Monocular Visual Odometry Scale Recovery Using Geometrical Constraint

Conference Paper · May 2018

DOI: 10.1109/ICRA.2018.8462902

5 authors, including:

Xiangwei Wang
Carnegie Mellon University
9 PUBLICATIONS **279** CITATIONS
SEE PROFILE

Zhang Hui
Tongji University
3 PUBLICATIONS **36** CITATIONS
SEE PROFILE

Mingxiao Du
Tongji University
10 PUBLICATIONS **1,096** CITATIONS
SEE PROFILE

2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Monocular Visual Odometry Scale Recovery using Geometrical Constraint

Xiangwei Wang Hui Zhang Xiaochuan Yin Mingxiao Du and Qijun Chen

***Abstract*— Scale recovery is one of the essential problems for monocular visual odometry. The camera height is usually used as an absolute reference to recover the scale. In this case, the precision of scale recovery depends on the accuracy of the road region detection and road geometrical model calculation. In previous works, road detection and road geometrical model calculation are solved sequentially: the road geometrical model calculation is based on the road detection and the road region detection is based on the color information. However, the color information of a road is not stable enough. In the proposed method, the estimated road geometrical model is taken into consideration to detect the road region as a feedback. Therefore, the road region detection and road geometrical model estimation can benefit each other. Delaunay Triangulation method is used to segment an input image to many triangles with the matched feature points as vertices. Every triangle region is classified as a road region or not by comparing their geometrical model with that of the road and the road geometrical model is updated online. We evaluate our visual odometry scale recovery method on the KITTI dataset and the results show that our method is achieving the best performance among all existing monocular visual odometry scale recovery methods without additional sensors.**

I. Introduction

Visual odometry(VO) can be applied for visual localization and motion estimation of a robot, which are key and important problems for mobile robots, including autonomous ground vehicles, unmanned aerial vehicles and so on. VO has been researched for more than thirty years [1], but there are still lots of challenges such as challenging environments [2] (including large-scale environments, dynamic environments and featureless environments [3]) and for monocular VO, the scale problem is one of the central problems [4].

What is the scale problem? Because the monocular camera loses one dimension at the time of imaging, no depth of each pixel is available. As a result, monocular VO cannot calculate the absolute scale of the robot motion. The unknown scale can also lead to scale drift. Monocular VO scale problems cannot be solved without other absolute reference scale information and the information can be obtained from other sensors (such as IMU and Lidar [5]), the baseline distance (binocular camera), camera height (road model) and offline training (learning-based method [6]).

* This work was supported in part by the National Natural Science Foundation of China (Key Program) (Grant No. 61733013), and Basic Research Project of Shanghai Science and Technology Commission (Grant No. 16JC1401200).

Xiangwei Wang, Hui Zhang, Xiaochuan Yin, Mingxiao Du and Qijun Chen are with Department of Control Science and Engineering, Tongji University, NO. 4800 Caoan HW., Shanghai, China wangxiangwei.cpp@gmail.com

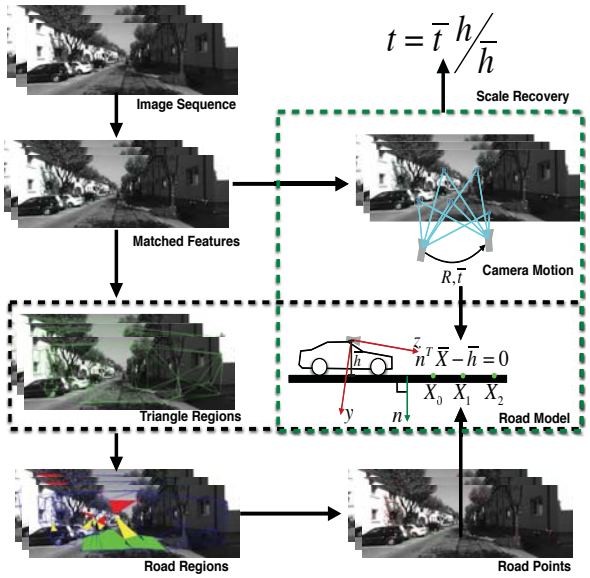


Fig. 1. This figure shows the structure of the proposed monocular VO scale recovery algorithm. The initial VO process calculates the initial ego-motion $R, \bar{t}$ and $\bar{t}$ is not in the absolute scale. The road pitch angle, which a central unit of road geometrical model, can be estimated from the ego-motion. The input frame is segmented into a set of triangles by Delaunay Triangulation and each triangle is checked whether it is a road region by taking the road geometrical model into consideration. The road feature points are used to calculate the road model and the scale is recovered taking the absolute camera height into account.

Among all kinds of information, for an autonomous ground vehicle, the height of a mounted camera is the most convenient and inexpensive one because it doesn't need other sensors nor offline training. Besides, the absolute camera height remains stable while the vehicle is running on the road. When taking the camera height as a reference, the scale recovery problem is converted to road geometry estimation in the relative scale. The first task of that is to detect the road region. Previous methods solved the road detection problem and road model estimation problem separately. For the road detection problem, most methods [7], [8], [4] choose to assume a fixed region as the road region instead of detecting that. However, fixed road region based methods have two drawbacks. The first one is that the fixed region may not be the road region while there is a car or something else in front of the camera; besides, the fixed region is only a small part of the road region and the number of feature points on the fixed region is limited. In [9], segmentation and trained classifier proposed in [10] is used to detect the road but this kind of method is not robust to unfamiliar

circumstances. [11] is more robust by updating the road classifier online. All the previous classifier based methods focus on the color information of a road, which has much uncertainty. We explore to take estimated road geometrical model into consideration to detect the road region as a feedback. The road geometrical model is updated online by considering the detected road region. Therefore, the road region detection and road geometrical model estimation can benefit each other.

In this work, we use Delaunay Triangulation [12] to segment the input frame into many triangles with the matched feature points as vertices. After that, the geometrical model of each triangle is calculated and then used to check the possibility of every triangle region to be a part of road region by comparing their geometrical model with that of the road. Our algorithm structure is shown in Fig.1. The contributions of our paper are as follows:

1) The main contribution of this work is that we propose a novel road-based solution for monocular VO scale recovery. We combine the road region detection and the road geometrical model calculation into one problem: detecting the road region based on the road geometrical model and updating the road model based on the detected road region.
2) Secondly, the initial ego-motion without absolute scale is used to estimate a coarse road pitch angle to select the road region in the beginning.
3) Besides, we explore filtering the road model to make the algorithm more robust and based on the comparison experiment, we discover that median filtering method is suitable and find out the suitable filtering size.
4) Last but not least, this work puts forward the precision of monocular visual odometry, and the proposed method is quite simple to realize.

This paper is organized as follows: firstly, background and notion are presented in Section II. Our approach for scale recovery is given in Section III. The experimental results on the KITTI dataset are given in Section IV. Finally, the paper ends with a conclusion in Section V.

II. Background and Notation

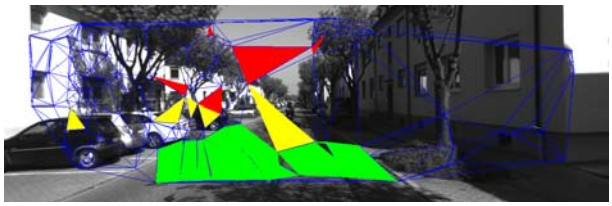
The objective of monocular visual odometry is to obtain the current camera pose $\mathbf{P}_t$ relative to the initial camera pose $\mathbf{P}_0$. In the beginning, two frames $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$ are used to calculate the camera motion $\mathbf{T}$ and then we get the current pose by $\mathbf{P}_t = \mathbf{P}_{t-1} * T$. For the first two frames in the initialization period, no 3D point map exists. The commonest method is fundamental matrix method [13]

$$\mathbf{u}_{t-1}{}^T \mathbf{F} \mathbf{u}_t = 0 \tag{1}$$

where $\mathbf{F} = \mathbf{K}^{-1^\mathbf{T}}[\mathbf{t}]\mathbf{R}\mathbf{K}^{-1}$ is the fundamental matrix, $\mathbf{u}_{t-1}$ and $\mathbf{u}_t$ are matched feature positions in the frame $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ and $\mathbf{K}$ is the camera intrinsic parameter matrix. In this method, foundation matrix $\mathbf{F}$ is solved first, then $\mathbf{R}$ and $\mathbf{t}$ are obtained from $\mathbf{F}$ [14]. We can observe from the equation that if we multiply any scale parameter s to the obtained $\mathbf{t}$, the equation is always correct. It means that we can not identify the scale of translation vector $\mathbf{t}$. We denote the $\mathbf{t}$ without absolute scale by $\bar{\mathbf{t}}$.

(a) Matched Feature Points

(b) Triangulation Result



(c) Triangle Regions On Road

(d) Matched Feature Points On Road

Fig. 2. This figure shows the result of road region detection: Fig. 2(a) shows all the matched feature points, and in Fig. 2(b) the frame is segmented by Delaunay Triangulation. Then the road triangle region is marked as green in Fig. 2(c) and the road feature points and other feature points are marked as green and red in Fig. 2(d).

We cannot get the absolute scale of $\bar{\mathbf{t}}$ but we can try to keep the same scale for temporal consistency and this is relative scale. After obtaining the first camera motion, we can get 3D positions $\bar{\mathbf{x}}_s$ in relative scale of matched feature points. The following camera pose is calculated by the 3D map and current frames using PnP method [15] by solving

$$\mathbf{R}, \mathbf{t} = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_{\mathbf{x}_i, \mathbf{u}_i} \left| \frac{\mathbf{K}(\mathbf{R}\mathbf{x}_i + \mathbf{t})}{\mathbf{x}_i(3)} - \mathbf{u}_\mathbf{i} \right|. \tag{2}$$

This method can maintain the scale. However, as the error accumulates, the scale drift. The result of monocular visual odometry will be less accurate as shown in Fig. 6. Most monocular visual odometry or SLAM method such as DSO [16], LSD-SLAM [17], ORB-SLAM [18], SVO [19] do not consider the absolute scale but they all try to keep the same scale by bundle adjustment or loop closure.

Without the fusion of other sensors, the only method for

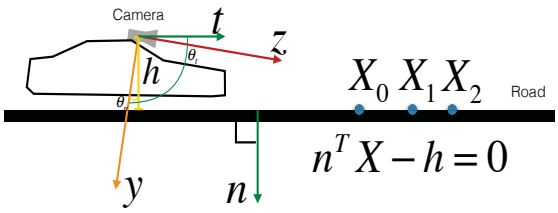


Fig. 3. If the vehicle is running on a flat road, the direction of t is vertical to the direction of road norm.

scale recovery is to employ the known scale information of the surroundings. In detail, we need to know the size l of something in absolute scale, and we calculate its size in the relative scale $\bar{l}$, and then the scale parameter $s = \frac{l}{\bar{l}}$. The translation vector with absolute scale is recovered $\mathbf{t} = s\bar{\mathbf{t}}$.

Throughout the paper, we will express matrices as bold capital letters ($\mathbf{R}$) and vectors as bold lower case letters ($\mathbf{t}$). All values in the form of ($\bar{\mathbf{t}}$) are in relative scale and the others are in absolute scale. We can identify skew-symmetric matrix of a vector as $[\mathbf{t}]$. t_i is the **i**th element in the vector $\mathbf{t}$ and R_{ij} is the **i**th row and **j**th column element in matrix $\mathbf{R}$. Besides the inside region and vertices of a triangle Tri are expressed as $\widetilde{Tri}$ and $\widehat{Tri}$ respectively.

III. Scale Recovery Approach

In this section, we will introduce the proposed monocular visual odometry scale recovery method. Because we take the absolute camera height as the scale reference, the key of scale recovery problem is to obtain the road geometrical structure $\mathbf{n}\bar{\mathbf{x}} - \bar{h} = 0$ and get the camera height $\bar{h}$ that in the same relative scale with the camera motion $\bar{\mathbf{t}}$. In the proposed method, the road region detection and road geometrical model calculation are processed iteratively: the road region is detected by considering the road geometrical model; then the road geometrical model is updated by the matched 3D feature points on the road region. They can benefit each other.

A. Road Region Detection

In this paper, the road geometrical structure constraint instead of color information is taken into consideration to detect the road region. The geometrical structure is more robust than color information because the color of a road may change but the geometrical remains stable relatively and the geometrical model is updated every frame.

The 3D coordinates of matched feature points are available after initial visual odometry process [18], and though the 3D coordinates are not in the real scale, they maintain the same geometrical structure with the coordinates in the real scale. Their pixel coordinate on frame $\mathbf{I}^t$ are noted as $\mathbf{u}_i^t$ $i = 1, 2, ..., n$ and n is the number of matched feature points in frame $\mathbf{I}^t$; and the corresponding 3D coordinate of each matched feature point is $\bar{\mathbf{x}}_i^t$. The road region detection is based on $\mathbf{u}_i^t$ and $\bar{\mathbf{x}}_i^t$.

Firstly, the 2D feature points set $\{\mathbf{u}_i\}$(the blue points in Fig. 2(a)) is segmented by Delaunay Triangulation [12] into a set of triangles $\{\mathbf{Tri}\}$ as shown in Fig. 2(b). Then we calculate the geometrical model (norm $\mathbf{n}_{t_i}$ and height $\bar{h}_{t_i}$) of the plane that each triangle region $\mathbf{Tri}_{t_i}$ belonging to, with the 3D coordinate of its vertices $\bar{\mathbf{x}}_{ti}$by solving

$$\begin{cases} \mathbf{n_{ti}}^T \bar{\mathbf{x}}_{ti_1} - \bar{h}_{ti} = 0 \\ \mathbf{n_{ti}}^T \bar{\mathbf{x}}_{ti_2} - \bar{h}_{ti} = 0 \\ \mathbf{n_{ti}}^T \bar{\mathbf{x}}_{ti_3} - \bar{h}_{ti} = 0 \end{cases} \tag{3}$$

and the solution of this equation is

$$\bar{\mathbf{n}}_{ti} = \begin{pmatrix} \bar{x}_{ti_{1_x}} & \bar{x}_{ti_{1_y}} & \bar{x}_{ti_{1_z}} \\ \bar{x}_{ti_{2_x}} & \bar{x}_{ti_{2_y}} & \bar{x}_{ti_{2_z}} \\ \bar{x}_{ti_{3_x}} & \bar{x}_{ti_{3_y}} & \bar{x}_{ti_{3_z}} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{4}$$

and

$$\begin{cases} \mathbf{n}_{ti} = \frac{\bar{\mathbf{n}}_{ti}}{\|\bar{\mathbf{n}}_{ti}\|} \\ \bar{h}_{ti} = \frac{1}{\|\bar{\mathbf{n}}_{ti}\|} \end{cases} \tag{5}$$

In Eq. 3 there are four variables and only there constrains, so this equation has infinite solutions. We add two more constrains to get unique solution: $\|n_{ti}\| = 1$ and $n_{ti_y} > 0$. In this case, $\bar{h} > 0$ means the triangle is below camera (we assume that the camera is mounted to look forward).

After the geometrical structure of each triangle region $\mathbf{n}_{t_i}\mathbf{x} - h_{t_i} = 0$ is obtained, we try to select the road regions among all triangle regions. The first criterion is that the road region should be below the camera, therefore the $\bar{h}_{t_i}$ should be positive and all regions with negative $\bar{h}_{t_i}$ will be eliminated.

For the remaining triangle regions, we check their norm and consider that the norm should be close to estimated road norm n_t that can be estimated by camera motion $\mathbf{R}$ and $\bar{\mathbf{t}}$. When the vehicle is running on the road, its motion vector $\bar{\mathbf{t}}$ is tangent to the road surface and orthogonal to road norm when its pitch rotation is little. As shown in Fig. 3, the estimated road norm's pitch angle $\theta_n' = \theta_t - \frac{pi}{2}$. The camera pitch angle's absolute value can be calculated from

$$|\theta_{\mathbf{R}}| = \begin{cases} |\arctan -\frac{\mathbf{R}_{32}}{\mathbf{R}_{33}}| & \text{if} \quad \mathbf{R}_{33} \neq 0 \\ \frac{\pi}{2} & \text{if} \quad \mathbf{R}_{33} = 0 \end{cases} \tag{6}$$

The motion patch can be obtained by

$$\theta_{\mathbf{t}} = \begin{cases} \arcsin -\frac{t_2}{|t|} & \text{if} \quad |t| \neq 0 \\ \text{NaN} & \text{if} \quad |t| = 0 \end{cases} \tag{7}$$

When $|t|$ is zero, the scale does not need to recover.

The pitch angle of each triangle region norm can be calculated by

$$\theta_{n_{t_i}} = \arcsin -\frac{n_{t_{i_2}}}{|\mathbf{n}_{t_i}|} \tag{8}$$

and they are compared with the estimated angle θ_n' and only triangle region with small error will be kept.

As the road model should be similar in subsequent frames, we compare the geometrical model of left triangle region with the road model calculated last time and select the

triangle regions with the similar geometrical model with road model. The distance of two norms is expressed as

$$d_n = |\arccos \mathbf{n} \cdot \mathbf{n}_{t_i}| \tag{9}$$

At last the feature points belong to the selected triangle regions are selected to estimate the road geometrical model. The details are described in Algorithm 1.

Algorithm 1: Road Region Detection by Feature Points Triangulation

Input: Feature points coordinates $\mathbf{u}_i^t$ and $\bar{\mathbf{x}}_i^t$, $i = 1,2,..,n$, initial camera motion $\mathbf{R}_{t-1}^t, \bar{\mathbf{t}}_{t-1}^t$ from time $t-1$ to time t

Output: The road region patches $\{Re_j^t\}, j = 1,2,...,p$ and road feature sets $\{f_k^t\}, k = 1,2,..,m$

Obtain the pitch angle θ_R from $\mathbf{R}$ by Eq. 6

Obtain the camera motion pitch angle θ_t from $\mathbf{t}$ by Eq. 7

The estimated road norm pitch angle is obatined $\theta_n' = \theta_t - \frac{pi}{2}$

Segment the $\mathbf{u}_i^t$ into a triangle set $\{\mathbf{Tri}\}_{t_i=0}^{t_n}$

Set a corresponding flag set $\{\mathbf{Flag}\}_{t_i=0}^{t_n}$of triangle set and $\{\mathbf{Flag}\}_{t_i} == true$ means that the region inside $\{\mathbf{Tri}_{t_i}\}$ is road region

for $t_i = 1$ *to* t_n **do**

  Set $\mathbf{Flag}_i = false$

  Obtain the norm $\mathbf{n}_{t_i}$ and distance $\bar{h}_{t_i}$ to camera optical center triangle $\mathbf{Tri}_{t_i}$ by solving Eq. 3-5

  if $\bar{h}_{t_i} > 0$ ***and*** $\left(|\theta_{t_i} - \theta_n'| < \frac{5\pi}{180}$ ***or*** $\theta_R > \frac{5\pi}{180}\right)$ **then**

    Obtain the distance between the road norm n and the norm of each triangle region $\mathbf{n}_{t_i}$ by Eq. 9

    if $|\theta_{t_i} - \theta| < \frac{5\pi}{180}$ **and** $|\frac{\bar{h}_{t_i} - \bar{h}_{last}}{\bar{h}_{last}}| < 0.2$ **then**

      Set $\mathbf{Flag}_i = true$

    end

  end

end

Set $\{\mathbf{Re}_j^t\} = \{\widetilde{\mathbf{Tri}_{t_i}^t}\}$ *where* $\left(\mathbf{Flag}_{t_i} == true\right)$

Set $\{\mathbf{f}_k^t\} = \widehat{\{\mathbf{Re}_j^t\}}$

return $\{\mathbf{Re}_j^t\}$, $\{\mathbf{f}_k^t\}$

B. Road Model Estimation and Scale Recovery

After we have obtained a set of feature points $\{\mathbf{Re}_j^t\}$ on the road and their 3D coordinates are available, the road model is ready to be calculated. Because the road feature points can be obtained accurately by the method in Section III-A, the noise is limited and the road geometrical model estimation process is pretty simple. We assume that road as a flat plane and its geometrical model can be expressed as

$$\mathbf{n}^T \mathbf{x} - h = 0 \tag{10}$$

where $\mathbf{n}$ is the road plane norm, and h is distance from camera to road plane.

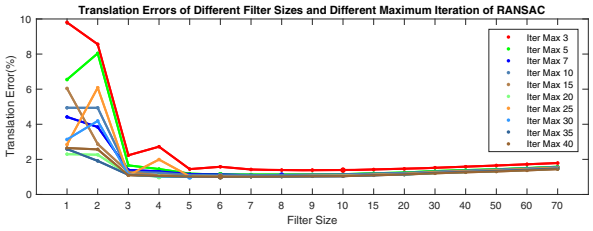
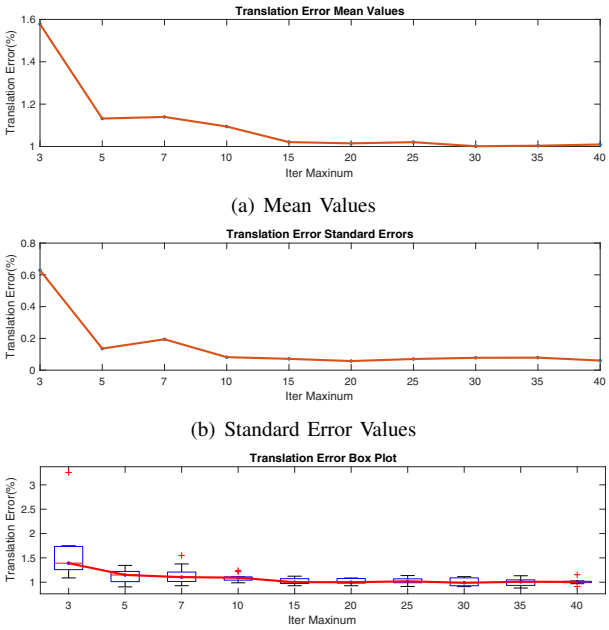


Fig. 4. We test our method with 10 different RANSAC maximum iteration numbers and 17 different filter sizes on KITTI dataset sequence 00. Each case is run 10 times and evaluated. This figure shows average translation errors.



(a) Mean Values

(b) Standard Error Values

(c) Box Plot

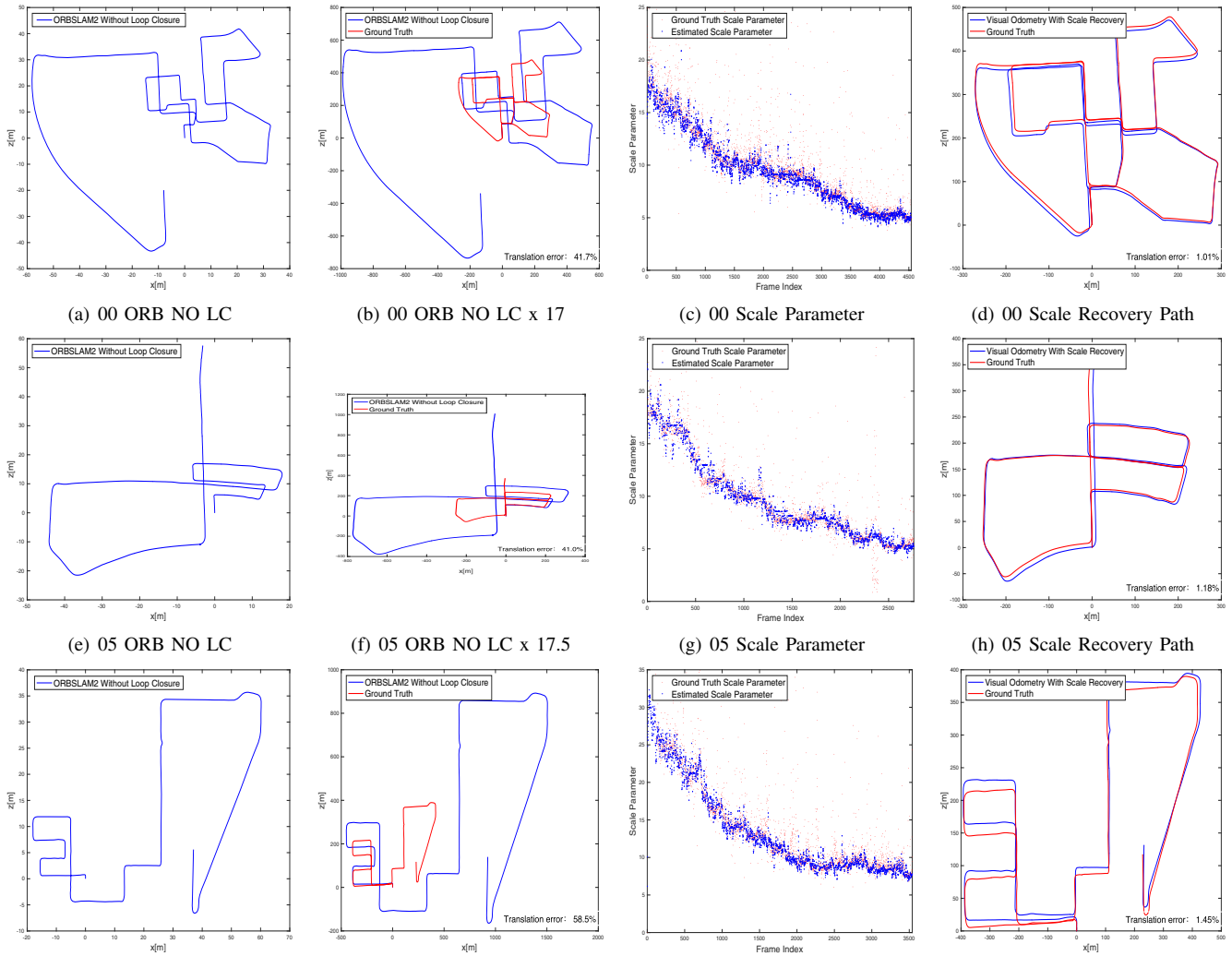
Fig. 5. We use median filter and set the filter size as 6 then run 10 times for each different RANSAC parameters on KITTI sequence 00. This figure shows the different mean and standard error values of different RANSAC maximum ieration numbers with the same filtering method in Fig. 5(a) and Fig. 5(b) respectively. Fig. 5(c) is the box plot of that.

Firstly RANSAC method [20] is used calculate a coarse road geometrical model and then a filtering method is used to remove camera height noise. Here if the amount of selected feature points is less than 12, we skip the RANSAC and keep the road geometrical model same as the previous one. In this paper, a median filtering method is used with good performance. The different performance with different parameters of RANSAC and filtering are compared in Section IV

The relative distance $\bar{h}$ between camera and road is obtained after the road model is obtained. We assume that the absolute distance h is available after the initial calibration and keeps stable. The scale parameter s can be derived by $s = \frac{h}{\bar{h}}$ and then the motion scale is recovered by $\mathbf{t} = s\bar{\mathbf{t}}$.

IV. Experiments

The performance of the proposed method is evaluated on the KITTI dataset [21] with evaluation metrics provided by [21] and [22]. This dataset consists 22 sequences including

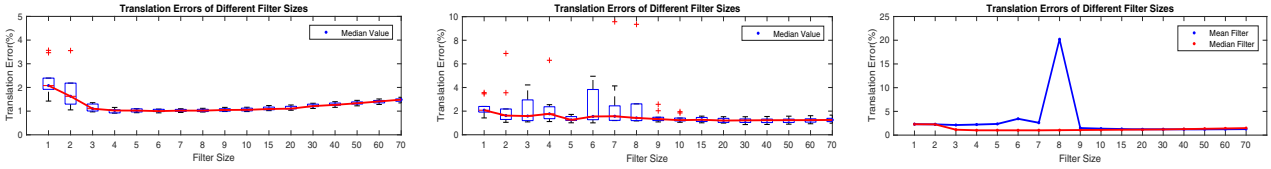


(a) 00 ORB NO LC (b) 00 ORB NO LC x 17 (c) 00 Scale Parameter (d) 00 Scale Recovery Path

(e) 05 ORB NO LC (f) 05 ORB NO LC x 17.5 (g) 05 Scale Parameter (h) 05 Scale Recovery Path

(i) 08 ORB NO LC (j) 08 ORB NO LC x 25 (k) 08 Scale Parameter (l) 08 Scale Recovery Path

Fig. 6. This figure shows the scale recovery performance on KITTI dataset sequences 00, 05 and 08. The three figures in the first column are the Monocular ORB-SLAM2 results without loop closure and they are not in the right scale. In the second column the motion obtained by Monocular ORB-SLAM2 are multiplied with three fixed scale parameters 17.0, 17.5 and 25.0 respectively but they still suffer scale drift problem. The figures in the third column are the scale parameters estimated by our proposed method and by multiplying them, the result can be in the right scale and with little scale drift.

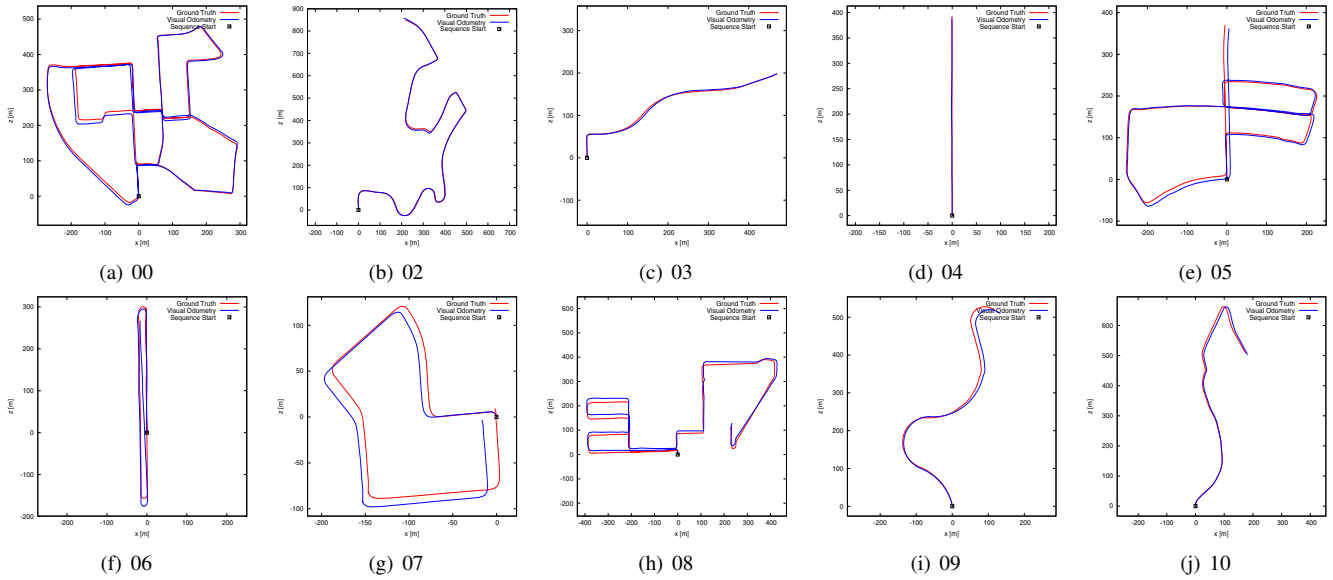


(a) Box Plot of Median Filter (b) Box Plot of Mean Filter (c) Comparasion of Median Filter and Mean Filter

Fig. 8. This figure shows the different performance of mean filter and median filter. Fig. 8(a) and Fig. 8(b) are the box plot of median filter and median filter respectively. In Fig. 8(c), the red line is the average translation error of median filter on 10 run and the blue line is that of mean filter.

residential, country, urban and highway environments and the running distance of each sequence varies from several hundred meters to thousand meters. The evaluation metrics from [21] calculates the averages of rotation errors and translation errors of every fixed distance segments in each sequence. The evaluation metrics from [22] calculate the absolute translation error(ATE) after a transform of initial result and this metrics can be used to evaluate the scale drift eliminating performance by a preprocessing similarity transformation [18].

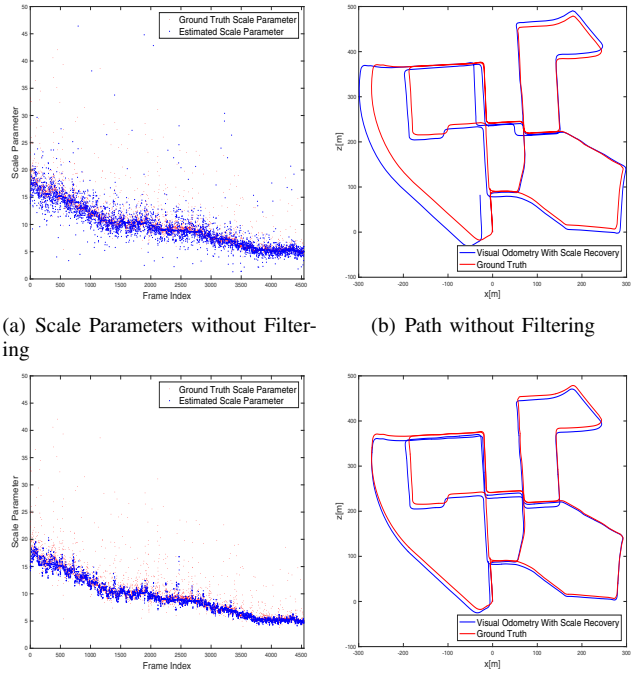
The proposed method is implemented in PYTHON and it will be available on `https://github.com/TimingSpace/MVOScaleRecovery`. The implementation is tested on a laptop PC with Intel Core i5 2.7GHz processor. The initial VO motion results are obtained from ORB-SLAM2 [18].



(a) 00 (b) 02 (c) 03 (d) 04 (e) 05

(f) 06 (g) 07 (h) 08 (i) 09 (j) 10

Fig. 9. This figure shows the result of visual odometry with scale recovery on KITTI benchmark 00 and 02-10. The red line is ground truth path and the blue line the scale recovery path with our method. For dataset sequence 02(Fig. 9(b)), 08(Fig. 9(h)) and 09(Fig. 9(i)), ORB-SLAM2 get lost on frame 2005/4661, frame 3539/4071 and 761/1591 respectively, so the paths are not the complete paths but the paths before ORB-SLAM2 getting lost.



(a) Scale Parameters without Filtering (b) Path without Filtering

(c) Scale Parameters with Median Filtering (d) Path with Median Filtering

Fig. 7. Fig. 7(a) is the scale parameters without filtering and Fig. 7(b) the path with scale recovery by parameters in Fig. 7(a). Fig. 7(c) and Fig. 7(d) are the scale parameters and path with median filtering

In this section, we will first evaluate the performances of different RANSAC parameter and different filtering method in Section IV-A. We will evaluate the scale recovery and scale drift eliminating performance respectively in Section IV-B and Section IV-C. In Section IV-D the over all accuracy of the proposed method will be compared with that of other road-based scale recovery methods.

A. Performance on Different RANSAC Parameters and Different Filtering Parameters

We use RANSAC method to calculate the road geometrical model as described in Section III-B and median filtering method is used to remove the noises. Different RANSAC parameters and filtering parameters will generate different results. We will select suitable parameters according to the experiment results.

Our method is tested with 10 different RANSAC maximum iteration numbers and 17 different filter sizes on KITTI dataset sequence 00. Filter sizes determine how many of the past road models are used to obtain the current road model. We run each case 10 times and evaluate all testing results with the metrics from [21]. The average translation errors are visualized as shown in Fig. 4.

From the figure, we can observe that the translation error is firstly decreasing with the filter size and then increasing with that when the maximum iteration number of RANSAC is fixed. The reason is that when the filter size is small, a bigger filter can remove more outlier, but when the filter size is large, it will make the scale recovery method dull and can not follow up the scale drift. It can be found that 6 is the best filter size. As shown in Fig. 7, the filtering method makes a difference.

Fig. 5 shows the mean value and standard error of each case whose filter size is 6, we can know that with the maximum iteration number increasing, the performance becomes better and stable when the number is over 15. Evidently, our road geometry model based feature points selection method achieves a good result, so the selected feature points are mostly on the road plane. When the number is 15, the mean value and the median value of the translation error

are 1.0207% and 1.0008% respectively. Besides, the standard error of translation errors is 0.0712% which means that our method is very robust and that can also be shown in the box plot (Fig. 5(c)). For all the following experiment, we use 20 as the maximum iteration to leave a margin and make the system more robust.

We also compare the median filtering method with mean filtering method. The RANSAC maximum iteration number is fixed as 20 and we change the filter size of mean filter and median filter. Fig. 8 shows the testing result where we can find that median filter is better.

B. Scale Recovery Performance on KITTI Dataset

We evaluate the performance of the proposed scale recovery method by using ORB-SLAM2 to calculate the initial VO motion. We disable the loop closure module of ORB-SLAM2 because loop closure module can also eliminate the scale drift.

Fig. 6(a), 6(e) and 6(i) show the paths of ORB-SLAM2 on KITTI dataset sequence 00, 05 and 08 with neither loop closure module nor our scale recovery method. The three sequences are chosen because the vehicle runs a long distance(3.7km, 2.2km and 2.8 km respectively) and the longer the vehicle runs, the more distinct the scale drift problem is. In Fig. 6(b), 6(f) and 6(j) the motions obtained by Monocular ORB-SLAM2 are multiplied with three fixed scale parameter 17.0, 17.5 and 25.0 respectively to make the paths in right in the beginning. However, because of the scale drift problem, the paths go further different with the ground paths.

The third column of Fig. 6 are the scale parameters estimated by our proposed method and they are compared with the ground truth scale parameters. The ground truth parameters are calculated by

$$s_g = \frac{\sum_{i=1}^{3} \frac{t_i}{\bar{t}_i} t_i}{\sum_{i=1}^{3} t_i} \tag{11}$$

where $\mathbf{t}$ is ground truth translation in ego-motion. The estimated scale parameter is filtered by median filter as described in Section III-B. By multiplying the estimated scale parameter to corresponding translation in initial ego-motion, we get the ego-motion results in real scale and with little scale drift as shown in Fig. 9(a), 9(e) and 9(h). The quantitative translation errors evaluated by [21] are shown in Fig. 6 as well. The translation errors turn very small after we recover the scale with our proposed method.

C. Scale Drift Eliminating Performance

In order to evaluate the scale drift eliminating performance of our proposed method, we calculate the absolute trajectory error(ATE) using the metric from [22]. We compare the ATE of our method on KITTI dataset with that of ORB-SLAM with loop closure. Loop closure is a global translation error, rotation error and scale drift eliminating method. When comparing the results, as shown in Table I, our performance is pretty good and the absolute error is quite small even without loop closure. For most sequences, our results are comparable if not better than that of ORB-SLAM with loop closure, and the average RMSE is much lower than that of ORB-SLAM. Only on sequence 03, 06 and 07, our result is worse but comparable. For small-scale sequences including sequence 03, 04, 09 and 10, as the scale drift is small, both our results and that of ORB-SLAM are achieving small errors. For large-scale sequences, as our method does not depend on loop closure, we can always obtain small errors. However, ORB-SLAM can only get a good result in sequence 00 and 05 and performs worse in sequence 02 and 08.

TABLE I
SCALE DRIFT ELIMATING EVALUATION AND COMPARISON WITH ORB-SLAM WITH LOOP CLOSURE

Sequence	Dimension [18] (mxm)	Length (m)	ORBwithLC [18] RMSE(m)	ORBnoLC+Ours RMSE(m)
00	564x496	3724	6.68	**5.56**
02	596x946	2085	21.75	**4.36**
03	471x199	561	1.59	**1.36**
04	0.5x294	393	1.79	**2.36**
05	479x426	2206	8.23	**8.03**
06	23x457	1233	14.68	**18.36**
07	191x209	694	3.36	**5.16**
08	808x391	2797	46.58	**5.64**
09	465x568	778	7.62	**2.53**
10	671x177	908	8.68	**2.33**
average			12.10	**5.56**

D. Overall Translation Error Evaluation on KITTI Dataset

We run the proposed scale recovery method on 10 KITTI dataset sequences(00, 02-10). we leave sequence 01 out because it is a highway scene where most VO methods cannot get a good performance. Our method outperforms other scale recovery methods such as [8], [4], [11] and Libviso stereo visual odometry. The results are shown in Table II.

Song et al. [8] was a published scale recovery method with the best performance so far. Comparing our performance on KITTI dataset with that of [8], it can be known that our proposed method has done a better job. [8] can not run on KITTI dataset sequence 07, because the method uses fixed road region, and on sequence 07 that region is occupied by another vehicle. Our method detects the road region online and is able to run sequence 07 and get a translation error 1.73%. For the other 9 sequences, our average translation error is 1.25% and that of [8] is 2.03% and we do better than that of [8] on 8 sequences including sequence 00, 02-05 and 08-10 and only worse in sequence 06.

Lee et al. [11] detects the road with online training classifier which is a very novel method. Comparing the results of [11] with that of us, our performance achieves an improvement, and we are with better results on sequences 00 and 02-09. [11] can run in sequence 01 which is a very difficult highway environment because the method takes the feature of a ground vehicle into consideration.

TABLE II

COMPARISON OF TRANSLATION AND ROTATION ERRORS FOR OUR METHOD VERSUS OTHER VISUAL ODOMETRY METHODS ON THE KITTI BENCHMARK.

Seq	VISO2-M (from [8])		Zhou et al. (from [4])		VISO2-Stereo (from [8])		Lee et al. (from [11])		Song et al. (from [8])		Ours Method	
	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)
00	11.91	0.0209	2.17	0.0039	2.32	0.0109	4.42	0.0150	2.04	0.0048	**1.01**	0.0014
01	-	-	-	-	-	-	6.91	0.0140	-	-	-	-
02	3.33	0.0114	-	-	2.01	0.0074	4.77	0.0168	1.50	0.0035	**0.93**	0.0018
03	10.66	0.0197	2.70	0.0044	2.32	0.0107	8.49	0.0192	3.37	0.0021	**0.52**	0.001
04	7.40	0.0093	-	-	0.99	0.0081	6.21	0.0069	1.43	0.0023	**1.16**	0.0023
05	12.67	0.0328	-	-	1.78	0.0098	5.44	0.0248	2.19	0.0038	**1.45**	0.0014
06	4.74	0.0157	-	-	1.17	0.0072	6.51	0.0222	2.09	0.0081	**2.92**	0.0027
07	-	-	-	-	-	-	6.23	0.0292	-	-	**1.73**	0.0023
08	13.94	0.0203	-	-	2.35	0.0104	8.23	0.0243	2.37	0.0044	**1.18**	0.0017
09	4.04	0.0143	-	-	2.36	0.0094	9.08	0.0286	1.76	0.0047	**1.17**	0.0020
10	25.20	0.0388	2.09	0.0054	1.37	0.0086	9.11	0.0322	2.12	0.0085	**0.93**	0.0029
Avg	14.39	0.0245	2.32	0.045	2.02	0.0095	6.86	0.02119	2.03	0.0045	**1.25**	0.0020

Zhou et al. [4] is a recently proposed scale recovery method, and our method is better than it according to the results.

As shown in Table II, our method is better than not only VISO monocular for every sequence but also VISO stereo method for 8/9 of all the sequences and our average error is also much lower than that of VISO stereo methods.

V. CONCLUSION

In this paper, a novel monocular visual odometry scale recovery method is proposed. We focus on the road geometrical structure to detect the road region and the road geometrical model is calculated and updated online based on the detected road region. As a result, we combine road region detection and road geometrical model calculation into one problem and the two sub-problems can benefit each other. The median filtering method is chosen to make the scale parameter more robust based on the experimental results. Our approach is easy to be realized and it outperforms existing monocular visual odometry scale recovery methods on the KITTI benchmark [21].

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part i - the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 4, 2011.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Simultaneous localization and mapping: Present, future, and the robust-perception age," *CoRR, vol. abs/1606.05830*, 2016.

[3] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1222–1229.

[4] D. Zhou, Y. Dai, and H. Li, "Reliable scale estimation and correction for monocular visual odometry," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, Conference Proceedings, pp. 490–495.

[5] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *Autonomous Robots*, pp. 1–13, 2015.

[6] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with cnns for frame-to-frame ego-motion estimation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, 2016.

[7] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn, and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," 2011.

[8] S. Song, M. Chandraker, and C. Guest, "High accuracy monocular sfm and scale correction for autonomous driving," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1–1, 2015.

[9] S. Choi, J. H. Joung, W. Yu, and J.-I. Cho, "What does ground tell us? monocular visual odometry under planar motion constraint," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. IEEE, 2011, Conference Proceedings, pp. 1480–1485.

[10] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, no. 1, p. 151, 2007.

[11] B. Lee, K. Daniilidis, and D. D. Lee, "Online self-supervised monocular visual odometry for ground vehicles," pp. 5232–5238, 2015.

[12] J. R. Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," in *Applied computational geometry towards geometric engineering*. Springer, 1996, pp. 203–222.

[13] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International journal of computer vision*, vol. 17, no. 1, pp. 43–75, 1996.

[14] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[15] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[16] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," July 2016.

[17] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[18] R. Mur-Artal, J. Montiel, and J. D. Tards, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[19] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, Conference Proceedings, pp. 3354–3361.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.