

## Article

# GMS-RANSAC: A Fast Algorithm for Removing Mismatches Based on ORB-SLAM2

Daode Zhang <sup>1,†</sup>, Jinlun Zhu <sup>1,†</sup>, Fusheng Wang <sup>1</sup>, Xinyu Hu <sup>1</sup> and Xuhui Ye <sup>1,2,\*</sup>

<sup>1</sup> School of Mechanical Engineering, Hubei University of Technology, Wuhan 430062, China; 19951044@hbut.edu.cn (D.Z.); 102000017@hbut.edu.cn (J.Z.); 102110162@hbut.edu.cn (F.W.); 19991012@hbut.edu.cn (X.H.)

<sup>2</sup> Collaborative Innovation Center of Intelligent Green Manufacturing Technology and Equipment, Qingdao 266000, China

\* Correspondence: yxh89@hbut.edu.cn

† These authors contributed equally to this work.

**Abstract:** This paper presents a new method of removing mismatches of redundant points based on oriented fast and rotated brief (ORB) in vision simultaneous localization and mapping (SLAM) systems. On the one hand, the grid-based motion statistics (GMS) algorithm reduces the processing time of key frames with more feature points and greatly increases the robustness of the original algorithm in a complex environment. On the other hand, aiming at the situation that the GMS algorithm is prone to false matching when there are few symmetry feature point pairs, the random sample consensus (RANSAC) algorithm is used to optimize and correct it. Experiments show that the method we propose has an average error correction rate of 28.81% for individual GMS while the time consumed at the same accuracy threshold is reduced by 72.18% on average. At the same time, we compared it to locality preserving matching (LPM) and progressive sample consensus (PROSAC), and it performed the best. Finally, we integrated GMS-RANSAC into the ORB-SLAM2 system for monocular initialization, which results in a significant improvement.

**Keywords:** visual SLAM; ORB-SLAM2; GMS; RANSAC; feature matching



**Citation:** Zhang, D.; Zhu, J.; Wang, F.; Hu, X.; Ye, X. GMS-RANSAC: A Fast Algorithm for Removing Mismatches Based on ORB-SLAM2. *Symmetry* **2022**, *14*, 849. <https://doi.org/10.3390/sym14050849>

Academic Editor: Mihai Postolache

Received: 19 March 2022

Accepted: 11 April 2022

Published: 20 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/>).

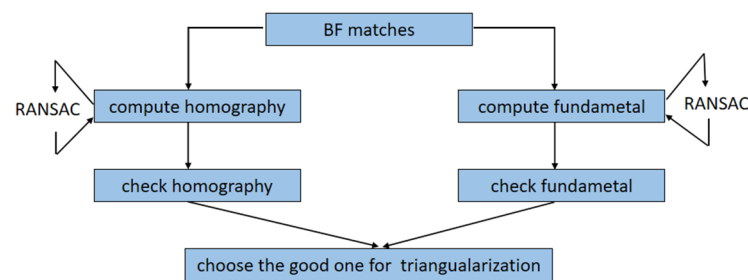
## 1. Introduction

With the proposal and popularization of intelligence in robotics, this also accelerates the development of simultaneous localization and mapping (SLAM), which plays an important role in applications such as service robots, VR, 3D reconstruction and autonomous navigation platforms. Depending on the kinds of sensors, we usually divide SLAM into two main categories. One is based on a laser. The distance and angle of markings in adjacent keyframes is used to estimated position and motion with Rao-Blackwellized Particle Filter in the robot coordinate system [1]. The other category is vision SLAM, which can be carried out by using just a monocular camera. With good robustness and accurate recognition in position, these cameras which provide rich environmental information are also cheap.

MonoSLAM [2] is the first real-time monocular vision SLAM system, in which the Extended Kalman Filter (EKF) [3] is used as the back-end to track the sparse feature points at the front-end. In 2007, Klein et al. proposed Parallel Tracking and Mapping (PTAM) [4] that realized the parallelization of the tracking and drawing process and used nonlinear optimization instead of conventional filters at the back-end. ORB-SLAM [5] was put forward in 2015. It has good versatility for supporting monocular, stereo and RGB-D. In the traditional feature point method of the SLAM process, feature matching is a critical step in visual SLAM [6], which fixes an issue with data association in SLAM. This means the correspondence was determined between a road sign that is currently seen and before it was seen. By accurately matching the description subs between the image and the image or the map [7], we can reduce a lot of burden for subsequent attitude estimation, optimization and

other operations. However, due to the local characteristics of image features, mismatches are widespread, and this has not been effectively solved for a long time [8]. At present, it has become a bottleneck in visual SLAM that restricts performance improvement [9].

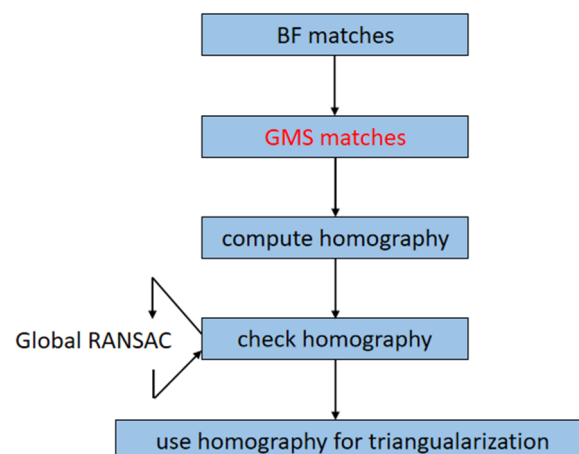
As Figure 1 shows, researchers usually estimate the Fundamental matrix and Homography matrix [10], and then choose the one which has smaller reprojection error to perform the exercise estimation. We prefer to use random sample consensus (RANSAC) [11] techniques when there may be mismatches. To meet the corresponding accuracy requirements when processing large amounts of data, the number of iterations must be increased, which greatly increases the time it takes to match images [12]. The response time has a great impact on real-time performance [13], and its reduction will sacrifice the accuracy of the SLAM system. Therefore, it is difficult to improve the response time and accuracy at the same time when the amount of data is large.



**Figure 1.** Partial flow of feature point method in SLAM.

This paper presents a new method in which grid-based motion statistics (GMS) [14] and RANSAC techniques are introduced to eliminate the incorrect matches. The GMS algorithm reduces the processing time of key frames with more feature points and greatly increases the robustness of the original algorithm in a complex environment. On the other hand, aiming at the situation that the GMS algorithm is prone to false matching when there are few feature points, the proposed algorithm can correct it. By experimental comparison, GMS-RANSAC improved the accuracy of GMS by about 28.81% and saved RANSAC about 72.18% of the time by setting the RANSAC threshold instead of setting the number of iterations.

Finally, we projected the method to create a new initializer based on the ORB-SLAM2 [15] system for monocular initialization, in which the number of the RANSAC model is decreased significantly. Figure 2 presents the initialization process for new design. The number of models is rapidly controlled within a corresponding range by GMS and canceling the number of iterations makes optimization global to achieve a better model, which results in a significant improvement.



**Figure 2.** Initialization process for new design.

The remainder of the paper is organized as follows: Section 2 describes related work. In Section 3, we present our algorithm. The results are evaluated in Section 4. Section 5 concludes the paper. The source code of the demo is at <https://github.com/fycie/demo.git>, accessed on 10 March 2022.

## 2. Related Work

### 2.1. RANSAC

RANSAC is used to randomly find a few points from all observations to fit the model and calculate the residuals of the model and all observation data in turn. All points will be judged as intrinsic points and be counted when they are less than a given threshold. On the contrary, a point could be judged as an outer point. Then, a few points are randomly selected to fit the model iteration again. If the number of fit inner points is larger than the previous model, the old is iterated into the new one.

The traditional RANSAC algorithm needs to set an upper bound of iteration value and error threshold, calculate the transformation matrix  $H_1$  through four symmetry pairs of sample data, and record it as a data detection model  $M_1$  which is used to measure all of the points and calculate the reprojection error. When the error is less than the set threshold, this point is classified as an inner point and added to the set of inner points. If the current set of inner points is more than the optimal, the current set is updated to the optimal set of inner points. Then, we update the number of iterations  $n$ . The process ends when the current is greater than the set number of iterations, otherwise the above steps are repeated and iterations are set to  $n + 1$ . It is expressed as:

$$N = \frac{\lg(1 - p)}{\lg(1 - \phi^m)} \quad (1)$$

If a maximum overlap is not set in advance generation value, the iterations will continue in this process. The optimal solution obtained at the end is determined by the limitation of the number of iterations, and the upper limit of the number of iterations is strongly consistent with the probability of obtaining the best model, which means the greater the upper limit of the number of iterations, the greater the probability of obtaining the best model. Although the iterations are increased at the same time, it slows down the algorithm. In the ORB-SLAM2 system, RANSAC rudely iterates a smaller number of times to ensure that the response time meets the real-time requirements. The proposed method is not to limit the number of iterations, but to set the threshold to ensure the accuracy of the results. No matter how large the amount of data is, in our method the GMS algorithm can quickly filter data to prepare for the RANSAC algorithm. Figure 3 shows the example of RANSAC algorithm.

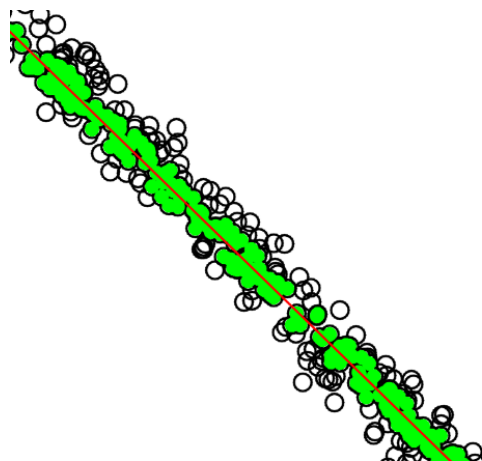


Figure 3. The example of RANSAC algorithm.

## 2.2. GMS

Combining smoothness constraints into feature matching is known to enable super robust matching. However, it was complex and time-consuming for us to succeed in this process in the past. GMS proposes a simple method to illustrate motion smoothness as a number of matching statistical possibilities in an area, which makes high-quantity matching with displacement have a high match quality. This provides a real-time, robust matching system. Evaluation of a video with low texture, blur and wide baselines (a pair of images that can be thought of as rotating displacements) shows that GMS exceeds other real-time matching methods and achieves the same effect as more complex, slower matching methods.

The key idea of GMS technology is to combine motion smoothing constraints (adjacent pixels of images from different viewpoints share similar motion) [12] into a statistical framework to reject wrong matching. The motion smoothing assumption means that the real matching neighborhood view is the same 3D region. Formally, the number of similar neighbors is used to identify good correspondences. The flaw in the model is that when there are few points within each grid, confidence of neighborhoods is low, which may lead to a large number of error results. By fusing RANSAC, the proposed approach is more robust than GMS when faced with a dataset containing errors.

## 2.3. LPM

Locality preserving matching (LPM) [16] is to maintain the local neighborhood structures of those potential true matches. It derives a closed-form solution with linearithmic time and linear space complexities in which the problem was formulated into a mathematical model [17]. It utilizes a more relaxed geometric constraint, yet it achieves surprisingly excellent performance and becomes the most popular performer considering the time cost [18]. Compared with LPM, our method obtains better performance in accuracy and runtime.

## 3. Methodology

### 3.1. Workflow of the Method

Figure 4 shows the workflow of the proposed method. Specifically, the main workflow is as follows:

- (1) Firstly, the ORB feature points of two images are extracted, respectively;
- (2) Then, the ORB features are matched by Hamming distance of descriptor;
- (3) Thirdly, the results of the previous step are roughly screened by GMS algorithm, which makes the number of matches greatly reduced;
- (4) Finally, the outliers are further removed by setting the random sampling consistency of the threshold.

As shown in Figure 4, first, the same number of ORB feature points was created to match two images  $I = \{I_1, I_2\}$ . We recorded their horizontal and vertical coordinates as  $kp_i$ , so that  $KP_i = \{kp_1, kp_2, \dots, kp_n\}$ . Secondly, by calculating the Hamming distance of ORB features, we found the corresponding matching points through  $I_2$  in the order in which the feature points were stored in  $KP_1$  and then recorded their serial numbers in the  $KP_2$ .  $m_i$  is the number of  $KP_2$ , which is matched to  $kp_i$  in  $KP_1$ . If there were no points matched,  $m_i = -1$ . After that, we would obtain  $M_1 = \{m_1, m_2, \dots, m_n\}$  and then

$$F_{gms}(i, m_i) = \begin{cases} -1, m_i \leq 0 \\ f_{gms}(i, m_i), m_i > 0 \end{cases} \quad (2)$$

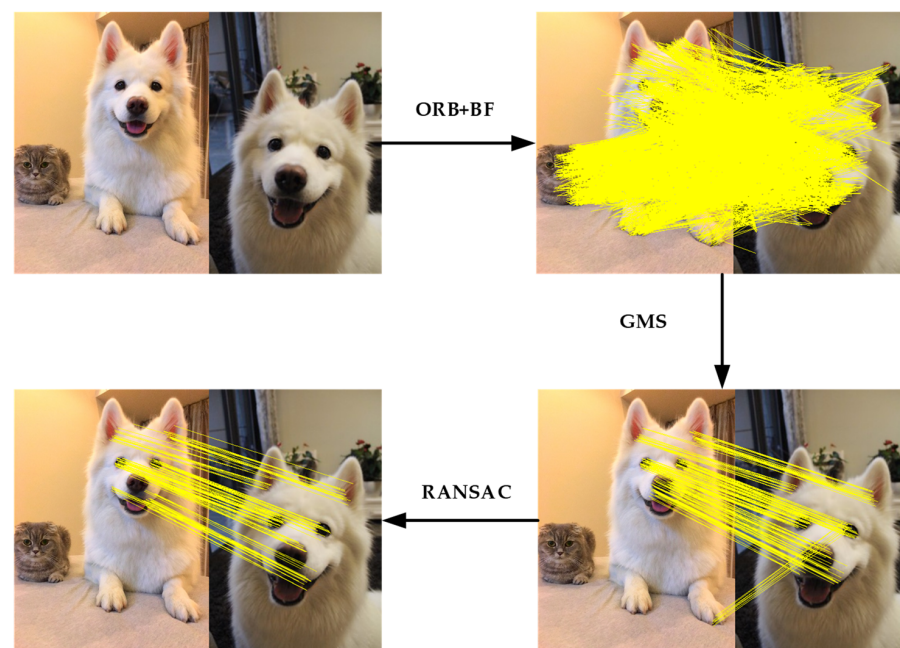
We used Equation (2) that is GMS to set the value of  $m_i$  which did not meet the requirements to  $-1$ .



According to the idea of GMS algorithm, the image was divided to be matched into  $G \times G$  grids, and we set  $G$  to 20 in this paper.  $M_2 = \{m_1, m_2, \dots, m_n\}$  was obtained through GMS. Then, we sorted out the data. When  $m_i \neq -1$ , we set

$$p_n = f(i, m_i) \quad (3)$$

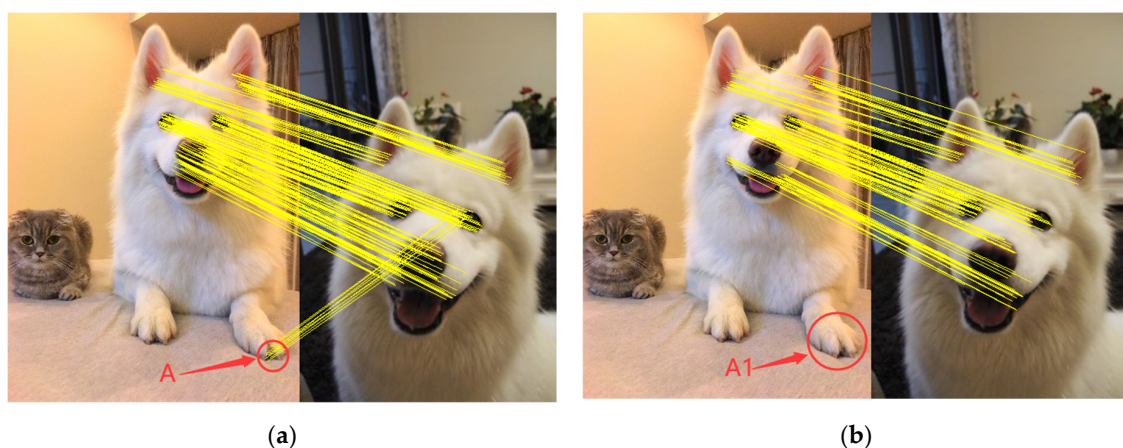
As a result,  $P_g = \{p_1, p_2, \dots, p_n\}$ , which was prepared for RANSAC. After normalizing the points, we used RANSAC algorithm to calculate the Homography matrix, and set the threshold of the algorithm to 5. According to the same form before, the outer points were eliminated by re-projection error. Finally, we obtained  $P_{g+r} = \{p_1, p_2, \dots, p_n\}$ , which are excellent matches.



**Figure 4.** The workflow of the proposed method.

### 3.2. GMS Mismatch Correction

Figure 5 presents false matches of GMS in left picture when 2000 features are created, and the matches are corrected in the right image after RANSAC. RANSAC takes only 0.68 ms and improves 42% percent of accuracy at the criterion of the RANSAC threshold of 5.



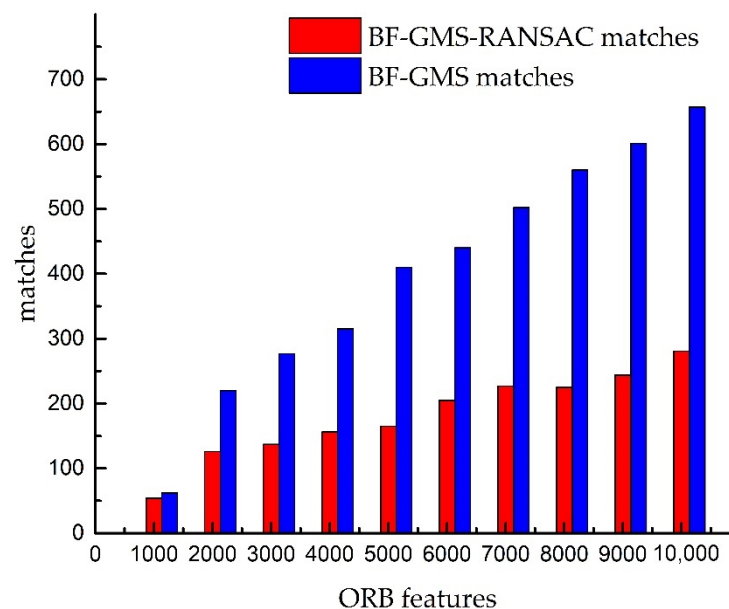
**Figure 5.** Comparison of two methods under 2000 features: (a) ORB-BF-GMS, (b) ORB-BF-GMS-RANSAC.

Then, GMS was compared with GMS-RANSAC at different numbers of ORB feature points. Figure 6 shows that GMS-RANSAC becomes more effective with the number of points increasing, and the great gap from 1000 to 2000 is the weaknesses of the GMS. The trend of optimization also reflects changes in GMS accuracy. We processed the data according to the threshold value of 5. The highest point occurs when the number of ORB feature points is 5000, and the accuracy is 59.76%. Each of feature points were matched for record, and then set as  $Q_i = \{X_1, X_2, \dots, X_n\}$ . The maximum and minimum values in  $\bar{Q}_i$  were removed for analysis.

$$\bar{Q}_i = \frac{\sum_{i=1}^n \bar{X}_i}{n} \quad (4)$$

The average improvement efficiency  $A$  is about 54.24%.

$$A = \frac{\sum_{i=1}^n \bar{Q}_i}{n} \quad (5)$$



**Figure 6.** Number of matches after filtering.

### 3.3. The Acceleration of RANSAC

GMS for continuity-based neighborhood search matching can quickly reduce a large number of coarse matches. Time of different matching methods was measured by taking same number of ORB feature points from the same two pictures, and numbers of fine matches were recorded.

As Figure 7 shows, RANSAC takes more time with ORB feature points increasing at the same RANSAC threshold. For example, after rough matching, outliers are removed by the RANSAC algorithm. It takes 64.23 ms to reach the standard algorithm with a threshold of 5 based on 1000 ORB features, but the method proposed only takes 1.86 ms in total. GMS consumed 1.18 ms and other times are RANSAC. In addition, through the calculation from 1000 to 10,000 feature points at 1000 intervals, the average time consumed was reduced by 94% in parallel to RANSAC. Because GMS rapidly reduces the number of matches, the number of RANSAC reaching the 5 threshold is greatly reduced. This is the reason why the time required by RANSAC algorithm was reduced.

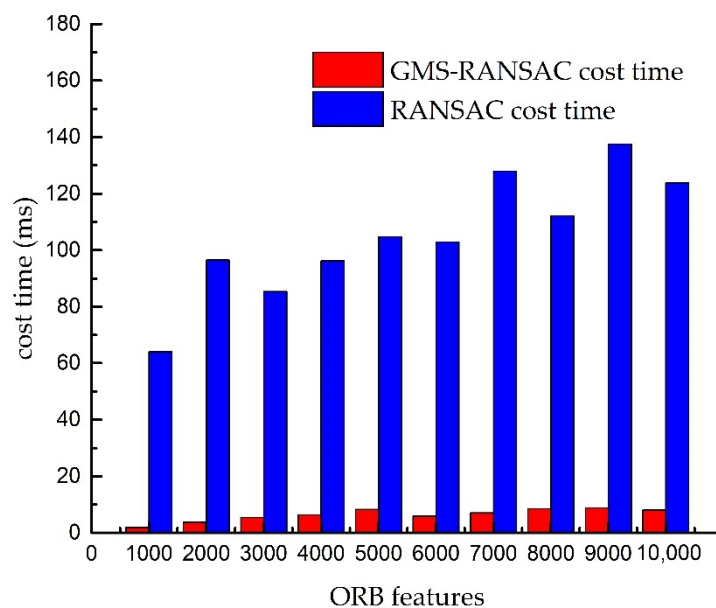


Figure 7. The cost time of two methods.

#### 4. Experiments

The experiments were conducted on a 4-threaded, 4-RAM virtual machine ubuntu16.04 installed by VMware Workstation Pro under intel i5-8400 CPU, 16 G RAM, 256 G SSD desktops and C++ code.

##### 4.1. GMS-RANSAC for Correspondence Selection

In order to evaluate GMS-RANSAC performance comprehensively, we experimented with different local features and different number of points. Two frames were selected from a range of datasets for experimental comparison to simulate the operation of scenarios.

##### 4.1.1. Datasets and Metrics

It consists of 2 datasets, including TUM [19] and KITTI [20], which provide indoor scenes and street views. We calculated the ratio by counting the time spent by the algorithm and the number of good matches obtained in Section 3. Figure 8 presents the comparison of RANSAC and GMS-RANSAC in different datasets at 3000 feature points, and a.1, b.1 and c.1 represent the results of RANSAC under TUM and KITTI. The other results are GMS-RANSAC’s. We evaluate the two methods by comparing the number of matches and the time it takes.

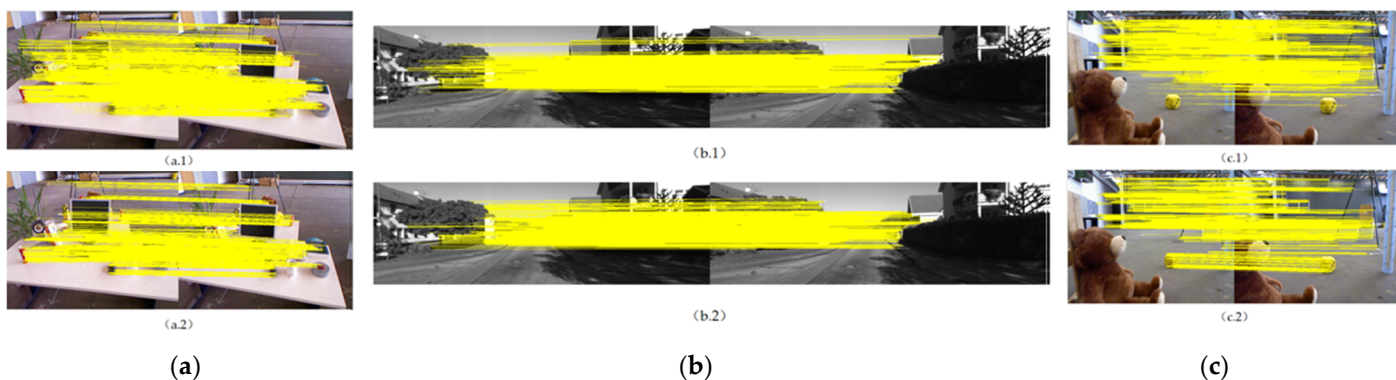
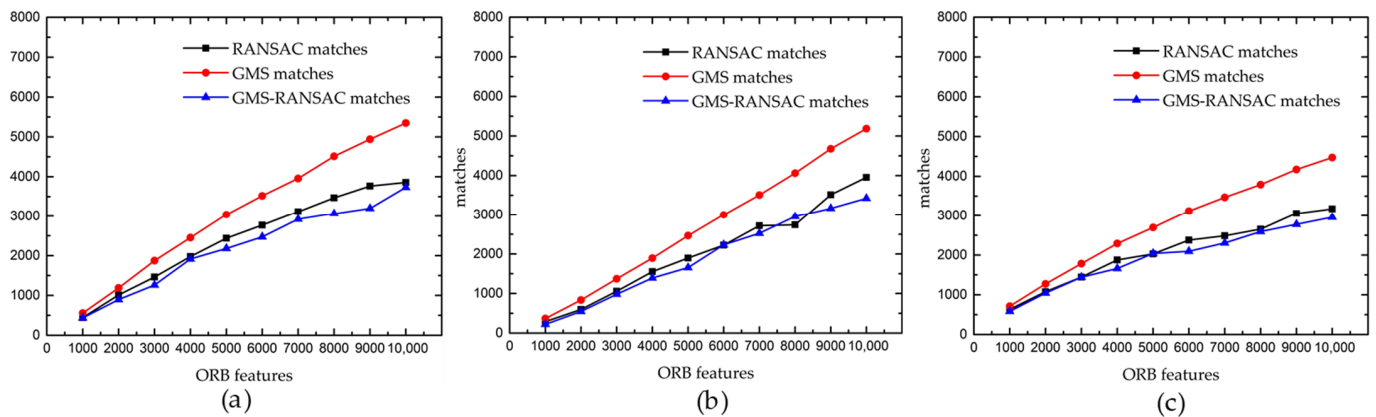


Figure 8. Comparison of RANSAC and GMS-RANSAC in different datasets at 3000 ORB feature points: (a) on TUM desk, (b) on KITTI, (c) on TUM doll.

#### 4.1.2. Experimental Results

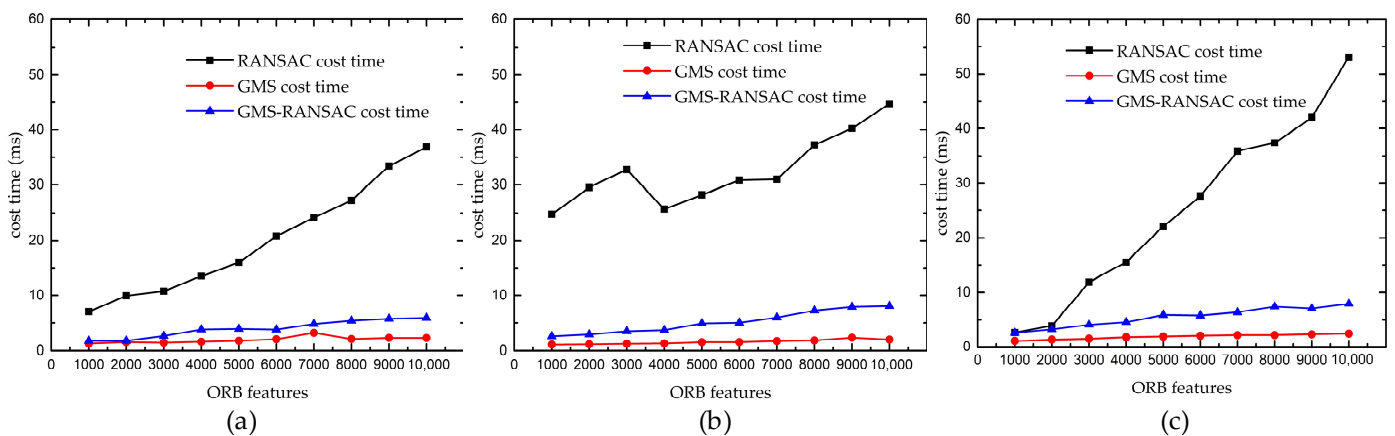
It can be seen from Figure 9a that the matching numbers of RANSAC are more than GMS and GMS-RANSAC. As the number of ORB points increases, all three methods gradually increase, and the gap between RANSAC and the other two methods increases. GMS-RANSAC is slightly higher than GMS around 8000 points of ORB features on KITTI in Figure 9b. Figure 9c shows the results on TUM dataset in which the total number filtered by the method is lower than that of the other two datasets.



**Figure 9.** Matches of the three algorithms in different datasets: (a) on TUM desk, (b) on KITTI, (c) on TUM doll.

Figure 9a–c shows that the number of GMS-RANSAC results is similar to the number of high-quality matches obtained by the original RANSAC method. Using Equations (4) and (5), we summarize the correction rate of the above GMS-RANSAC compared to the GMS. The data show that the number of matches obtained after GMS processing is greater than RANSAC. However, when further filtered with the same threshold RANSAC, the resulting number of fine matches is not much different from the original.

As can be seen in Figure 10a, the cost time of three methods increases with the increasing number of ORB features on TUM desk. All three algorithms reach their maximum values at 10,000 ORB features. It can be seen in Figure 10b that there are several inflection points in RANSAC on KITTI. The extremum of RANSAC is 32.735 ms at 3000, and the time consumed by the three algorithms is relatively close when ORB feature points are 1000–2000 (Figure 10c). The gap becomes increasing gradually with the number of ORB features rising.



**Figure 10.** Cost time of the three algorithms in different datasets: (a) on TUM desk, (b) on KITTI, (c) on TUM doll.



Figure 10a–c present that as the ORB feature points increase, the time consumed by RANSAC increases linearly. In contrast, the time required for GMS stabilized at 1–2ms. GMS-RANSAC, where the growth of GMS-RANSAC is stabilizing.

As shown in Table 1, the method we propose has an average error correction rate of 28.81% for individual GMS. On the other hand, there is a reduction of about 72.18% in time for individual RANSAC. Overall, it is a significant improvement in feature matching. The effect of GMS-RANSAC is most pronounced on the TUM dataset, and the maximum of correction rate between GMS and GMS-RANSAC is 30.94%. The time reduction maximum between GMS and GMS-RANSAC is 84.54%. It plays an important role in both ways.

**Table 1.** Data obtained from three sets of experiments.

Datasets	Correction Rate for GMS GMS-RANSAC	Time Reduction for RANSAC GMS-RANSAC
TUM desk [19]	28.28%	67.94%
KITTI [20]	27.23%	64.07%
TUM [19]	30.94%	84.54%
Average value	28.81%	72.18%

#### 4.1.3. Comparisons

Table 2 reports the experimental results for PROSAC [21], LPM and GMS-RANSAC. Here, we mainly compare algorithms in terms of %Precision, %Recall and time cost after RANSAC-based outlier removal.

**Table 2.** Experimental results of three algorithms under 2000 ORB features.

Datasets		PROSAC [21]	LPM [16]	GMS-RANSAC
TUM desk [19]	%Precision	88.05	71.94	<b>88.74</b>
	%Recall	<b>100.00</b>	84.77	93.98
	Cost time (ms)	28.39	25.93	<b>3.23</b>
KITTI [20]	%Precision	<b>91.53</b>	57.55	86.86
	%Recall	<b>100.00</b>	84.77	87.18
	Cost time (ms)	27.89	23.94	<b>2.89</b>
TUM [19]	%Precision	<b>93.29</b>	77.42	89.27
	%Recall	90.41	88.29	<b>90.93</b>
	Cost time (ms)	29.47	24.96	<b>3.17</b>

All the above results demonstrate that GMS-RANSAC can show better performance with LPM using the same feature correspondences as input. Although PROSAC is outstanding, it is less stable at runtime with different number of points and consumes more time.

#### 4.2. The New Initializer Based on ORB-SLAM2

To save computing resources, we inserted GMS-RANSAC in *SearchForInitialization*. According to its original practice,  $v_{matched12}[i]$  at the outer point was set to  $-1$  to mark it. Furthermore, the number of iterations of the original group was replaced by the threshold method, which was integrated into the initialization of ORB-SLAM2. The number of iterations and the parallax angle of the initialization decision were changed to keep system stability. We set the number of iterations of the original system from 200 to 300 and adjusted the parallax angle limit from 0.36 to 0.2 to test the initializer.

##### 4.2.1. Datasets and Metrics

Because the monocular initialization in KITTI datasets was easy, we chose TUM dataset freiburg1\_xyz and freiburg1\_desk as the datasets for this experiment, and this was beneficial for us to identify the ability of initializers. In each test, we measured (1) how

many 3D points were triangulated; (2) how much time the algorithm cost. In addition, we calculated the ratio of the number of points to the initialization time to evaluate the efficiency of the initializers.

#### 4.2.2. Experimental Results

Table 3 shows the results, in which RANSAC is in parallel to GMS-RANSAC based on the original initializer. The results show that the proposed initializer can significantly improve the initialization efficiency and make the 3D map denser. This is of great significance for the ORB-SLAM2 system, especially when they meet the complex texture environment. The proposed method can greatly increase the robustness of the system.

**Table 3.** Modified ORB-SLAM2 initialization results on the TUM dataset freiburg1\_xyz (R:RANSAC G-R:GMS-RANSAC #3D Points: points by triangulation).

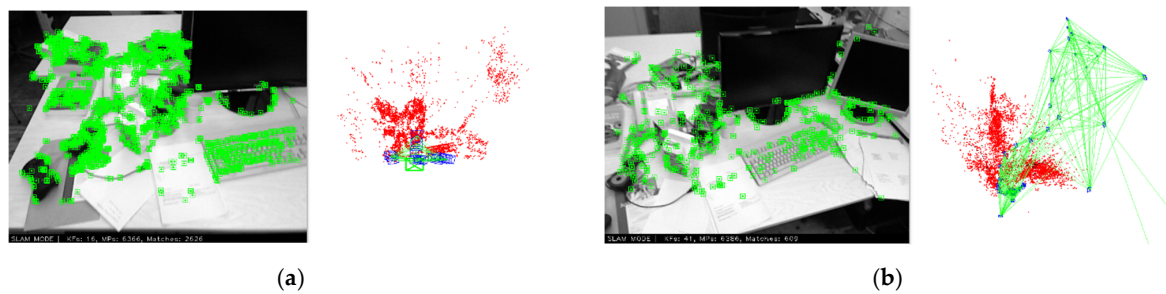
ORB Numbers	#3D Points		Initialization Time (ms)		Rate (Points/Time)	
	R	G-R	R	G-R	R	G-R
1000	109	<b>141</b>	17.4	<b>13.9</b>	6.3	<b>10.1</b>
2000	127	<b>217</b>	30.9	<b>30.7</b>	4.1	<b>6.5</b>
3000	143	<b>326</b>	55.3	<b>50.2</b>	2.6	<b>6.5</b>
4000	107	<b>392</b>	74.5	<b>69.1</b>	1.4	<b>5.7</b>
5000	131	<b>454</b>	85.5	<b>84.8</b>	1.5	<b>5.4</b>
6000	179	<b>970</b>	<b>104.6</b>	133.6	1.7	<b>7.3</b>
7000	672	<b>979</b>	<b>110.9</b>	130.2	6.1	<b>7.5</b>
8000	209	<b>973</b>	<b>122.6</b>	132.8	1.7	<b>7.3</b>
9000	210	<b>973</b>	<b>128.9</b>	139.4	1.6	<b>6.9</b>
10000	210	<b>973</b>	<b>129.1</b>	133.9	1.6	<b>7.3</b>

Table 4 presents that the original initializer tends to be stable in the number of triangulated map points while our proposed method increases steadily. Although the time will increase significantly with the increase of 3D points after 3000 ORB points, the rate is always higher than the original. Figure 11 presents the experimental scenarios under different datasets.

**Table 4.** Modified ORB-SLAM2 initialization results on the TUM dataset freiburg1\_desk (R:RANSAC G-R:GMS-RANSAC #3D Points: points by triangulation).

ORB Numbers	#3D Points Numbers		Initialization Time (ms)		Rate (Points/Time)	
	R	G-R	R	G-R	R	G-R
1000	102	<b>108</b>	21.8	<b>13.6</b>	4.7	<b>8.0</b>
2000	166	<b>171</b>	28.9	<b>27.4</b>	5.7	<b>6.2</b>
3000	125	<b>314</b>	<b>33.9</b>	50.0	3.7	<b>6.3</b>
4000	103	<b>411</b>	<b>32.5</b>	76.6	3.2	<b>5.4</b>
5000	103	<b>451</b>	<b>42.6</b>	97.0	2.4	<b>4.6</b>
6000	103	<b>556</b>	<b>34.9</b>	132.1	3.0	<b>4.2</b>
7000	103	<b>610</b>	<b>34.9</b>	146.9	2.9	<b>4.2</b>
8000	103	<b>665</b>	<b>31.8</b>	158.3	3.2	<b>4.2</b>
9000	103	<b>616</b>	<b>33.3</b>	138.6	3.1	<b>4.4</b>
10000	103	<b>616</b>	<b>32.6</b>	138.7	3.1	<b>4.4</b>





**Figure 11.** The scenes of ORB-SLAM2 changed: (a) on TUM dataset freiburg1\_xyz, (b) on TUM dataset freiburg1\_desk.

## 5. Conclusions

This paper proposes a fast correspondence selection algorithm which combines GMS and RANSAC. Consequently, it can make up for the shortcomings of the original two algorithms used alone and make them more robust, especially when the ORB features are too many or too little. Moreover, experiments show that the method we propose has an average error correction rate of 28.81% for individual GMS while the time consumed at the same accuracy threshold is reduced by 72.18% on average. This was also verified in the ORB-SLAM2 system, in which it shows its great potential in real-time applications.

We think that the RANSAC in ORB-SLAM2 is quantitative from the method. As experiments show, it can handle a peak of 3000, which does not fully utilize the acquired image information. This paper proposes a new method of global optimization, which can greatly improve the robustness of the system in a complex environment. We are simply using the Homography matrix for experiments at present, and a new contrast structure will be designed on the Homography matrix and the Fundamental matrix to improve the monocular initialization in the future. In addition, we are considering replacing bag of words (BOW) with GMS-RANSAC to make great progress in real time of the SLAM system. Furthermore, mapping needs to be further developed, and building more reliable and dense maps to support navigation is also the core pursuit of the SLAM system.

**Author Contributions:** D.Z. and J.Z. contributed equally to this manuscript. Conceptualization, D.Z., J.Z. and X.H.; Data curation, J.Z.; Formal analysis, J.Z., F.W. and X.H.; Investigation, X.Y.; Methodology, J.Z., F.W. and X.H.; Project administration, D.Z.; Resources, J.Z. and X.Y.; Software, J.Z.; Supervision, D.Z. and X.Y.; Validation, D.Z., J.Z. and F.W.; Visualization, J.Z. and F.W.; Writing—original draft, J.Z. and X.Y.; Writing—review and editing, J.Z. and F.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (No. 52075152) and the Collaborative Innovation Center of Intelligent Green Manufacturing Technology and Equipment, Shandong (IGSD-2020-006).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets used or analyzed during the current study are available from the corresponding author upon reasonable request.

**Acknowledgments:** We are grateful to the Hubei University of Technology for creating good experimental conditions for us. We are grateful to the National Natural Science Foundation of China for the grant support (No. 52075152) and the Collaborative Innovation Center of Intelligent Green Manufacturing Technology and Equipment, Shandong (IGSD-2020-006). Co-first author Jinlun Zhu thanks Gao Xiang's sharing at the technical level, for which he laid the foundation for his visual SLAM study. He also gives thanks for support and encouragement of his parents and teachers. At the same time, he would like to thank the people of Wuhan Relaxation, especially Xu, who accompanied him when he was depressed.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Grisetti, G.; Stachniss, C.; Burgard, W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In Proceedings of the IEEE International Conference on Robotics & Automation, Barcelona, Spain, 18–22 April 2005.
2. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [[CrossRef](#)] [[PubMed](#)]
3. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*; International Society for Optics and Photonics: Bellingham, WA, USA, 1997; Volume 3068, pp. 182–193.
4. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the IEEE & Acm International Symposium on Mixed & Augmented Reality, Cambridge, UK, 15–18 September 2008.
5. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
6. Opdenbosch, D.V.; Steinbach, E. Collaborative Visual SLAM Using Compressed Feature Exchange. In Proceedings of the International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2018; pp. 57–64.
7. Bescós, B.; Fàcil, J.; Civera, J.; Neira, J. DynSLAM: Tracking, Mapping and inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [[CrossRef](#)]
8. Kim, T. Feature Detection Based on Significance of Local Features for Image Matching. *IEICE Trans. Inf. Syst.* **2021**, *104*, 1510–1513. [[CrossRef](#)]
9. Gao, X.; Zhang, T. Robust RGB-D simultaneous localization and mapping using planar point features. *Robot. Auton. Syst.* **2015**, *72*, 1–14. [[CrossRef](#)]
10. Bian, J.W.; Wu, Y.H.; Zhao, J.; Liu, Y.; Zhang, L.; Cheng, M.M.; Reid, I. An Evaluation of Feature Matchers for Fundamental Matrix Estimation. In Proceedings of the British Machine Vision Conference, Cardiff, UK, 9–12 September 2019.
11. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications To Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
12. Bian, J.-W.; Lin, W.-Y.; Liu, Y.; Zhang, L.; Yeung, S.-K.; Cheng, M.-M.; Reid, I. GMS: Grid-Based Motion Statistics for Fast, Ultra-robust Feature Correspondence. *Int. J. Comput. Vis.* **2019**, *128*, 1580–1593. [[CrossRef](#)]
13. Tang, J.X.; Ericson, L.; Folkesson, J.; Jensfelt, P. GCNv2: Efficient Correspondence Prediction for Real-Time SLAM. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3505–3512. [[CrossRef](#)]
14. Bian, J.; Lin, W.Y.; Matsushita, Y.; Yeung, S.K.; Cheng, M.M. GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
15. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
16. Ma, J.; Zhao, J.; Jiang, J.; Zhou, H.; Guo, X. Locality Preserving Matching. *Int. J. Comput. Vis.* **2019**, *127*, 512–531. [[CrossRef](#)]
17. Ma, J.; Jiang, J.; Zhou, H.; Zhao, J.; Guo, X. Guided Locality Preserving Feature Matching for Remote Sensing Image Registration. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4435–4447. [[CrossRef](#)]
18. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image Matching from Handcrafted to Deep Features: A Survey. *Int. J. Comput. Vis.* **2020**, *129*, 23–79. [[CrossRef](#)]
19. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the (IROS) 2012: IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012.
20. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
21. Chum, O.; Matas, J. Matching with PROSAC—Progressive sample consensus. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–25 June 2005.