# Visual Odometer Based on Image Region Texture Weights of ORB Features

Di Wu *, Zhihao Ma, Weiping Xu, Haifeng He, Zhenlin Li

Posted Date: 21 July 2023

doi: 10.20944/preprints202307.1477.v1

Keywords: key stereo vision odometry; systematic error; prognostic model; texture area weighting; positioning error

*Article*

# Visual Odometer Based on Image Region Texture Weights of ORB Features

**Di Wu, Zhihao Ma, Weiping Xu, Haifeng He and Zhenlin Li**

[1] College of Mechanical and Electrical Engineering, Guizhou Normal University, Guiyang, Guizhou 550001, China

* Correspondence: 201907001@gznu.edu.cn

**Abstract:** In order to improve the visual odometry with ORB (Oriented FAST and Rotated Brief) features, we propose to improve the visual odometry with ORB features to address the instability of the system in the process of driving at low speed. First, the poor corner point properties of ORB features and the lack of rich environment texture lead to mismatching, and the poor corner point properties of feature points due to poor environment texture are solved by using weight calculation based on different texture regions. The keyframes are proposed for motion model estimation so that the overlap between adjacent frames can be reduced when the camera movement speed is low, and the system computation tends to be stable, improving the system's stability under low-speed motion. The experimental results using the KITTI dataset show that the keyframe rate is best between 10% and 12%, which is also compared with the presence or absence of keyframes, and a comparison was also made with and without keyframes, and a comparative analysis was done with three common open source VO systems, and the positioning accuracy was higher than these open source systems.

**Keywords:** key stereo vision odometry; systematic error; prognostic model; texture area weighting; positioning accuracy

## 1. Introduction

Unmanned driving has brought great convenience in ore mining and transportation, car driving, factory production, and agriculture [1,**Error! Reference source not found.**], and localization algorithms are the basis and key to realizing unmanned driving. Currently, Simultaneous Localization and Mapping (SLAM) is commonly used and is divided into two types, laser-based and vision-based, depending on the sensor [3]. Laser SLAM started earlier than vision SLAM; while the technology of both products is relatively mature, but the cost is higher, while vision SLAM is receiving more and more attention due to the richer environmental information and low sensor cost [4].

Visual SLAM consists of five main components: sensor information acquisition, visual odometry, back-end optimization, Loop Closure Detection, and mapping [5]. The task of visual odometry is to estimate the motion of cameras between adjacent images, and its accuracy directly affects the performance of SLAM systems. Most of these visual odometry systems are feature-based, i.e., after extracting certain image features and representing them with descriptors, the camera motion is represented by matching between images in order to compute the conversion matrix between frames [6].

In this paper, the feature extraction algorithm is selected as ORB (Oriented FAST and Rotated Brief) features, which combines the FAST (Features from Accelerated Segment Test) extraction algorithm and BRIEF (Binary Robust Independent Elementary Features) descriptor, which makes it highly computationally fast and rotation invariant and scale-invariant and is currently the most commonly used feature operator in visual SLAM [7]. In the extraction process of ORB features, the feature points are usually concentrated, and in order to match more conveniently and also to calculate the minimization reprojection error more accurately, it should try to make the feature points evenly distributed in the whole image that is, homogenization [8-10].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



Both domestic and foreign scholars have provided unique insights on the improvement of ORB features; for example, Mur-Artal R et al. proposed the introduction of quadtree homogenization to improve the homogeneity of ORB feature point extraction, but the extraction rate of feature points is still low in weak texture regions [11]. , proposed an improved algorithm to improve the extraction rate of feature points by introducing adaptive thresholding, but the problem of low feature point uniformity was not effectively solved [12]. Chen Mianshu et al. improved the uniformity of feature point extraction based on the idea of grid division and hierarchical determination of key points but reduced the real-time performance of the system [13]. Yao Jinjin et al. proposed to set different quadtree depths for different pyramid layers to improve the computational efficiency, and the extraction time was reduced by 10% compared with the traditional algorithm, but the enhancement of the underlying texture image was not obvious [14]. Zhao Cheng et al. used quadtree homogenization and adaptive thresholding to reduce the degree of aggregation of feature points in texture information-rich regions and improve the homogeneity of feature point extraction [15]. However, because its adaptive thresholding depends on the image texture, it still does not solve the problem of poor matching accuracy of feature points in low-texture regions.

In response to the above research status, the main contributions of this paper are:

(1) To propose a matching algorithm based on the weight of feature point response values by studying the homogenization of ORB features in visual odometry.

(2) Incorporating a predictive motion model in the keyframes bit pose estimation.

## 2. Visual Odometry

### *2.1. System Framework*

A complete stereo vision odometry system consists of four parts [16]: image acquisition and preprocessing, feature extraction and matching, feature tracking and 3D reconstruction, and motion estimation. The system flowchart is shown in Figure 1, and the specific processes are as follows: extracting feature points on the left and right eye images; matching feature points based on Euclidean distance and polar line constraints; reconstructing the 3D coordinates of matched feature point pairs; tracking feature points in the next frame of the image pair; and calculating the camera pose by solving the minimum reprojection error problem for the feature points.

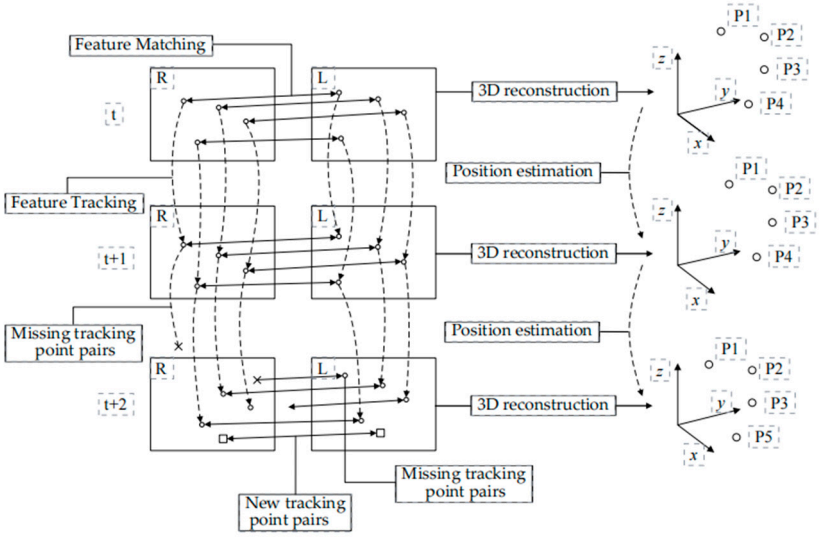



**Figure 1.** Block diagram of stereo vision odometer system.

### *2.2. ORB algorithm parameters selection*

ORB features are used to reduce the computation time and mismatch rate [17]. It is necessary to select the appropriate scale parameter in openCV with the number of s pyramid layers N. Assuming that the original map is at layer 0, the scale of layer i is:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



$$S_i = s^i \tag{1}$$

where s is the initial scale. Then the image size of the ith layer is:

$$S_i = \left(\frac{H}{s_i}\right) \times \left(\frac{W}{s_i}\right) \tag{2}$$

$H \times W$ is the size of the original image. From the above equation, it can be seen that the scale parameter determines the size of each layer of the pyramid, and the larger the scale parameter is, the smaller the image of each layer is, i.e., the scaling ratio is approximately larger. In order to select a suitable combination of parameters, this paper conducts experiments under different parameter combinations using 05 image sequences from the KITTI database.

The results of the parameter comparison are shown in Figure 2. The vertical coordinates represent the variation of the number of pyramid layers from 2 to 8, and the horizontal coordinates represent the variation of the scale parameter from 1.2 to 1.8. The color in each square in the figure then represents the size of the corresponding result. The total time from extraction to matching of ORB features with different combinations of parameters is given in Figure 2(a). From the results, it is seen that the computation time increases as the number of pyramid layers increases, while it decreases when the scale parameter becomes larger. This indicates that the image size of each layer becomes smaller after the scale parameter becomes larger, which leads to a shorter time for computing feature points on the image.

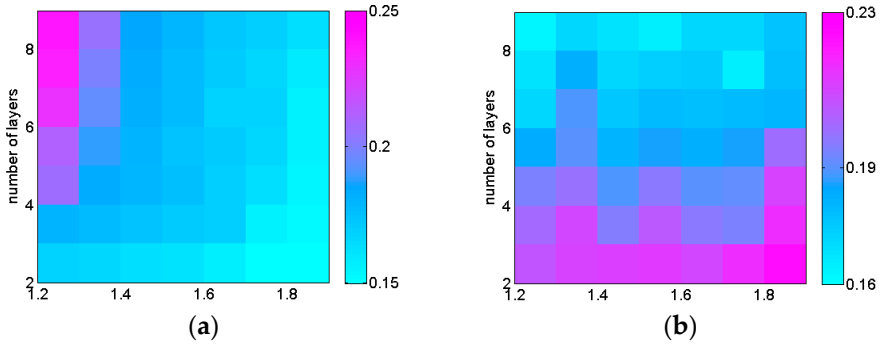



**Figure 2.** Calculation time and mismatch rate under different parameter matching. (a) Calculation time / 10-2s; (b) False match rate.

The false match rate when matching ORB feature points with different combinations of parameters is given in Figure 2(b). From the results, it is seen that the false match rate decreases significantly when the number of pyramid layers increases, which indicates that the feature points can ensure a good matching correct rate when the number of pyramid layers is large. Similarly, when the scale parameter becomes larger, there is a small increase in the false match rate, which indicates that the false match of feature points increases when the image size of each layer in the pyramid becomes smaller.

Combined with the above analysis, the scale parameter is chosen 1.8 and the number of image pyramid layers is chosen 8 when ORB features are used in this paper under the comprehensive consideration of computational efficiency and correct matching rate.

## 3. Improvement of ORB features

### *3.1. Calculation of different texture area weights*

In the actual extraction process of ORB features, it is necessary to divide the image into several smaller regions for extracting feature points, while the number of feature points in each region is consistent so that the calculation results can be guaranteed to be more accurate. The main steps are as follows [18]:

1. segmentation of the image;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1

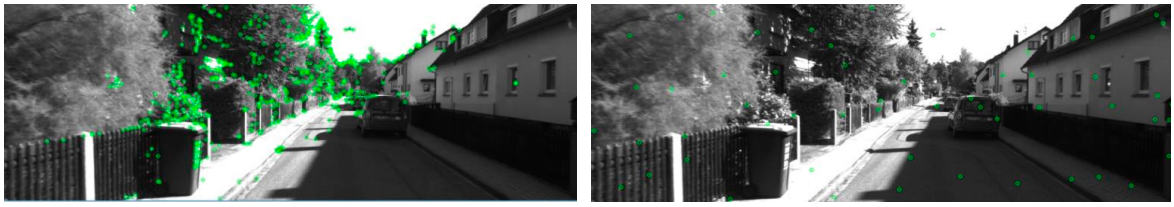


2. tracting feature points;

3. axing the extraction condition and extracting again if the number of feature points in the region is less than the minimum threshold;

If the number of features is greater than the maximum threshold, select a few of them with the largest Harris response value, and discard the rest.

Figure 3(a) image resolution is 1226*370, which is the road screen when the 05 image sequence of the KITTI dataset is not a region. As seen in the figure, the feature points are mainly concentrated in the plants as well as the outlines of the houses, which not only leads to mismatching and reduces the correct matching rate; but also makes the calculation results cause large errors.



(**a**) (**b**)

**Figure 3.** Features Distribution. (a) Distribution of original features; (b) Region segmentation results.

First, the image is segmented so that its feature points can be distributed as evenly as possible in the graph. For an image of original size $W \times H$, given the segmentation coefficients $s_w$ , $s_w$ for width and height, the width and height of the image are divided equally as follows:

$$n_w = \frac{W}{s_w}, n_h = \frac{W}{s_h} \tag{3}$$

The FSAT parameter for the first feature extraction in each region after segmentation is 30, and if no feature points are extracted, the parameter is changed to 3 and extracted again. A total of 1862 feature points were extracted, and the result shows that the extracted feature points are more evenly distributed in the image after region segmentation. The next step is to filter the feature points according to their Harris response values in each region and keep the points with the largest response values in each region, and the results are shown in Figure 3(b).

The above screening process results in the retention of only the local optimums because the response values of feature points in each region are compared. The relationship between the response values of feature points before and after screening is given in Figure 4. The blue curve in the figure indicates the response values of all 1862 feature points, and the red dots indicate the response values of the final feature points obtained after the screening. From the figure, it can be seen that a considerable number of feature points are local response value extremes, but their response values are still low in the whole, which indicates that the corner point nature of these points is not obvious compared with other points.

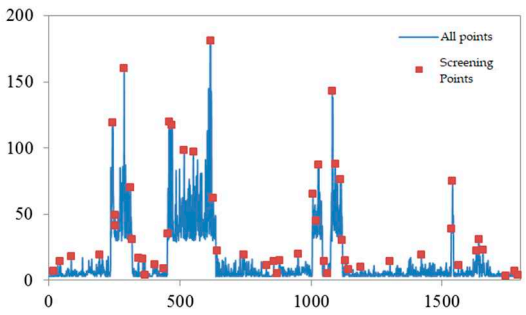



**Figure 4.** ORB Feature Harris Response Value.

While point feature homogenization ensures that feature points are distributed as evenly as possible, it also makes the corner point properties of some feature points poor. This method works

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



well in the case of rich textures; less rich environmental textures lead to poor matching of feature points. To solve the above problem, the image regions are differentiated [19]. For the image $I(x,y)$, calculate the matrix:

$$G = \sum \left( \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right)^T \left( \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) \quad (4)$$

The two eigenvalues of the matrix G represent the texture information of the region. When both eigenvalues are larger, it means that the region is a high-texture region, and vice versa, it means that the region is a low-texture region [20].

For different texture regions, different weights are given. That is, the weights should be small for low-texture areas of the image, while for high-texture areas of the images, the weights should be large. Definition of weights:

$$w = \min\left(eigenvalue(G)\right) \quad (5)$$

The weight values are determined by the grayscale gradient of the pixel points in the local region of the image. This results in feature points with weights that are uniformly distributed over the image, ready for feature tracking and motion estimation.

*3.2. Keyframe-based predictive motion model*

Stereo-matching aims to find the corresponding projection points of the same spatial point in images acquired from different viewpoints [21]. The parallel binocular vision system uses a polar constraint to perform feature matching between the left and right images. From the pair-pole geometry, let the projection points of the same spatial point P on the left and right images be P1 and P2, then the corresponding point P2 of the point P1 must be on the polar line l2 corresponding to P1. For the parallel binocular system, for the same spatial point, the polar line is on the same line; that is, it is P2 on the extension line of the polar line where P1 is located so that when searching for the matching point, it is only necessary to search in the domain of the polar line.

For the feature matching problem of inter-frame images, the robustness and real-time performance of the visual odometry are not guaranteed if we rely only on the unique constraint to reduce the error, and the method of estimating the motion model to narrow the search range is currently used in the front and back frame feature point matching to solve the above problem [22]. The method is to estimate the motion model of the system based on the images at moment t-1 and moment t. The position of the feature points in the image at moment t in moment t+1 is calculated under this motion model, and the best matching points are searched around this position.

However, in the above method, the overlap between adjacent frames increases for slower vehicles, resulting in almost no change in the projection of feature points, leading to high sensitivity of the system to errors. In this paper, we propose to use keyframes for motion model estimation to solve this problem. These keyframes are characterized by easy identification of feature points between adjacent keyframes, and the mean Euclidean distance between the current frame and the 3D coordinates of all matching points of the previous keyframe are considered as keyframes only when they are within a certain threshold.

$$d_{\min} < d_{k_i,k_{i-1}} < d_{\max} \quad (6)$$

where $d_{\min}$ - the minimum value of the distance threshold;

$d_{\max}$ - the maximum value of the distance threshold;

$d_{k_i,k_{i-1}}$ - the mean value of the Euclidean distance of the 3D coordinates of all matching points of the ith keyframe and the i-1st keyframe.

The specific steps are as follows:

Let the first frame of the input be the reference frame, and the subsequent consecutive frames are calculated with the selected reference frame in Euclidean distance until the one that meets the condition is the current keyframe. After using the current frame, the above operation is repeated to find all the keyframes. As shown in Figure 6, T0 indicates the reference keyframe and T1 indicates

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



the current keyframe. The motion calculated from the two keyframes is used to estimate the motion model of the current frame and the next frame, and this motion model is used to calculate the position of the feature points in the image at moment t in moment t+1 and the best matching points are cycled around that position early.

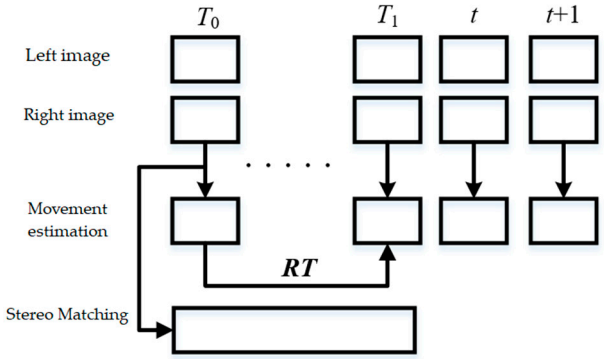



**Figure 6.** Motion model estimation based on keyframe.

*3.3. 3D reconstruction*

The 3D reconstruction is first performed using the matched pairs of feature points on the image, and then a second projection, called reprojection, is performed using the coordinates of the computed 3D points and the computed camera matrix. For a three-dimensional point P, the projection is:

$$d_i p_i = K\left(RP_i + t\right) \tag{7}$$

Measurement errors are always inevitable due to the imperfect accuracy of measuring instruments and the influence of human factors and external conditions. Therefore, there is a certain projection error between the projection points, which is called reprojection error, referring to the difference between the projection and reprojection of the real three-dimensional space points on the image plane, and in order to deal with the problem of errors in these projection points, the number of observations is often more than the number of observations necessary to determine the unknown quantity, that is, to make redundant observations.

Redundant observations can also cause contradictions between observation results, and these contradictions require optimization of the model to find the most reliable results of the observed quantities and to evaluate the accuracy of the measurement results [25]. The reprojection error of the point feature is calculated as follows:

$$e_p = p - h\left(\xi, P\right) \tag{8}$$

By constructing a least squares problem with the reprojection error of all points as a cost function, Eq:

$$\xi = \arg\min_{\xi} \sum_i \left\| p_i - \frac{1}{d_i} K \exp\left(\xi^{\wedge}\right) p_i \right\|^2 \tag{9}$$

For the calculation of the minimized reprojection error, the texture weight values of the feature points are added:

$$\xi = \arg\min_{\xi} \sum_i w \left\| p_i - \frac{1}{d_i} K \exp\left(\xi^{\wedge}\right) p_i \right\|^2 \tag{10}$$

Before calculating the least squares optimization problem, it is necessary to know the derivative of each error term with respect to the optimization variables, i.e., linearization:

$$e\left(x + \Delta x\right) \approx e\left(x\right) + J\Delta x \tag{11}$$

When the pixel coordinate error $e$ is two-dimensional and the camera pose $x$ is six-dimensional, $J$ is a 2 × 6 matrix. Transforming to the spatial point sitting under the camera coordinates is marked as $P'$, taking out its first three dimensions:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



$$P' = \left(\exp\left(\xi^{\wedge}\right)P\right)_{1:3} = \left[X', Y', Z'\right]^T \quad (12)$$

Then the camera projection model is:

$$du = KP' \quad (13)$$

Eliminating d yields:

$$u = f_x \frac{X'}{Z'} + c_x, v = f_y \frac{Y'}{Z'} + c_y \quad (14)$$

Consider the derivative of the change in $e$ with respect to the amount of perturbation:

$$\frac{\partial e}{\partial \delta\xi} = \lim_{\delta\xi \to 0} \frac{e\left(\delta\xi \oplus \xi\right)}{\delta\xi} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta\xi} \quad (15)$$

Where $\oplus$ denotes the left multiplicative perturbation on the Lie algebra. With the relationship between the variables obtained, it is deduced that:

$$\frac{\partial (TP)}{\partial P'} = (TP)^{\odot} = \begin{bmatrix} I & -P'^{\wedge} \\ 0^T & 0^T \end{bmatrix} \quad (16)$$

By taking the first 3 dimensions in the definition of $P'$ and multiplying the two terms together, we obtain the 2 × 6 Jacobi matrix:

$$\frac{\partial e}{\partial \delta\xi} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{f_x X'}{Z'^2} & \frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \quad (17)$$

This Jacobi matrix describes the first-order variation of the reprojection error with respect to the Lie algebra of camera poses. For the derivative of e with respect to $P$ at $e$ spatial point:

$$\frac{\partial e}{\partial P} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial P} \quad (18)$$

Regarding the second item, by definition:

$$P' = \exp(\xi^{\wedge})P = RP + t \quad (19)$$

Then:

$$\frac{\partial e}{\partial P} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} R \quad (20)$$

So, the two derivative matrices of the observed camera equations with respect to the camera pose and feature points are obtained.

## 4. Experimental verification

### *4.1. System Validation*

The visual odometer system built in this paper was experimented with using the KITTI dataset, which uses data obtained from GPS and inertial guidance system measurements as the reference path, the image acquisition frequency is 10Hz, the image resolution is 1241*376, and the camera parameters are shown in Table 1:

**Table 1.** Binocular camera parameters.

| Focal length/mm | Coordinates of main point | Aberration factor | Baseline /m |
|---|---|---|---|
| 718.86 | (607.19,185.22) | 0.00 | 0.54 |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



Because different distance thresholds yield different keyframe intervals, it is necessary to know and find the effect of different keyframe intervals on the accuracy of the system. The KITTI dataset 01 image sequence has a total of 1101 frames of binocular images, and the statistics of different keyframe rates are shown in Table 2 for five sets of experiments; 05 sequence has a total of 2761 frames, and eight sets of experiments, and the statistics of different keyframe rates are shown in Table 3.

**Table 2.** Key frame interval statistics of sequence 01.

| Serial number | Number of frames | Key Frame Count | Key Frame Rate(%) |
|---|---|---|---|
| 1 | 1101 | 66 | 5.99 |
| 2 | 1101 | 88 | 8.00 |
| 3 | 1101 | 110 | 10.00 |
| 4 | 1101 | 133 | 12.05 |
| 5 | 1101 | 167 | 15.15 |

**Table 3.** Key frame interval statistics of sequence 05.

| Serial number | Number of frames | Key Frame Count | Key Frame Rate(%) |
|---|---|---|---|
| 1 | 2761 | 236 | 8.55 |
| 2 | 2761 | 277 | 10.03 |
| 3 | 2761 | 312 | 11.30 |
| 4 | 2761 | 358 | 12.97 |
| 5 | 2761 | 410 | 14.85 |
| 6 | 2761 | 456 | 16.52 |
| 7 | 2761 | 495 | 17.93 |
| 8 | 2761 | 534 | 19.34 |

The average translation error and rotation error of the system at different keyframe intervals for 01 sequence and 05 sequence are counted in Figure 7(a) and (b). It can be seen from the results that the average translation error and rotation error have the same trend. The error of the system first decreases and then increases as the key frame rate becomes larger, mainly because the keyframe used becomes less effective when the interval is small, and when the interval is too large, the use of uniform motion. The actual motion of the vehicle cannot be accurately described, and the effect is not very good. From the statistical results, the keyframe rate should be selected at 10%-12%.

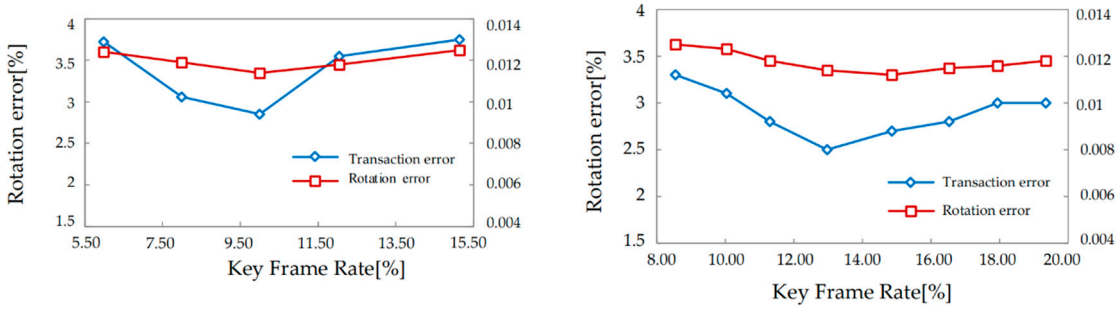



**Figure 7.** Comparison of errors at different key frame rates. (a)KITTI_01; (b)KITTI_05.

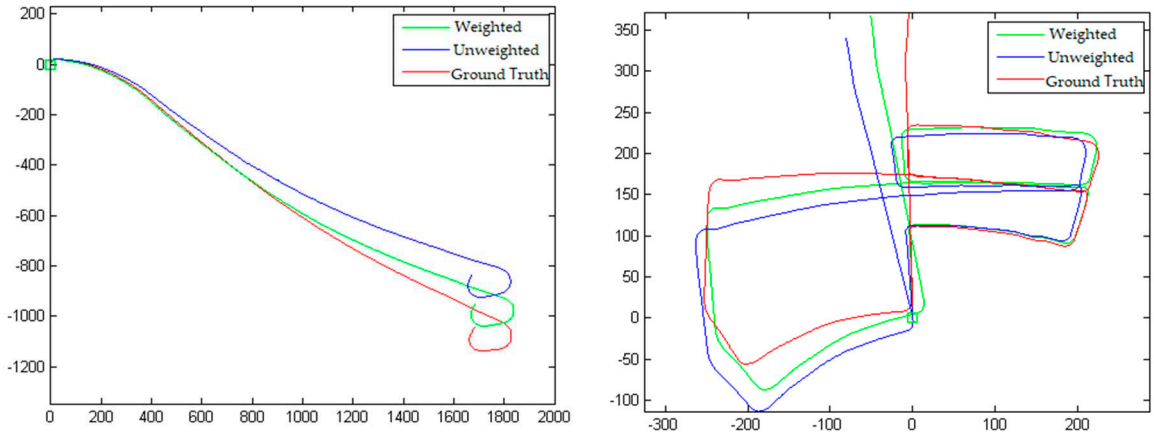
*4.2. Verify texture weighting impact*

The computational results of 01 and 05 image sequences without weighting (VISO2) and with weighting are compared in Figure 8, where the coordinate unit is m. The environment of 01 image

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



sequence is a highway and that of 05 image sequence is a small highway. The red path represents the groundtruth, the blue path is the calculation result when there is no weight, and the green path is the calculation result when there is weight. From the figure, it can be seen that the calculated results with weights are better than those without weights in both experiments, which initially verifies the effectiveness of this method.

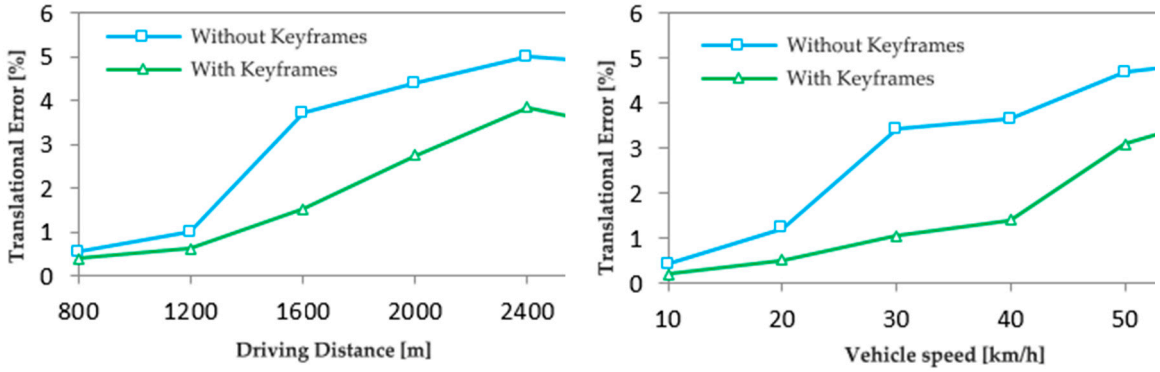



(**a**) (**b**)

**Figure 8.** Comparison of Dataset Results. (a)KITTI_01; (b)KITTI_05.

*4.3. Verification of keyframes*

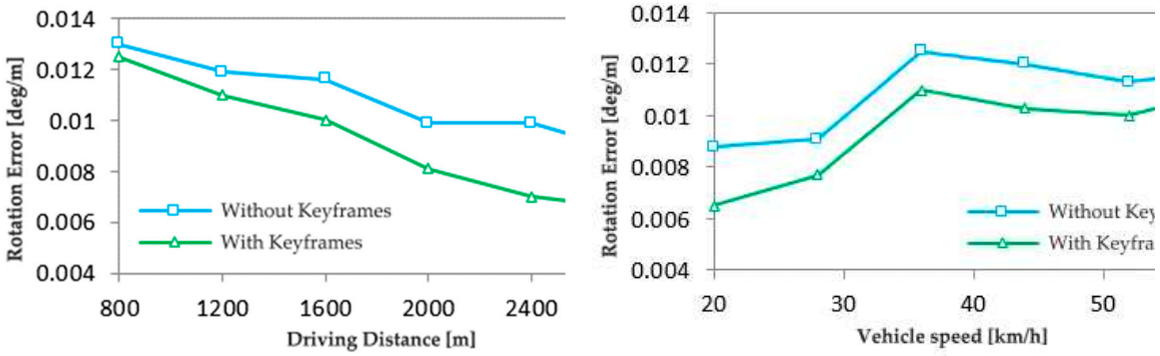
Comparing the calculation results with and without keyframes, Figures 9 and 10 give a comparison of the translation error and rotation error of the 01 and 05 image sequences. From the experimental results, the average translation error of the 01 image sequence calculated with keyframes in Figure 9 is 2.8%, the maximum translation error is 3.8%, the average rotation error is 0.0095deg/m, and the maximum rotation error is 0.0125deg/m. The average translation error for the 05 image sequence calculated with keyframes in Figure 10 is 2.2%, the maximum translation error is 2.9%, the average rotation error is 0.0087deg/m, and the maximum rotation error is 0.0125deg/m. The method of adding keyframes for inter-frame feature matching greatly reduces systematic error at low speeds.




(**a**)

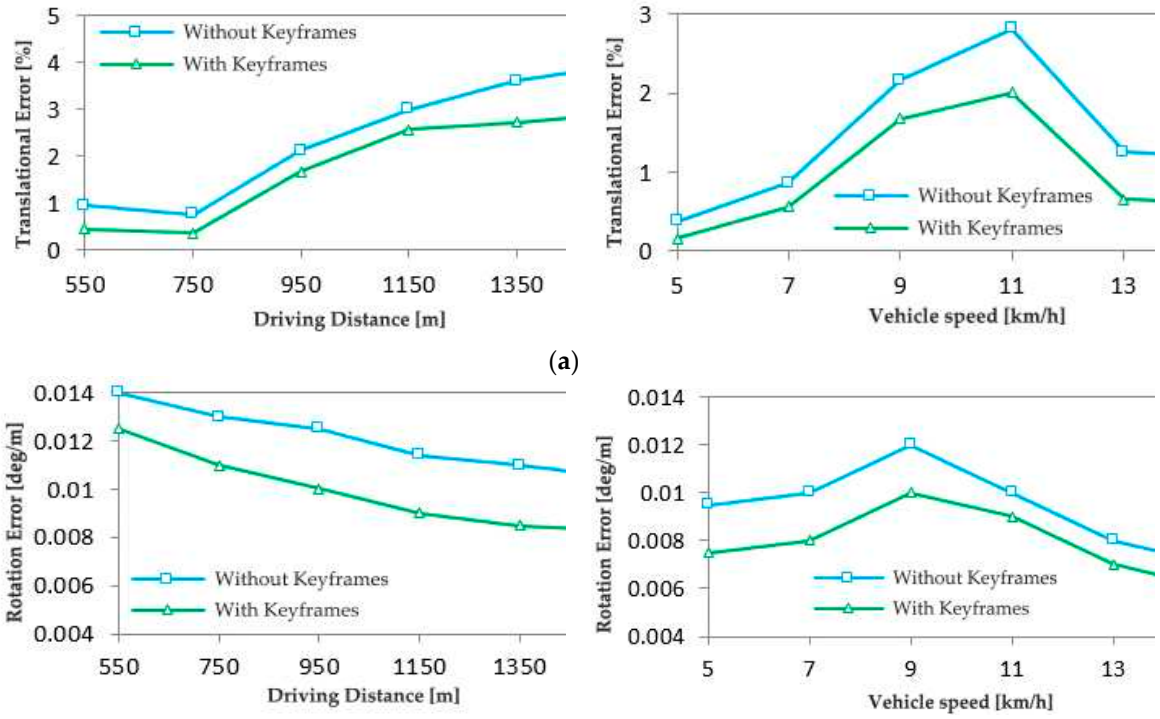
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1






(**b**)

**Figure 9.** Error contrast diagram of sequence 01. (a) Translation error; (b) Rotation error.




(**a**)

(**b**)

**Figure 10.** Error contrast diagram of sequence 05. (a) Translation error; (b) Rotation error.

The reason for the gradual increase in positioning error of the above two methods is that visual odometry is a relative positioning method, so the error of the system increases with mileage.

To further verify the performance of the system, a comparative analysis was done with the current common VISO2[23], VISO2+GP[24], and TGVO[25] open-source systems, and the system performance was described using a boxplot of the relative positioning error. The upper and lower sides of the rectangular box in the boxplot represent the upper and lower quartiles of the data, the end lines of the upper and lower extensions of the rectangular box represent the maximum and minimum observations, the horizontal line in the rectangular box represents the median of the data, and the outlier points are represented by the scatter points on the outside of the observations. Comparative experiments were conducted on four data sets, KITTI_00, KITTI_01, KITTI_05, and KITTI_007, and the results are shown in Figure 11.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 July 2023 doi:10.20944/preprints202307.1477.v1



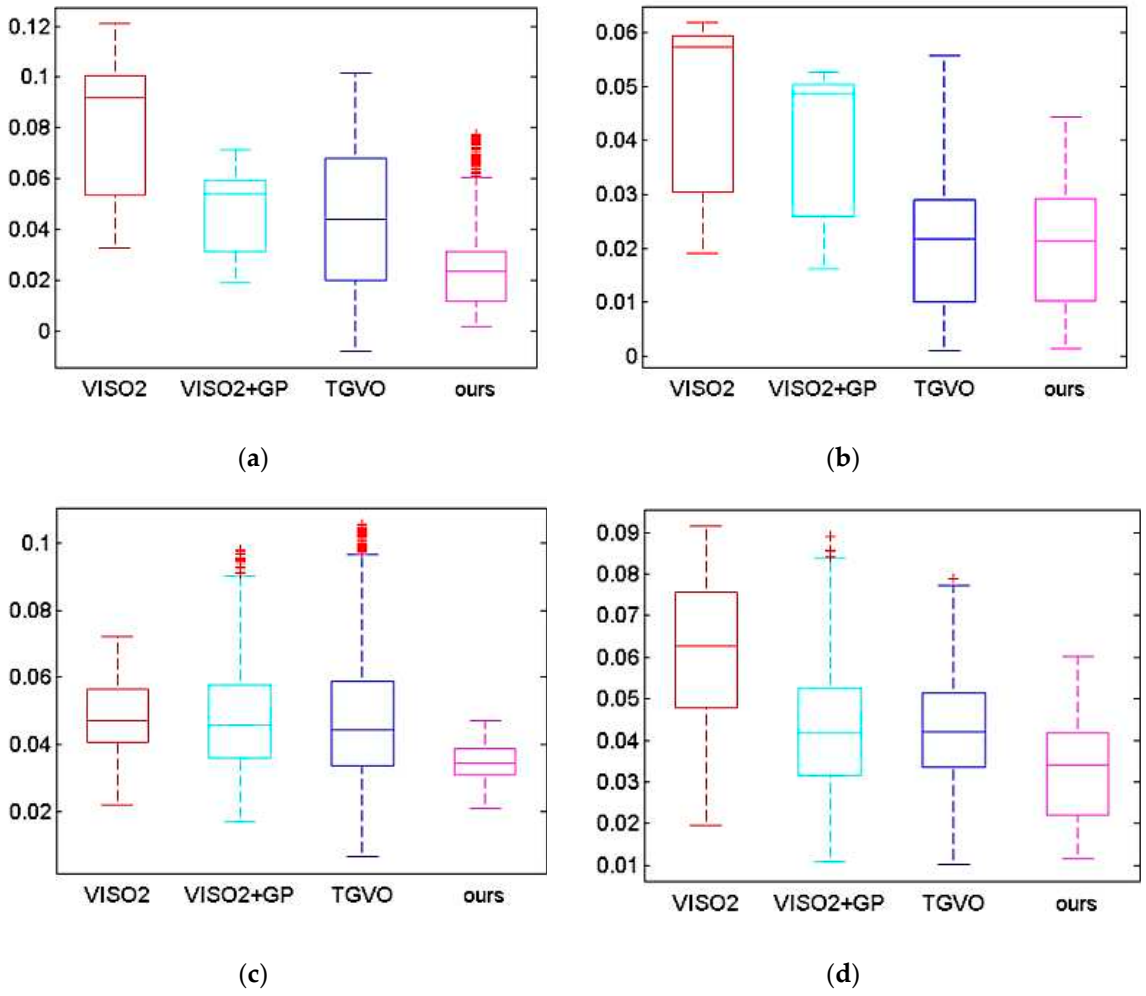



**Figure 11.** Relative error contrast diagram. (a)KITTI_00; (b)KITTI_01; (c)KITTI_05; (d)KITTI_07.

As seen from the results in Figure 11, the visual odometers proposed in this paper all show good performance with average relative errors of 2.3%, 2.2%, 3.5% and 3.4%, respectively. In Table 5, the statistics of the time used by the visual odometer system for each main process in the calculation process in ms, Min represents the shortest time used, Max represents the longest time used, and Avg represents the average time used for each frame.

**Table 4.** Algorithm calculation time statistics table.

| Time(ms) | Min | Max | Avg |
|---|---|---|---|
| Feature extraction and matching | 14.6 | 34.6 | 20.7 |
| 3D reconstruction | 5.1 | 12.6 | 8.5 |
| Movement estimation | 2.9 | 10.7 | 6.4 |
| Total time | 23.9 | 52.3 | 35.6 |

The above experiments demonstrate that the point feature-based visual odometry proposed in this paper has some improvement in localization accuracy in the dataset experiments compared with several other common open-source VO systems.

## 5. Conclusions

In response to the problem that the feature point method leads to poor stability of vehicle driving in complex environment, the visual odometer based on improved ORB features is proposed to obtain better results.

1. Propose the calculation of weights for different texture regions, with high matching weights for high texture regions and low matching weights for low texture regions, so that feature points can be evenly dispersed throughout the image and better matching results.
2. Using the predicted motion model of keyframes, i.e., the motion of feature points between adjacent keyframes is obvious so that the system is more stable and less prone to errors when the vehicle is driving at slow speed. The test using KITTI dataset shows that the key frame rate reaches 10%-12% error minimum. When comparing the translation and rotation errors with and without keyframes using the KITTI dataset, the presence of keyframes clearly shows a reduction in translation and rotation errors.
3. This system compares the performance of the VISO2, VISO2+GP, and TGVO verification systems, and the comparison experiments were conducted on four sequences of the KITTI dataset, and the errors were lower than those of the above three systems.

**References**


1. Li, S.; Wang, G.; Yu, H.; Wang, X. Engineering Project: The Method to Solve Practical Problems for the Monitoring and Control of Driver-Less Electric Transport Vehicles in the Underground Mines. World Electr. Veh. J. 2021, 12, 64. https://doi.org/10.3390/wevj12020064.
2. Boersma, R.; Van Arem, B.; Rieck, F. Application of Driverless Electric Automated Shuttles for Public Transport in Villages: The Case of Appelscha. World Electr. Veh. J. 2018, 9, 15. https://doi.org/10.3390/wevj9010015
3. Zhang C, Lei L, Ma X, Zhou R, Shi Z, Guo Z. Map Construction Based on LiDAR Vision Inertial Multi-Sensor Fusion. World Electric Vehicle Journal. 2021; 12(4):261.
4. Ma F W, Shi J Z, Ge L H et al. Research progress of monocular vision odometer for unmanned vehicles [J]. Journal of Jilin University (Engineering Edition), 2020, 50(03): 765-775.
5. Zeng Q H., Luo Y X., Sun K C., et al. A review on the development of SLAM technology for vision and its fused inertia[J]. Journal of Nanjing University of Aeronautics and Astronautics, 2022, 54(06): 1007-1020.
6. Zhou, F Y, Gu, P L., Wan, F. et al. Methods and techniques for multi-motion visual odometry[J]. Journal of Shandong University (Engineering Edition), 2021, 51(01): 1-10.
7. Campos C, Elvira R, JJG Rodríguez, et al. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM[J]. IEEE Transactions on Robotics,2020,37(6):1874-1890.
8. Z. Chen and L. Liu, Navigable Space Construction from Sparse Noisy Point Clouds[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 4720-4727.
9. B. Zhang and D. Zhu. A Stereo SLAM System with Dense Mapping[J]. IEEE Access, 2021,9: 151888- 151896.
10. D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld and H. M. Gross. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis [C]. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021: 13525-13531.
11. Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 3(15): 1147-1163.
12. Bei Q, Liu H, Pei Y, Deng L and Gao W. An Improved ORB Algorithm for Feature Extraction and Homogenization Algorithm[C]. 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI), 2021: 591-597.
13. Chen J S, Yu L L, Li X N, et al. Loop detection based on uniform ORB[J]. Journal of Jilin University (Engineering and Technology Edition), 2022: 1-9.
14. Yao, J J, Zhang, P C, Wang, Y. et al. An algorithm for uniform distribution of ORB features based on improved quadtrees[J]. Computer Engineering and Design, 2020, 41(06): 1629-1634.
15. Zhao C. Research on the uniformity of SLAM feature points and the construction method of semantic map in dynamic environment[D]. Xi'an University of Technology, 2022.
16. L. Lai, X. Yu, X. Qian and L. Ou.3D Semantic Map Construction System Based on Visual SLAM and CNNs[C]. IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, 2020: 4727-4732.
17. R. Ranftl, A. Bochkovskiy and V. Koltun. Vision Transformers for Dense Prediction[C]. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021: 12159-12168.
18. Li G, Zeng Y, Huang H, et al. A Multi-Feature Fusion Slam System Attaching Semantic In-Variant to Points and Lines[J]. Sensors, 2021, 21(4): 1196.
19. Fan X N, Gu Y F, Ni J J. Application of improved ORB algorithm in image matching[J]. Computer and Modernization, 2019, 282(2): 5-10.
20. H. Xu, C. Yang and Z. Li. OD-SLAM: Real-Time Localization and Mapping in Dynamic Environment through Multi-Sensor Fusion[C]. 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), 2020:172-177.

21. Xu H, Yang C, Li Z. OD-SLAM: Real-Time Localization and Mapping in Dynamic Environment through Multi-Sensor Fusion[C]. 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM). 2020(7): 75604-75614.
22. L. Zhao, Z. Liu, J. Chen, W. Cai, W. Wang and L. Zeng. A Compatible Framework for RGB-D SLAM in Dynamic Scenes[J]. in IEEE Access, 2019, 7: 75604-75614.
23. Geiger A, Ziegler J, Stiller C. StereoScan: Dense 3d reconstruction in real-time[J]. IEEE Intelligent Vehicles Symposium, 2011, 32(14): 963-968.
24. Geiger A. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]// Computer Vision and Pattern Recognition. IEEE, 2012:3354-3361.
25. Kitt B, Geiger A, Lategahn H. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme[C]// Intelligent Vehicles Symposium. IEEE, 2010: 486-492.