold, RANSAC iterations and operational scenario on spatial point error and fault rate. This paper could be a reference for constructing the integrity monitoring framework of stereo VO based on the feature point method.

**Keywords** Stereo vision · Localization · Visual odometry · Error model · Fault detection · Integrity monitoring

## 1 Introduction

Due to the limitations of Inertial Navigation System (INS) and Global Navigation Satellite System (GNSS), visual odometry (VO) [3, 8, 11, 12] has been proposed with great complementary significance, which is a technique that estimates incrementally the ego-motion of camera or agent only with the visual input. VO has a wide range of applications in recent years, especially in mobile robot systems, such as space rovers, aerial drones, cleaning robots and even autonomous driving. VO is reliable and accurate under favorable conditions, such as when there is sufficient illumination and texture to establish correspondence, sufficient overlap between consecutive frames, and when the majority of the visible scene is static [20]. However, these conditions are not always met in the reality, and therefore, the uncertainty of VO output (the pose of vehicle) should be estimated for sake of safety. Integrity monitoring is a good fit for this job, which is a technique that can evaluate the integrity of a navigation system.

The integrity measures the degree of trust that can be placed on the correctness of the navigation solution [9]. Many studies have focused on integrity monitoring for GNSS [19] and GNSS/IMU coupling system [13]. There are a few works about the integrity of visual-based navigation. Li and Waslander proposed a novel integrity framework for the optimization-based visual localization, which includes fault detection and exclusion (FDE) and protection level calculation [9]. However, their work assumes that there is at most one fault after FDE, which is unrealistic in the context of VO. Zhu et al. studied the hazardous misleading information (HMI) probability caused by incorrect association [21], and Wang et al. developed a two-layer fault detection scheme through combining RANdom SAMpling Consensus (RANSAC) with multiple hypotheses solution separation (MHSS) to achieve high efficiency and high reliability [18]. Both focus on the map-based visual navigation, whose error model and fault rate are significantly different from VO's.

To construct the integrity monitoring framework of stereo VO, the first step is to establish the model of errors of measurements (the spatial points in stereo VO) and calculate the fault rate, which has not been found in the literature to our knowledge. The main contributions of this paper are: first,

✉ Xingqun Zhan  
xqzhan@sjtu.edu.cn

<sup>1</sup> School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China

the error propagation from pixel to spatial point is deduced, and we find that the error of spatial point is related to the position of feature point on the image and the distance of spatial point from the camera. Second, in the foundation that errors and faults of spatial points in stereo VO are defined and distinguished in detail, we divide the spatial points into different subsets according to the position of the feature points on the image and the distance of the spatial points from the camera, and then calculate their error variances in a statistical manner.

The rest of the paper is organized as follows. Section 2 briefly introduces the workflow of stereo visual odometry based on feature point method. Section 3 discusses the model of errors of spatial points and the definition of fault. Section 4 utilizes the KITTI Odometry dataset to calculate the variance of errors of the spatial points and fault rate, and conduct a series of sensitivity analyses on RANSAC [4] threshold, RANSAC iterations and operational scenario. Section 5 draws the conclusions.

## 2 The workflow of VO

VO aims at estimating the ego-motion of camera or agent, and the key is to find the projections of spatial points in the consecutive frames, that is, to find the correspondences of pixels in the consecutive frames. VO can be divided into feature point method and direct method according to different criteria of correspondence [16]. The feature point method utilizes feature vector as criterion of correspondence, and feature vector is carefully designed by researchers. The direct method utilizes intensity as criterion of correspondence. Feature point method has always been the mainstream trend because of its better robustness and accuracy [1]. The feature point method is based on the feature points in the image, which are considered as the local features of an image. The implementation of stereo VO based on the feature point method includes the following steps: feature extraction, feature matching, depth estimation, RANSAC and motion estimation. In the feature extraction stage, there are kinds of feature points that can be selected, such as Harris, scale invariant feature transform (SIFT) [10], speeded up robust features (SURF) [2], oriented FAST and rotated BRIEF (ORB) [15], etc. ORB is chosen in this work after trading off the real-time capability and robustness of feature point. ORB feature point is composed of key point and descriptor. The key point stores the location of the feature point in the image. The descriptor records the local feature around the feature point, which can be considered as the unique identification of the feature point [15]. The feature matching is to find the projections of same spatial point in the consecutive frames by taking the two closest descriptors of Hamming distance. There is an important parameter (the Hamming distance threshold) at this stage. It

is taken based on empirical regularity. When the distance of two descriptors is less than the Hamming distance threshold, the two corresponding feature points are considered to be the projections of the same spatial point.

Before introducing the principle of depth estimation, we first introduce the pinhole camera model for a clear explanation.

$$\frac{X}{Z} = \frac{u - c_x}{f_x}, \quad (1)$$

$$\frac{Y}{Z} = \frac{v - c_y}{f_y}, \quad (2)$$

where  $X, Y, Z$  are the coordinates of a spatial point in the camera coordinate system,  $u, v$  are the coordinates of the projection of spatial point in the pixel coordinate system, and  $c_x, c_y, f_x, f_y$  are the camera intrinsics. Obviously, Eqs. (1) and (2) alone cannot acquire the spatial coordinate of the feature point. Therefore, depth estimation is introduced and two observations  $u_l, u_r$  could be derived for the same 3D point with stereo camera. According to Eq. (1), there is

$$\frac{X_l}{Z} = \frac{u_l - c_x}{f_x}. \quad (3)$$

$$\frac{X_l - b}{Z} = \frac{u_r - c_x}{f_x}. \quad (4)$$

We subtract Eq. (4) from Eq. (3),

$$\frac{b}{Z} = \frac{u_l - u_r}{f_x}. \quad (5)$$

The disparity  $d$  is defined as,

$$d = u_l - u_r. \quad (6)$$

Therefore, Eq. (5) can be rewritten as,

$$Z = \frac{f_x b}{d}. \quad (7)$$

Hence, the key of depth estimation is to acquire the disparity. There are many ways to find the disparity  $d$ , including block matching (BM) [17], semi-global block matching (SGBM) [7], graph cut (GC), sum of absolute differences (SAD), etc. Simultaneously considering accuracy and time complexity, SGBM is employed to calculate the disparity map in this paper.

The spatial coordinates of the feature points in the camera coordinate system could be derived after depth estimation, which are known as the 3D points. Then, RANSAC is leveraged to reject outliers. The process of RANSAC in pseudo-code is shown as follows. The RANSAC iterations and the RANSAC threshold are two prominent parameters of

this algorithm. Both of these affect the accuracy of the final result to different degrees.

where  $i$  is from 1 to  $n$ . Then, we can acquire the corresponding point set  $Q = \{q_1, q_2, \dots, q_n\}$  and point set  $Q' = \{q'_1, q'_2, \dots, q'_n\}$ . Third, solve the 3 by 3 matrix  $W$ .

---

**Algorithm:** RANSAC

---

**Input:** 3Dpoints, iterations, threshold

**Output:** inliers

```

1. m = 1
2. inliers_best = {}, inliers_current = {}
3. for(int i = 0; i < iterations; i++)
4. {
5.     take four 3Dpoints to solve  $R, t$ 
6.     for(int j = 0; j < 3Dpoints.size(); j++)
7.     {
8.         errors(j) = 3Dpoints(j, 1:3)' - (R * 3Dpoints(j, 4:6)' + t)
9.         if(errors(j).norm < threshold)
10.            inliers_current(m++) = 3Dpoints(i, :)
11.     }
12.     if(inliers_best.size() < inliers_current.size())
13.         inliers_best = inliers_current
14. }
15. return inliers_best

```

---

The matched spatial point sets are obtained after RANSAC. We use them to estimate camera motion, which is represented by a rotation matrix  $R$  and a translation vector  $t$ . For such problem, iterative closest point (ICP) can solve it well. There are two main ways of ICP, one is to utilize linear algebra, mainly SVD. The other is to leverage nonlinear optimization [5]. Next, we will use the SVD method to estimate  $R$  and  $t$ . Assuming that the feature points in the current frame and the previous frame have matched, the 3D point set of the previous frame is recorded as  $P = \{p_1, p_2, \dots, p_n\}$ , and the 3D point set of the current frame is recorded as  $P' = \{p'_1, p'_2, \dots, p'_n\}$ . First, we compute the centroid of each point set as:

$$q = \frac{1}{n} \sum_{i=1}^n p_i. \quad (8)$$

$$q' = \frac{1}{n} \sum_{i=1}^n p'_i. \quad (9)$$

Second, we solve decentralized coordinates.

$$q_i = p_i - q, \quad (10)$$

$$q'_i = p'_i - q', \quad (11)$$

$$W = \sum_{i=1}^n q'_i q_i. \quad (12)$$

Fourth, perform singular value decomposition (SVD) on the matrix  $W$ .

$$W = U \Sigma V^T \quad (13)$$

Then, the optimal rotation matrix and translation vector can be expressed as [14]:

$$R^* = \begin{cases} UV^T, & |UV^T| \geq 0 \\ -UV^T, & |UV^T| < 0 \end{cases} \quad (14)$$

$$t^* = q' - Rq. \quad (15)$$

### 3 Analysis of errors and faults

In this section, we discuss the error model of 3D points, classify the error sources and derive the error propagation in VO. The error model of 3D points obviously has a great relationship with the type of feature point, based on of which, the spatial point error is modeled by recovering from ORB feature points. The errors of 3D points are settled once the imaging process is completed. We divide error sources into three categories: scenario, propagation and camera. The specific content is shown in Table 1. There are two additional explanations for this table. First, object movement in the scenario and mismatch of feature points are defined as faults

**Table 1** Error sources and faults

Categories	Sources	Impact	Error	Fault
Scenario	Illumination difference between left and right cameras	Intensity	✓	
	Illumination change in consecutive frames	Intensity	✓	✓
	Object motion	$u, v$		✓
	Lack of texture	Excluded by preprocessing		
	Similar texture	Excluded by preprocessing		
	Dense fog	Excluded by preprocessing		
Propagation	Light refraction due to water vapor	$u, v$	✓	
	Distortion correction residuals	$u, v$	✓	
	Lens blur	$u, v$ and intensity	✓	
	Shelter	Excluded by preprocessing		
	Resolution and A/D frequency	$u, v$	✓	
Camera	Thermal noise	Intensity	✓	
	Impulse noise	Intensity	✓	
	High speed motion	$u, v$ and intensity	✓	

in the VO. Second, a simple preprocessing of the image is required before performing VO. Image preprocessing refers to the exclusion of extreme scenarios, including lack of texture, similar texture, shelter, dense fog, dark night, heavy snowfall, etc.

No additional errors are introduced from feature point  $(u, v)$  in the image plane to 3D point  $(X, Y, Z)$  in the camera coordinate system, and there is only error propagation in this process. According to Eq. (7),

$$Z + \Delta Z = \frac{f_x b}{d + \Delta d}, \quad (16)$$

where  $Z, d$  are the ground-truth of depth and disparity respectively, and  $\Delta Z, \Delta d$  are the measurement errors. We subtract Eq. (7) from Eq. (16),

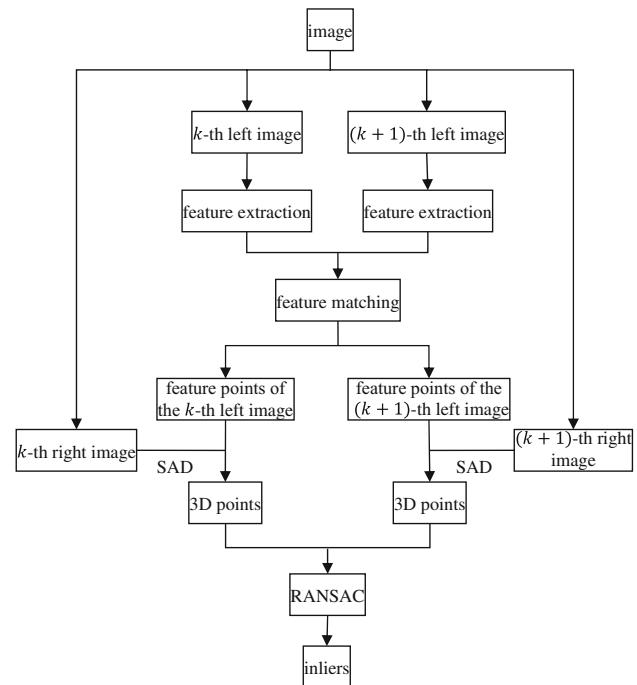
$$\Delta Z = f_x b \frac{\Delta d}{(d + \Delta d)d} \approx \frac{f_x b}{d^2} \Delta d = \frac{Z^2}{f_x b} \Delta d. \quad (17)$$

Therefore, the variance of the  $\Delta Z$  can be expressed as,

$$\sigma^2(\Delta Z) = \left( \frac{Z^2}{f_x b} \right)^2 \sigma^2(\Delta d). \quad (18)$$

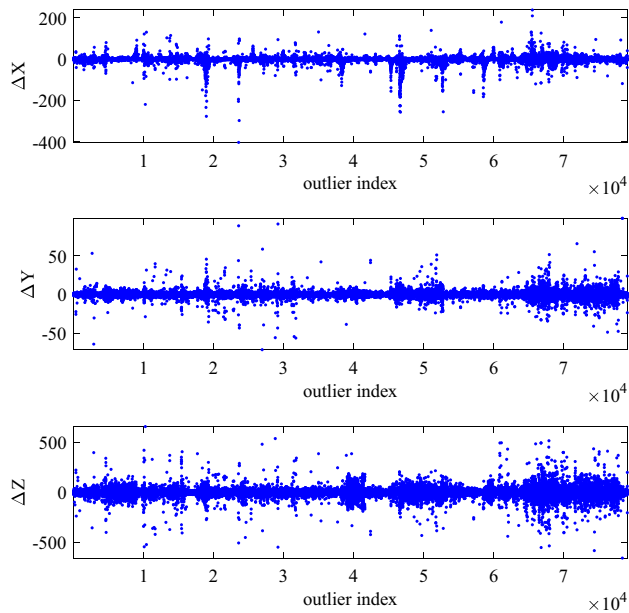
According to Eqs. (1) and (7),

$$X + \Delta X = \frac{(u + \Delta u) - c_x}{f_x} (Z + \Delta Z), \quad (19)$$

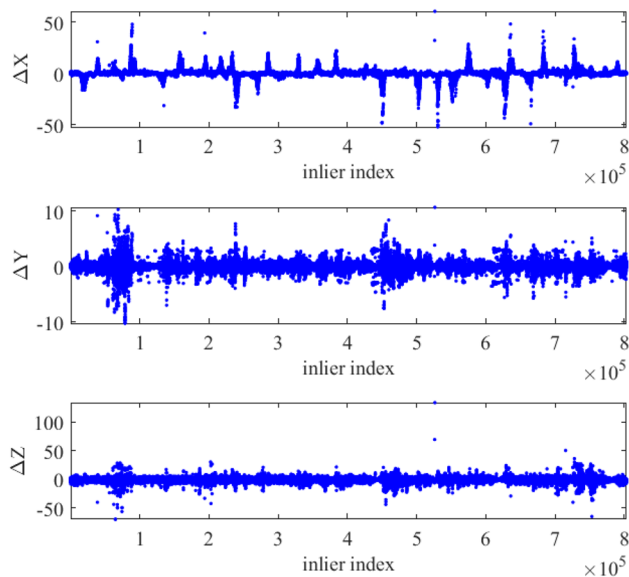
**Fig. 1** Extracting inliers flow chart

where  $u, Z$  are the ground-truth, and  $\Delta u, \Delta Z$  are measurement errors. We subtract the ground-truth of  $X$  from Eq. (19),

$$\Delta X = \frac{Z}{f_x} \cdot \Delta u + \frac{(u + \Delta u) - c_x}{f_x} \cdot \Delta Z. \quad (20)$$



**Fig. 2** The error of outliers



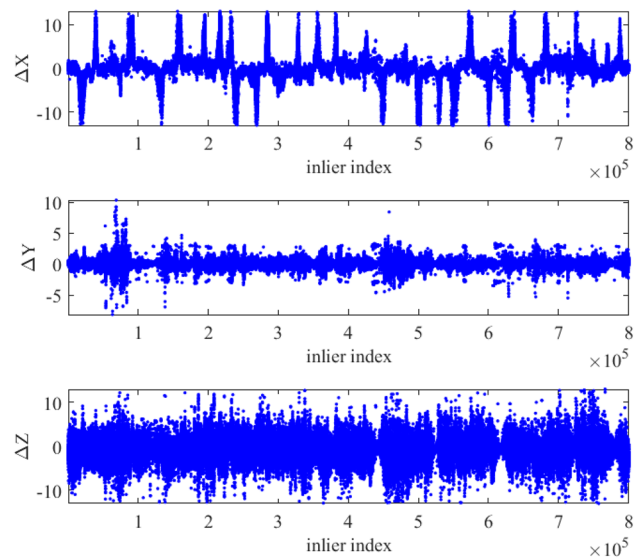
**Fig. 3** The error of inliers

According to the law of covariance propagation,

$$\sigma^2(\Delta X) = \left(\frac{Z}{f_x}\right)^2 \sigma^2(\Delta u) + \left(\frac{(u + \Delta u) - c_x}{f_x}\right)^2 \sigma^2(\Delta Z). \quad (21)$$

According to Eqs. (2) and (7),

$$\Delta Y = \frac{Z}{f_y} \cdot \Delta v + \frac{(v + \Delta v) - c_y}{f_y} \cdot \Delta Z, \quad (22)$$



**Fig. 4** The error of inliers after fault detection and exclusion

where  $v, Z$  are the ground-truth, and  $\Delta v, \Delta Z$  are the measurement errors. According to the law of covariance propagation,

$$\sigma^2(\Delta Y) = \left(\frac{Z}{f_y}\right)^2 \sigma^2(\Delta v) + \left(\frac{(v + \Delta v) - c_y}{f_y}\right)^2 \sigma^2(\Delta Z). \quad (23)$$

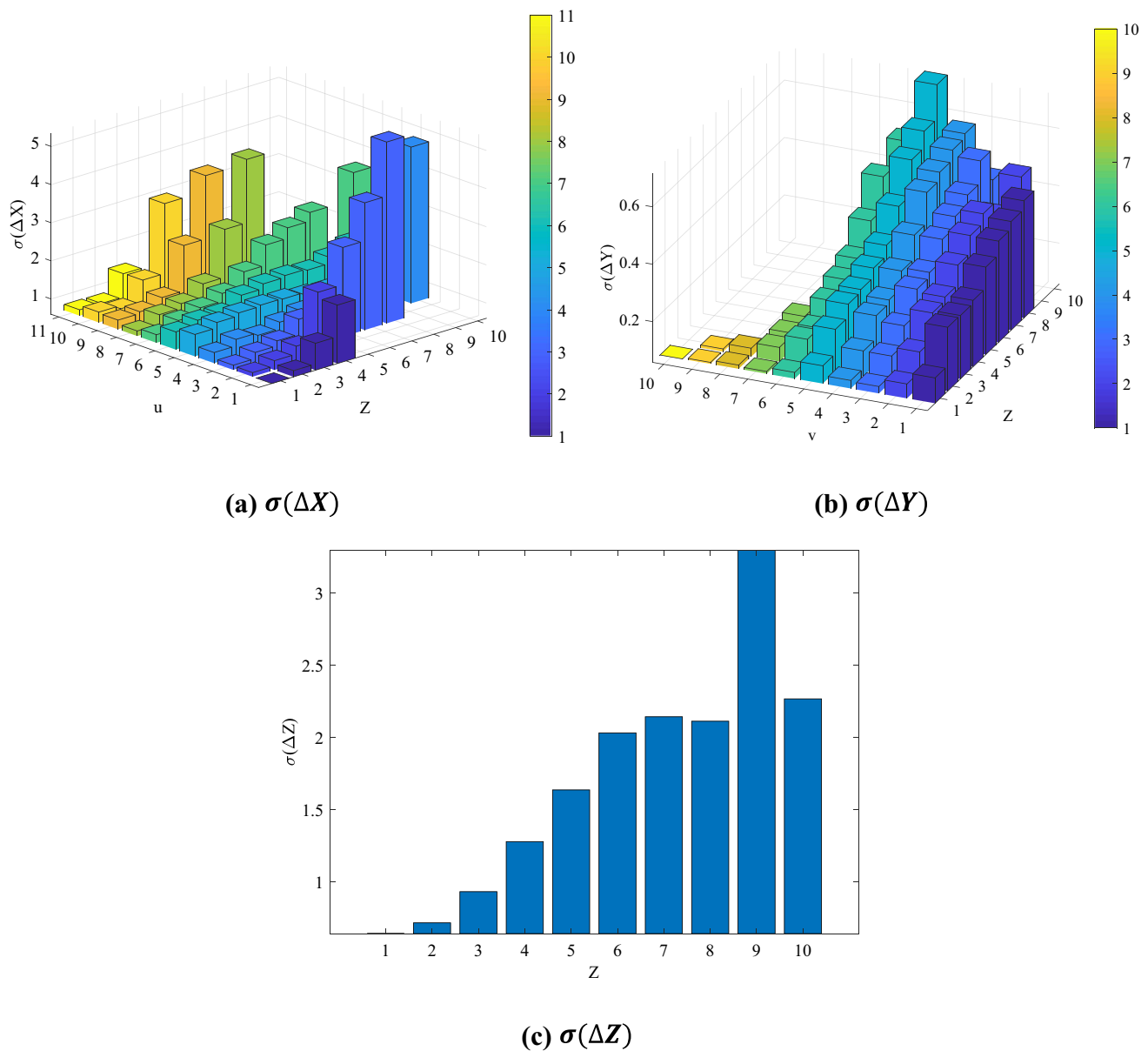
Therefore, we can solve the values of  $\sigma(\Delta Z)$ ,  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$  with the values of  $\sigma(\Delta d)$ ,  $\sigma(\Delta u)$ ,  $\sigma(\Delta v)$ . According to Eq. (17), the farther the 3D point is from the camera, the greater error of the depth. It can be concluded from Eqs. (20) and (22) that the closer the feature point is to the optical center of the camera, the smaller error of the X coordinate and the Y coordinate.

## 4 Experiments and discussion

To compute the values of  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$ ,  $\sigma(\Delta Z)$ , we designed the following experiment with KITTI odometry dataset [6]. We choose the images in the 00 sequence. For each image, we extract its inliers and save them in the corresponding file. The specific flowchart is shown in Fig. 1.

The feature points of the current frame and the previous frame are stored in the txt file. Besides, KITTI provides the rotation matrix  $R$  and the translation vector  $t$  between every two consecutive frames. The error of 3D point is defined as follows.

$$\begin{bmatrix} \Delta X_k \\ \Delta Y_k \\ \Delta Z_k \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix}_m - \left( R \begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ Z_{k-1} \end{bmatrix}_m + t \right). \quad (24)$$



**Fig. 5** The standard deviation of 3D point error in each direction

**Table 2** Threshold setting and fault rate

Experiment number	1	2	3	4	5
Threshold	1	2	3	4	5
Iterations	1000	1000	1000	1000	1000
Standard deviation	1.3966	1.6602	1.8769	2.0635	2.2367
Fault rate	0.0033	0.0030	0.0031	0.0034	0.0038

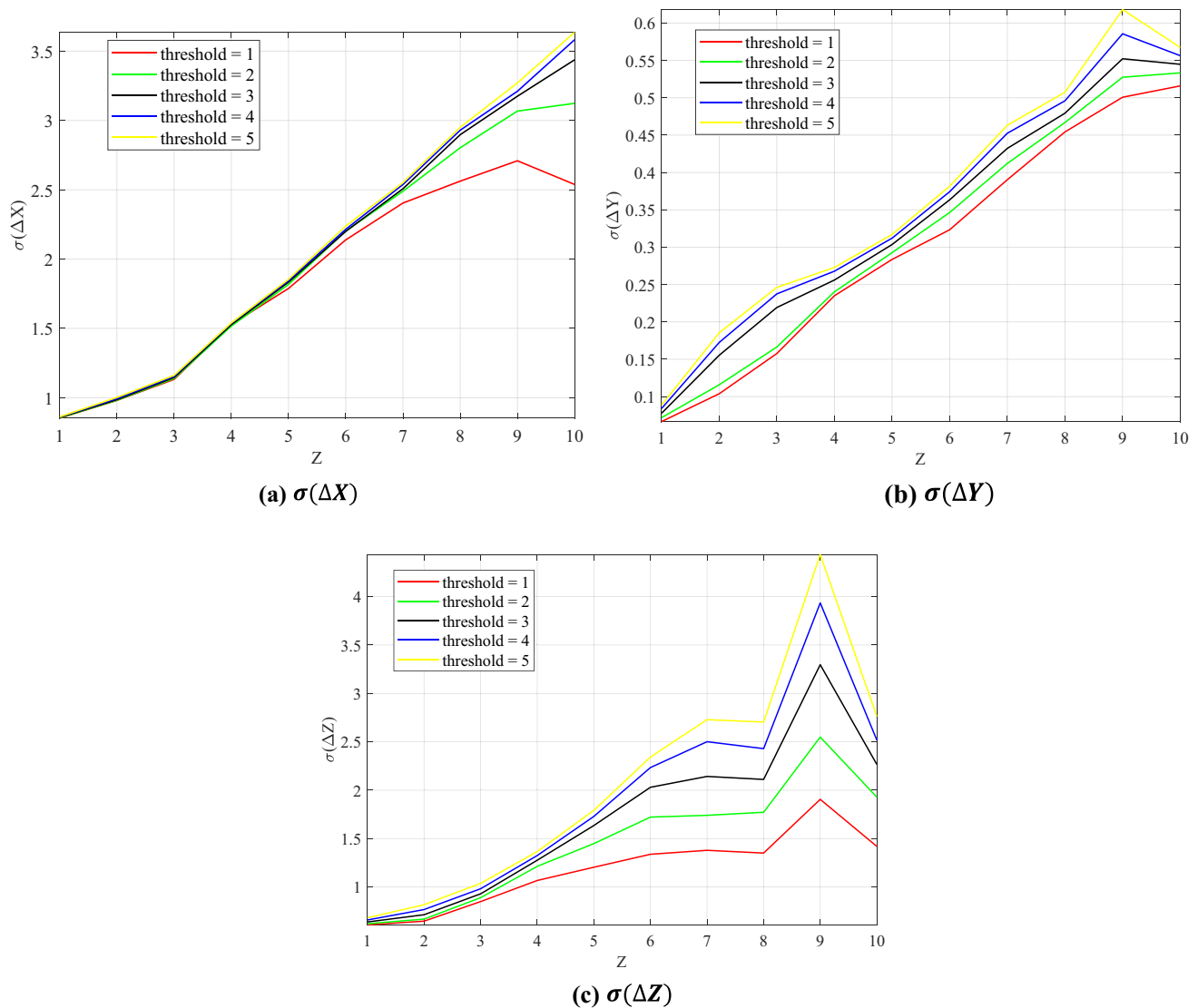
**Table 3** Iterations setting and fault rate

Experiment number	6	7	8	9	10
Threshold	3	3	3	3	3
Iterations	250	375	500	1000	1500
Standard Deviation	2.4507	2.2669	1.8506	1.8769	1.8958
Fault rate	0.0021	0.0022	0.0030	0.0031	0.0032

The  $\Delta X$ ,  $\Delta Y$ ,  $\Delta Z$  of outliers and the inliers can be obtained, as shown in Figs. 2 and 3, respectively. By comparison with these two figures, we can distinctly know the effectiveness of RANSAC. In Fig. 3, the value of  $\Delta X$  shows many spikes, the reason is that the 3D points with large errors

in those frames are the majority, which leads to a significant decrease of RANSAC performance.

Object movement in the scenario and mismatch of feature points are defined as faults, which lead to large errors. The standard deviation of 3D point errors is calculated in a



**Fig. 6** Standard deviation of 3D point error with different RANSAC thresholds

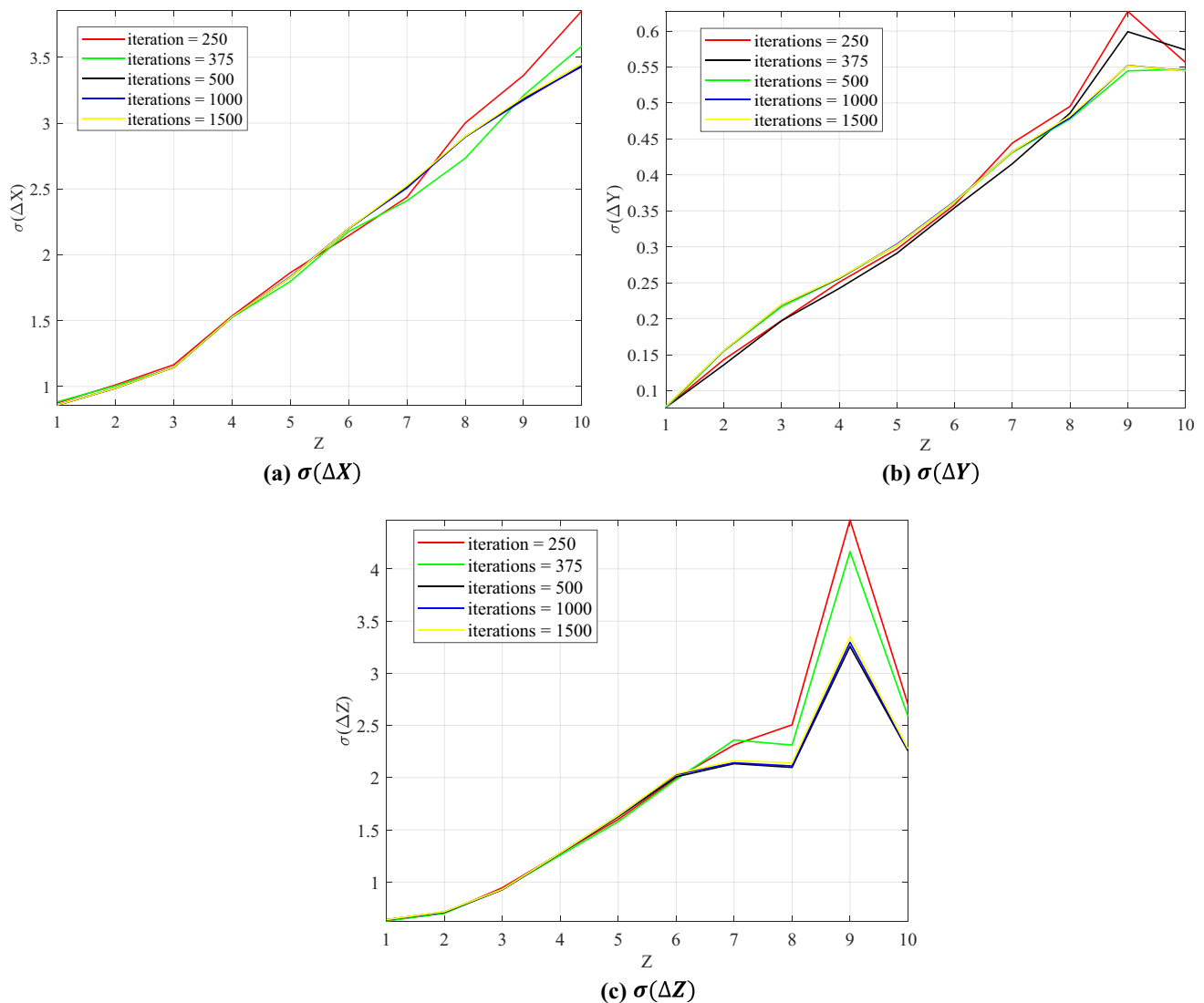
statistical manner. Furthermore, a threshold of six times the standard deviation is set, and the error of 3D point exceeding this threshold is regarded as a fault. The fault rate denotes the proportion of faulted points in the whole spatial points, and the calculation result is  $3.1 \times 10^{-3}$ . After fault exclusion, the values of  $\Delta X$ ,  $\Delta Y$ ,  $\Delta Z$  are shown in Fig. 4, and we determine  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$ ,  $\sigma(\Delta Z)$  with them.

It can be seen from Eq. (21) that the  $\sigma(\Delta X)$  of the 3D point is related to  $u$ ,  $Z$ ,  $\sigma(\Delta u)$ ,  $\sigma(\Delta d)$ . However,  $\sigma(\Delta u)$  and  $\sigma(\Delta d)$  are almost unavailable in practice. Consequently, the  $\sigma(\Delta X)$  is calculated in a statistical manner instead of using Eq. (21). Specifically, the 3D points are classified according to their position on the  $u-Z$  plane. First, we take the 3D points of  $0 \leq Z \leq 100$ ,  $100 \leq u \leq 1200$  on the  $u-Z$  plane. And then, we divide them equally into 110 sub-regions. Lastly, we calculate the  $\sigma(\Delta X)$  of errors of 3D points

in each sub-region. The results are shown in Fig. 5a. Among them, the vacant indicates that the number of 3D points in this sub-region is less than 1000, and the  $\sigma(\Delta X)$  in this subregion is not credible. It can be concluded from the figure that the upper bound of  $\sigma(\Delta X)$  is 4.358, most of which are less than 3.5. The  $\sigma(\Delta X)$  shows a plain upward trend as  $Z$  increases, while the trend of  $\sigma(\Delta X)$  is obscure in the  $u$  axis.

Similar analyses are performed to calculate the  $\sigma(\Delta Y)$  of 3D points. Considering the 3D points of  $0 \leq Z \leq 100$ ,  $30 \leq v \leq 330$  on the  $v-Z$  plane, they are divided equally into 100 subregions and then the  $\sigma(\Delta Y)$  of inliers in each subregion is calculated. The results are shown in Fig. 5b. The upper bound of  $\sigma(\Delta Y)$  is 0.7172, most of which are less than 0.55. The  $\sigma(\Delta Y)$  shows a clear upward trend as  $Z$  increases, while the trend of  $\sigma(\Delta Y)$  is not obvious in the  $v$  axis.





**Fig. 7** Standard deviation of 3D point error with different RANSAC iterations

**Table 4** Fault rate for different scenarios

Sequence number	00	02	08
Standard deviation	1.8769	2.0256	1.9736
Fault rate	0.0031	0.0037	0.0031

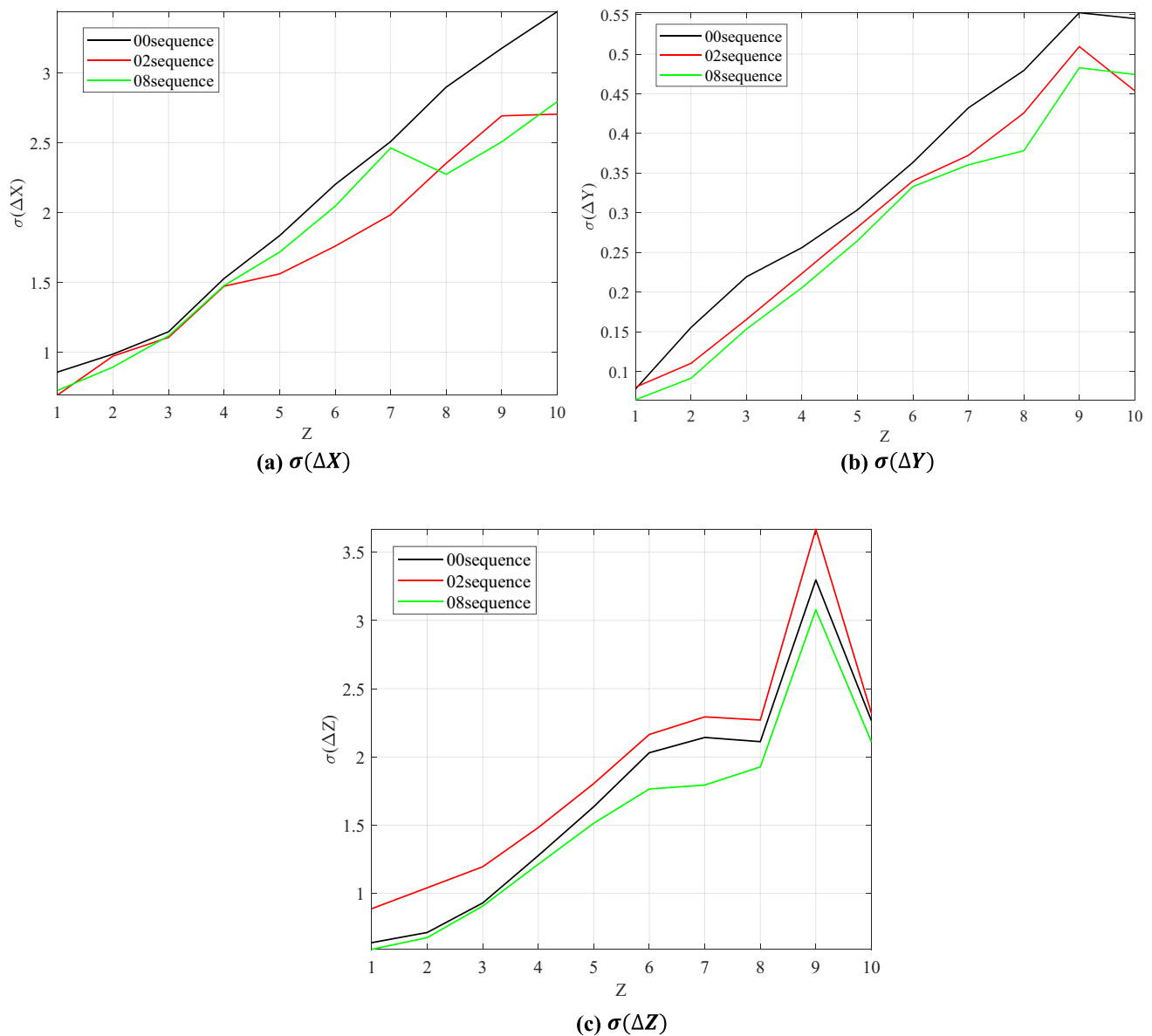
Similar analyses are conducted when computing the  $\sigma(\Delta Z)$  of 3D points. Hence, the 3D points are classified according to the value of  $Z$ . Specifically, the 3D points whose depth is under 100 m are divided into 10 intervals, and each interval is 10 m long. At last,  $\sigma(\Delta Z)$  is calculated in each interval, and the results are shown in Fig. 5c. The upper bound of  $\sigma(\Delta Z)$  is 3.297, and most of the values are less than 2.3. The  $\sigma(\Delta Z)$  shows a clear upward trend as  $Z$  increases.

Moreover, we analyze the impact of RANSAC iterations and RANSAC threshold on  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$ ,  $\sigma(\Delta Z)$  and

fault rate. In the previous experiment, the RANSAC iterations is 1000, and the RANSAC threshold is 3. To analyze the influence of RANSAC threshold on the experimental results, five sets of comparative experiments are set up, and the experimental parameters setting is shown in Table 2 (row 2 and row 3). For the fault rate, the experimental results are shown in Table 3 (row 5). When the RANSAC threshold is 2, the fault rate has a minimum value 0.0030; when the RANSAC threshold is 5, the fault rate has a maximum value 0.0038. In general, the fault rate does not change much as the RANSAC threshold changes.

Considering the insignificant change of  $\sigma(\Delta X)$  in the  $u$  axis and the visual effect of contrastive display, we distinguish 3D points by  $Z$  alone when we solve  $\sigma(\Delta X)$ . We divide 3D points whose depths are under 100 m into 10 intervals, and each interval is 10 m long. Then we calculate  $\sigma(\Delta X)$  of 3D





**Fig. 8** Standard deviation of 3D point error in different scenarios

points in each interval, and the results are shown in Fig. 6a. The ordinate represents  $\sigma(\Delta X)$ , and the abscissa represents the interval. The figure shows that the  $\sigma(\Delta X)$  increases as the RANSAC threshold increases. In a nutshell, the change of  $\sigma(\Delta X)$  is small with the change of RANSAC threshold.

The results for  $\sigma(\Delta Y)$  are shown in Fig. 6b. The ordinate represents  $\sigma(\Delta Y)$ , and the abscissa represents the interval. It can be seen from the figure that the  $\sigma(\Delta Y)$  increases as the RANSAC threshold increases. Overall, the change of  $\sigma(\Delta Y)$  is small with the change of RANSAC threshold.

The experimental results for  $\sigma(\Delta Z)$  are shown in Fig. 6c. The ordinate represents  $\sigma(\Delta Z)$ , and the abscissa represents the interval. The figure tells that the  $\sigma(\Delta Z)$  increases as the RANSAC threshold increases. When the RANSAC thresh-

old is 1, the  $\sigma(\Delta Z)$  in each interval is minimal. When the RANSAC threshold is 5, the  $\sigma(\Delta Z)$  in each interval is maximal. In a word, the  $\sigma(\Delta Z)$  changes more drastically than  $\sigma(\Delta X)$  or  $\sigma(\Delta Y)$  does as the RANSAC threshold changes.

To analyze the influence of RANSAC iterations on the experimental results, five sets of comparative experiments are conducted, and the experimental parameters setting is shown in Table 3 (row 2 and row 3). The experimental results for the fault rate are shown in Table 3 (row 5). The fault rate changes very little when the RANSAC iterations are larger than 500.

The results for  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$  and  $\sigma(\Delta Z)$  are shown in Fig. 7. They change very little as the RANSAC iterations

increases. The  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$  and  $\sigma(\Delta Z)$  become stable when the RANSAC iterations is larger than 500.

To analyze the adaptability of fault rate and  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$ ,  $\sigma(\Delta Z)$  to different scenarios, the experiment of different sequences under the same parameter settings are performed. The 00 sequence, 02 sequence and 08 sequence in the KITTI Odometry dataset were selected for contrast experiments. The RANSAC threshold and the RANSAC iterations are set to 3 and 1000, respectively.

The result for the fault rate is shown in Table 4. It tells that the change of the scenario has great effect on the fault rate. The results for  $\sigma(\Delta X)$ ,  $\sigma(\Delta Y)$  and  $\sigma(\Delta Z)$  are shown in Fig. 8. The change of  $\sigma(\Delta X)$  with the scenario is more obvious than that of  $\sigma(\Delta Y)$  or  $\sigma(\Delta Z)$ . Generally speaking, scenarios have a dramatical impact on standard deviation of 3D point.

## 5 Conclusions

This paper investigates the fundamental issues needed for constructing the integrity monitoring framework of stereo VO based on the feature point method. Specifically, we determine the error sources and faults in the stereo VO, and construct the model of errors of spatial points. The sensitivity analyses for the RANSAC threshold and the RANSAC iterations are carried out in this paper. It is found that the fault rate and standard deviation of errors of spatial points are not sensitive to the RANSAC iterations when they are larger than 500. Standard deviation of errors of spatial points along the Z axis is more sensitive to the RANSAC threshold than that along the X axis or Y axis. Furthermore, the fault rate changes a little with different RANSAC thresholds. Finally, the fault rate and standard deviation of errors of spatial points are obtained in the different scenarios to analyze the generalization ability of our model. It turns out that scenarios have a dramatic impact on them. In the future, researches on discriminator design will be carried out to reduce the fault rate in the stereo VO, and then the error model of spatial point without any faulty measurement will be constructed.

**Acknowledgements** This work is supported by China Postdoctoral Science Foundation (Grant no. 2019M661511) and Innovation Fund from Engineering Research Center of Aerospace Science and Technology, Ministry of Education.

## References

1. Aqel MOA, Marhaban MH, Saripan MI et al (2016) Review of visual odometry: types, approaches, challenges, and applications. SpringerPlus 5(1):1897
2. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: European conference on computer vision, pp 404–417
3. Engel J, Koltun V, Cremers D (2017) Direct sparse odometry. IEEE Trans Pattern Anal Mach Intell 40(3):611–625
4. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
5. Gao X, Zhang T, Liu Y et al (2019) 14 lectures on visual SLAM: from theory to practice, 2nd edn. Publishing House of Electronics Industry, Beijing
6. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suit. In: IEEE conference on computer vision and pattern recognition, pp 3354–3361
7. Hirschmuller H (2005) Accurate and efficient stereo processing by semi-global matching and mutual information. IEEE Comput Soc Conf Comput Vis Pattern Recognit 2:807–814
8. Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: 2007 IEEE and ACM international symposium on mixed and augmented reality, pp 225–234
9. Li CY, Waslander SL (2019) Visual measurement integrity monitoring for UAV localization. In: IEEE international symposium on safety, security, and rescue robotics (SSRR), pp 22–29
10. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
11. Mur-Artal R, Tardós JD (2017) Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans Rob 33(5):1255–1262
12. Nister D, Naroditsky O, Bergen JR (2004) Visual odometry. In: IEEE computer society conference on computer vision and pattern recognition, vol 1, p I
13. Pan WC, Zhan XQ, Zhang X (2019) Fault exclusion method for ARAIM based on tight GNSS/INS integration to achieve CAT-I approach. IET Radar Sonar Navig 13(11):1909–1917
14. Pomerleau F, Colas F, Siegwart R (2015) A review of point cloud registration algorithms for mobile robotics. Found Trends Robot 4(1):1–104
15. Rublee E, Rabaud V, Konolige K et al (2012) ORB: an efficient alternative to SIFT or SURF. In: 2011 international conference on computer vision, pp 2564–2571
16. Scaramuzza D, Fraundorfer F (2011) Visual odometry [tutorial]. IEEE Robot Autom Mag 18(4):80–92
17. Scharstein D, Szeliski R (2001) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vis 47(1–3):7–42
18. Wang SZ, Zhan XQ, Fu YW et al (2020) Feature-based visual navigation integrity monitoring for urban autonomous platforms. In: Aerospace systems, pp 1–13
19. Zhai YW, Joerges M, Pervan B (2018) Fault exclusion in multi-constellation global navigation satellite systems. J Navig 71(6):1281–1298
20. Zhan H, Weerasekera CS, Bian J, et al (2019) Visual odometry revisited: what should be learnt? [J]. arXiv preprint arXiv:1909.09803
21. Zhu C, Joerges M, Meurer M (2020) Quantifying feature association error in camera-based positioning. In: IEEE/ION position, location and navigation symposium (PLANS), pp 967–972