

# A Simple and Low Cost Method for Traffic Sign Measurement Based on GPS and Epipolar Geometry

## **Guo-hang Shan**

Télécom Bretagne Département Intelligence Artificielle et Sciences Cognitives: IMT Atlantique Bretagne-Pays de la Loire - Campus de Brest Departement logique des usages des sciences sociales et de l'information

## Shuang-cheng Jia

Mogo Auto Intelligence and Telematics Information Technology Co., Ltd

## Qian li ( 569284445@qq.com )

Mogo Auto Intelligece and Telematics Information Technology Co., Ltd https://orcid.org/0000-0003-4037-6152

## **Research Article**

**Keywords:** Deep Learning, Reconstruction algorithms, Simultaneous localization and mapping, Image matching, Feature extraction

Posted Date: September 28th, 2022

### DOI: https://doi.org/10.21203/rs.3.rs-681517/v1

License: (a) This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

## A Simple and Low Cost Method for Traffic Sign Measurement Based on GPS and Epipolar Geometry

GuoHang Shan, ShuangCheng Jia, Qian Li\*

Mogo Auto Intelligence and Telematics Information Technology Co., Ltd, Beijing, China Email: shanguohang@zhidaoauto.com, jiashuangcheng@zhidaoauto.com, liqian@zhidaoauto.com

Abstract—Traffic sign is an important part of traffic signals, as well as an important element of high-precision navigation maps. This article provides a low-cost and high-efficiency technical solution for traffic sign drawing of high-precision navigation maps. Based on a monocular camera and GPS receiver mounted on a vehicle, this paper uses a series of mathematical methods to get accurate geographic coordinates of traffic signs, including selecting two images that are more suitable in space and time interval, exacting and matching features, solving pose, triangulation. The measurement results in this paper are compared with the actual perimeter of the traffic signs, which shows that the measurement method in this paper can meet the relative accuracy requirement of 6% in most cases, the accuracy of our method is the state-of-the-art.

*Index Terms*—Deep Learning; Reconstruction algorithms; Simultaneous localization and mapping; Image matching; Feature extraction

#### I. INTRODUCTION

A traffic sign is an important part of traffic signals, and its main functions are warnings, prohibitions, instructions, and directions. Base on a monocular camera and GPS receiving device, This article uses epipolar constraint to calculate the relative pose of photos and then utilizes GPS position to get the geographic coordinates of traffic signs. This provides a technical route for producing high-precision navigation maps.

Most of the current approaches to building high-precision maps use multiple sensors, including lidar, multi-cameras, IMU, etc. By combining multi-sensors information, multiple targets in the environment can be mapped at the same time [1]. It is simple to build a map using lidar, 3D point clouds of the surrounding environment can be directly obtained, but it is difficult to distinguish objects in the environment. At the same time, lidar is quite expensive [2]. Using multi-cameras to build a map (structure from motion, SFM) can distinguish objects in the environment, but this takes a very long time, usually requires pre-processing, and high precision calibration for cameras [3]. It is also possible to combine multiple sensors, which requires system-level time and space synchronization. However, such algorithms are usually complicated and the equipment is very expensive [3].

This paper proposes a reconstruction method for traffic signs. Compared with existing solutions, this one is concise, low-cost, and quite stable. It needs only one monocular camera and a GPS receiving device, even a smart mobile phone or car driving recorder can satisfy its requirement. We install the driving recorder and fix its camera on the front windshield of the vehicle. Then take photos and record the timestamp of each frame while the vehicle is moving. At the same time, record its GPS position and the corresponding time. With these data, using deep learning to recognize traffic signs, feature points extracting and matching, epipolar geometry, and triangulation, traffic signs can be measured.

The contributions of this paper are summarized as follows:

- Based on the real-world requirements, we identify and address a more challenging but practical problem in traffic sign measurement.
- To deal with the traffic sign measurement problem, we propose a new method based on GPS and epipolar geometry.
- To the best of our knowledge, there is no work doing the same based on the actual engineering needs. Moreover, extensive experiments verified the effectiveness of the proposed method.

The remainder of the paper is organized as follows: Section 2 presents the proposed method. Section 3 conducts the experimental evaluation on the comparison between the proposed method and other methods. Section 4 concludes the paper and identifies the future work.

#### II. METHOD

In this section, we introduce the details of the proposed method.

#### A. Camera Parameter Calibration

The calibration of distortion parameters and intrinsic parameters in this paper use the pinhole camera model, which is the same as OpenCV[6]. The intrinsic parameters have four parameters, namely the focal length in the x and y directions  $(f_x, f_y)$ , and the relative offset between the center of the lens and the center of the imaging unit  $(c_x, c_y)$ . They are placed in a 3x3 matrix, denoted as K:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix}$$
(1)

The distortion parameter describes radial distortion and tangential distortion, with them we can remove the influence of lens distortion, and restore the picture to a pinhole camera model [4] [5].

<sup>\*</sup>Corresponding author: Qian Li (liqian@zhidaoauto.com)



Fig. 1: Technological process of our method

The calibration of the above intrinsic parameters and distortion parameters can use mature tools such as MATBAL camera calibration toolbox or OpenCV calibration function.

The extrinsic parameters calibration uses the data of parallel lane lines and the distance between each lane line [5]. The calibration result includes a rotation matrix between the camera and the vehicle coordinate system, and the installation height of the camera (h). We define the transformed relationship between the camera coordinate system and the vehicle coordinate system as:

$$P_{car} =_{car} R_{cam} \times P_{cam} \tag{2}$$

Where  $P_{car}$  and  $P_{cam}$  are the coordinates position (expressed by a three-dimensional vector) of a certain point in space in the vehicle coordinate system and the camera coordinate system, respectively.

#### B. Extracting and Matching Feature Points

Firstly, extract the feature points from the two selected pictures [6]. The extraction standard is to generate feature points in multiple different areas of each picture, and the number of feature points in the symmetrical direction (left and right/up and down) should be as equal as possible.

At present, there are many algorithms for extracting and describing feature points in images, including brisk [7], surf [8], orb [1], BRIEF, and so on. In this article, we choose the brisk algorithm as the feature point extraction algorithm. Brisk can detect corner points in a picture, then describe it with a 512-bit descriptor. The higher similarity between the two points, the smaller hamming distance of the two descriptors will be. So this algorithm can be used to match the feature points in the two pictures. In this way, we can find feature point A in the first picture, which is corresponding to feature point B in another picture. In order to ensure that the number of feature points in each region is similar, the picture is first symmetrically divided into 16 regions, then feature points are extracted in these regions separately, and the number of feature points in each region is constrained to n. Because the vehicle's movement is at a regular speed, which usually won't turn suddenly in a short time (within 1s), so the content in two images is mostly the same, and the position of the same feature point on them is usually near, we can remove



Fig. 2: From left to right, from top to bottom, the picture represent in turn: a)Deeplab V3 detect result of traffic signs. b)Features extracted with brisk of two consecutive images. Features are marked with a yellow circle. c)Feature matching result of two images. d)Data with vehicles and pedestrians. e)The result of direct matching without removing the moving target. f)Matching result after removing moving target. g)Only keep moving objects matches. h)Actually used feature points exacting strategy.

some mismatched features according to this. Finally, matched feature points are restored to their original positions (pixel coordinates) using distortion parameters. Fig. 2 is the matched result of two images, and they are undistorted.

#### C. Using epipolar to Determine the Relative Photos

By establishing epipolar constraint, the relative motion (rotation matrix R and translation vector t) between two frames can be derived, which is quite common in SFM and monocular vision SLAM. [2] [3] [1] [9]. After matching all the feature points, the relative motion between two frames (denoted as Rand  $t_{unit}$ ) is further solved by epipolar constraint.

Denote the two pictures as  $F_t$  and  $F_q$  respectively and set the pixel coordinate of the feature point on  $F_t$  as  $x_t$ , which matches the feature point on  $F_q$  with coordinate  $x_q$ . According to epipolar constraint,  $x_t$  and  $x_q$  satisfy the following formula:

$$x_t^T \times K^{-T} \times t_{tq} \times R_{tq} \times K^{-1} x_q = 0 \tag{3}$$

where  $t_{tq}$  and  $R_{tq}$  are the relative rotation and translation from  $F_q$  to  $F_t$ , that is, if the coordinates of a space point in the  $F_q$  coordinate system is  $p_q$ , in Ft coordinate system is  $p_t$ respectively, then we get their relationship:

$$p_t = R_{tq} \times p_q + t_{tq} \tag{4}$$

Denote the Essential Matrix as E, which is calculated by  $E = t_{tq} \times R_{tq}$ , the above formula can be written as:

$$x_t^T \times K^{-T} \times E_{tq} \times K^{-1} x_q = 0 \tag{5}$$

where  $x_t$  and  $x_q$  are the homogeneous pixel coordinates of the feature point that have been successfully matched, Kis the intrinsic parameter matrix which is obtained through calibration and only  $E_{qt}$  is unknown. Through the eight-point method [10] and least-squares optimization, this function can be solved. By SVD decomposition and the geometric model of the camera, we can finally get  $R_{tq}$  and  $t_{tq}$ .

Record the  $t_{tq}$  at this time as  $t_{tq\_unit}$ . This is because we can not get the true length by the monocular system, and the solution  $t_{tq\_unit}$  is actually a unit vector that can only indicate a direction, which means you can multiply  $t_{tq\_unit}$  by an arbitrary constant number. So it's impossible to describe in real length units, such as meters and feet. GPS information will solve this.

#### D. Using GPS Information

1) Calculate Real Distance: Assuming that within the period of  $ts_{first}$   $ts_{last}$ , the GPS coordinates of the vehicle are recorded at a fixed frequency(N Hz), then for each time point  $ts_aim \in [ts_{first}, ts_{last}]$ , two GPS coordinates  $G_{prev}$  and  $G_{next}$  can be found, whose time is  $ts_{prev}$  and  $ts_{next}$ ,

which are the closest previous and next GPS coordinates to  $ts_{aim}$ , respectively.

Assuming that the vehicle is at a constant speed during this period, we can get the GPS location of  $ts_{aim}$ (linear interpolation):

$$G_{aim} = G_{prev} + E \tag{6}$$

$$E = (ts_{next} - ts_{prev}) \times (ts_{aim} - ts_{prev}) \times (G_{next} - G_{prev})$$
(7)

Denote the shooting time of  $F_q$  as  $t_q$  and  $F_t$  as  $t_t$ .  $G_q$  and  $G_t$  can be obtained separately. Then the distance between  $F_q$  and  $F_t$  can be obtained as:

$$d = GPS_{Distance}(G_q, G_t) \tag{8}$$

where  $GPS_{Distance}$  is the function for calculating two GPS points, and we will not describe it further. Then the actual relative translation of the two photos can be calculated by the following formula:

$$t_{tq\_real} = d * t_{tq\_unit} \tag{9}$$

2) Calculate Heading Angle: We only consider the situation that the vehicle is traveling along a straight line, and the heading of the vehicle is consistent with its traveling direction. According to this feature, we can use GPS information to calculate the direction of the vehicle (usually called heading angle).

For the previously mentioned  $t_{aim}$ , we can simply use the direction of the vector  $\langle G_{prev}, G_{next} \rangle$  as the heading at this point. Define the heading angle toward north as 0°, east as 90°, south as 180°, and west as 270°. Denote the heading angle at  $t_{aim}$  time as  $A_{aim}$ .

In fact, this is a little rough when GPS frequency is low, and the error becomes large when the trajectory of the vehicle is slightly curved (the vehicle is turning). Usually, we will do cubic polynomial interpolation. That is, collecting some nearby coordinates before and after  $t_{aim}$ , then do cubic polynomial fitting to them, and then calculate the heading angle at  $t_{aim}$  according to the fitting results.

*3)* Geographic Coordinate System Location: By the definition of extrinsic parameter calibration [5], we set the origin of the vehicle coordinate system the same as the origin of the camera coordinate system, the front of the vehicle is z-axis, the vertical upward is the y-axis, and the right side is the x-axis.

Generally, in the geographic coordinate system, the x-axis point to the east, the y-axis point to the north, and the z-axis point to the sky. According to the definition of the coordinate system, the rotation relationship between the point P in the vehicle coordinate system and the geodetic coordinate system is:

$$P_{world} = \begin{bmatrix} \cos a & 0 & \sin a \\ -\sin a & 0 & \cos a \\ 0 & 1 & 0 \end{bmatrix} P_{car}$$
(10)

where  $\alpha$  is the heading angle calculated above.  $P_{car}$  is the coordinate position of point P in the vehicle coordinate system. It expresses the positional relationship between point P and the vehicle in this way: Point P is located at  $P_{car}$ .x meters to the right of the vehicle,  $P_{car}$ .y meters above the vehicle, and  $P_{car}$ .z meters ahead of the vehicle.  $P_{world}$  is the relative relationship between point P and the vehicle in the north-east-sky coordinate system, which can be described as: Point P is located at  $P_{world}$ .x meters to the east of the vehicle,  $P_{world}$ .y meters to the north of the vehicle, and  $P_{world}$ .z meters above the vehicle.

It should be noted that there is an assumption, that the z-axis of the vehicle coordinate system is parallel to the z-axis of the world coordinate system. There must be an error in this assumption, but because the angle between them is usually very small (within 5 degrees), and the distance between the target space point and the car is not very far (usually around 20 meters), the position error is within  $1.7m (20 \times \sin 5)$ .

4) Triangulation to Solve the Point Coordinates in Camera Coordinate System [11]: In trigonometry and geometry, triangulation is the process of determining the location of a point by forming triangles with two known points, instead of directly measuring the distance to a specific location (trilateral measurement method). When baseline length and two observation angles are known, the observation target point can be calibrated as the third point of a triangle.

According to the imaging model of the camera, the pixel coordinates  $x_q$  and  $x_t$  (homogeneous coordinates) of any spatial point P on  $F_q$  and  $F_t$  satisfy the following formula:

$$s_t x_t = K p_t \tag{11}$$

$$s_q x_q = K p_q = K (R_{qt} \times p_t + t_{qt}) \tag{12}$$

where  $p_t$  and  $p_q$  are the coordinates of P in  $F_t$  and  $F_q$  coordinate systems. respectively,  $s_t$  and  $s_q$  are the depths of P in  $F_q$  and  $F_t$ , respectively, which is the z coordinates of  $p_t$  and  $p_q$ .

In these two formulas, only  $p_t$  and  $p_q$  is unknown  $(s_t$  and  $s_q$  can be calculated by  $p_t$ , which can be regarded as a known variable), so we can find  $p_t(p_q)$  by solving the equation. But due to noise and other inaccurate factors, such as the inaccuracy of  $R_{tq}$  and  $t_{tq}$ , the inaccuracy of the matching same feature points taken by  $x_t$  and  $x_q$ , and so on, we choose to determine  $p_t$  (and  $p_q$ ) through the least square solution.

In the procedure of measuring traffic signs, the corresponding image points of the traffic sign on  $F_q$  and Ft are identified manually. For polygonal traffic signs, we choose their vertices. On one hand, all the vertices can accurately describe the position, size, shape, and other essential factors of a traffic sign. On the other hand, not like the center point, vertices can be clearly marked in different photos. The matching error will be greatly reduced.

#### III. EVALUATION

In this section, we introduce the service dataset, implementary details, evaluation metrics, and the evaluation results of



Fig. 3: From left to right, from top to bottom, the picture represent in turn: a)Feature exacting result of different divide method. b)Test traffic signs appearance. c)number of Uniform region with different feature exact method. d)Relative error distribution (inside less than 10%).

our method, which includes the comparison results with other traditional methods.

#### A. Data Description

The data includes the three parts:

1) *Picture:* Pictures were taken during driving and their shooting time.

2) Camera: Camera calibration data, which includes intrinsic parameters, extrinsic parameters, and distortion parameters. Distortion parameters are used to restore the distortion in pictures. Intrinsic parameters are used to establish the camera shooting model, and it contains the focal length of the lens, the offset between the lens center and the image center, the size of the imaging unit, etc. [4]. The extrinsic parameter here is the relative spatial relationship between the camera and the vehicle. Since these two devices are always relatively stationary, we just need relative rotation between them and the installation height of the camera [5].

*3) GPS:* GPS position of the driving recorder during driving and the corresponding accurate time.

#### B. Implementation details

In this paper, the image size we use is  $1920 \times 1080$  and extract about 2000 feature points from the whole image. Then observe the distribution of the extracted feature points on the image. In order to describe whether its distribution is uniform, the image is divided into 25 (5x5) areas, and the number of

feature points falling in each area is counted. If the distribution is absolutely uniform, each area should contain 80 feature points. Considering that some regions are indeed complex, and some are monotonous, the number should fluctuate. Therefore, if there are 40-120 feature points in a certain region, we classify this region to be uniform and count the number of uniform regions.

First of all, we divide the picture with m\*n regions, and each region's area is equal. Then, try to exact N=2500/(m\*n) feature from each region. If there are more potential features, just choose those whose scores are higher. If the number of features in this region is less than N, finish the exacting of this region with the current result. Because our images are taken from the vehicle moving on the road, it is quite normal to meet not a uniform distribution, because sky and road are usually hard to exact or match features however, buildings are full of pattern. In this chapter, we try to find the best way to divide pictures, which can make feature distribution more uniform.

Choose the case of n \* n for comparison. With each picture, we tested different exact strategies. Fig. 5 is a statistical distribution curve based on the evaluation method we set. Since the evaluation method we chose is 5x5, correspondingly, 5x5 division has the best effect. Remove this special case, it can be seen that the distribution of feature points is more uniform as the number of segmentation increases. In general, the more partitions, the more even the distribution of feature points. But, among them, 3x3 and 4x4 are abnormally inverted, which should be related to our test sample: there is a large area of the car at the bottom of the picture, a large area of sky and a digital watermark at the top, and you can see that the 3x3 data is a little unstable, the 7-th and 10-th images are quite bad than others, that is to say the difference between different samples is large. In this respect, 4x4 is more stable, and the uniformity level of each sample is not much different.

#### C. Evaluation Metrics

For fair experiment we choose the root-mean-square error(RMSE) to evaluate our method. The root-mean-square error (RMSE) is commonly used to measure the accuracy of predictive scoring systems, where a smaller RMSE value indicates higher accuracy. More precisely, the RMSE represents the sum of the squares of the deviation between the observation value and the true (or best) value. The best value then refers to the reference trajectory results, and the observation value refers to the predicted trajectory during RTK outages.

#### D. Experimental Results and Discussion

1) Results of Extracting Feature Points : The following figure shows the extraction of feature points in different ways of segmenting regions. It can be clearly seen that in the case of 1x1, the feature points will be more concentrated in certain areas, and relatively few in other areas. However, if the image is divided into different regions to extract feature points, the distribution situation is greatly improved. This feature will facilitate our subsequent feature point matching and pose calculation. We list 1x1, 3x3, and 5x5 conditions in Fig. 3.a.

TABLE I: Different Matching Calculation Result

Item		Side Length			Perimeter	Relative Error
Actual side length	1.5	3	1.5	3	9	0
Full image matching	1.42333	2.67799	1.36256	2.82706	8.29094	0.078784
Remove moving objects	1.6742	2.85601	1.66523	3.01844	9.21388	0.023764
Only match moving objects	1.22058	2.48729	1.15694	2.63275	7.49756	0.166938

Based on the above experimental results, considering the running time and complexity of the program, as well as the uniformity of feature points distribution, we use the 4x4 division to exact feature points. At the same time, because the upper and lower parts of the picture contain a lot of invalid information (time watermark and car body), the surrounding parts of the image are severely distorted, we choose to extract the feature points inside the center area. As Fig.2.h shows.

2) Results of Removing Moving Objects: As a result of the objects in photos are relatively static, and moving objects will lead to great error. Fig 8 are two pictures that contain a lot of vehicles and pedestrians, along with them is a traffic sign that we want to measure. As show in Fig. 2.d and Fig. 2.e.

The actual size of the traffic sign in this picture is 1.5mx3m. If use these two pictures directly, then these moving objects

will also be matched during feature point matching. The following process, calculating pose, will be affected by this. After calculation, we got the size of the traffic sign as shown in the third row of Table I. Afterward, we add a mask to the vehicles in it. The matching result is shown in Fig. 2.f. The traffic sign size obtained is shown in the fourth row of Table I. It can be seen that the calculation accuracy has been significantly improved.

#### TABLE II: Overall Error Statistics

Total	Mean Error	Maximum Error	Maximum Error	More than 20%	10%-20%	6%-10%	Less than 6%
227	0.050804	58.58%	0.02%	10	16	12	189

TABLE III: Statistics of Different Size Traffic Signs

Ground truth (m)	12	14	9	8.4	5.2	15.2
Sample number	89	33	17	44	5	39
Average absolute error(m)	0.57358	0.166664	0.422942	0.805462273	0.873254	0.4006
Average relative error	0.047798	0.011905	0.046994	0.095888366	0.167933	0.026355

It should be noted that although there was the influence of vehicles in the first test, the calculation error is just 7.9%, this is because there are a lot of "good" matches in the whole picture, at the same time, the movement of moving objects is not violent, so "good" matches make those "bad" matches' influence reduced. If the number of "good" matches in the picture decreases, the situation will be reversed. In order to illustrate this situation, we remove most of the good matches and keep most of the bad matches, as shown in Fig.2.g. After calculation, the results are shown in row fifth in the Table I and the error is greatly increased.

3) Result of Overall Accuracy: In order to evaluate the accuracy of the method in this article, we select 6 traffic signs of different sizes for measurement. Their specific specifications are shown in Fig. 3.b. According to the characteristics of traffic signs, as long as the lengths of the sides are right, the shape and size are correct. Therefore, we choose the perimeter to evaluate the measurement error. Table II is the measurement error analysis. We selected 227 pairs of images, and this dataset contains all the six kinds of traffic signs above. From this table, we can see that most of the data's (about 88.5%) errors are ;10%. We list all those errors inside 10%, and their distribution is shown in Fig. 3.b and Fig. 3.d.

For traffic signs of different sizes, we count their error separately, and the result is shown in Table III. It can be seen that with the measurement method proposed in this article, more than 83.26 samples' error was less than 6%, and only 4% samples' error was greater than 20%. We sort out the largest error data in this test and then analyze the reasons for their larger errors. For larger traffic signs, the main error is wrong

matching, and the main reason for wrong matching is that the distance between the shooting positions of the two pictures is too far, and the road surface and road measurement patterns are relatively monotonous, which will cause the feature matching algorithm to easily match the wrong position, resulting in a large relative pose error.

#### IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed an easy, low-cost method to measure traffic signs, all we need is a camera and GPS receiving device. At the same time, we can get its geographical coordinates, so we can put it on an electronic map. In our test, 88.5% of sample data's relative error is less than 10%, and only 4% of sample data's relative error is more than 20%. To make the result more accurate, we remove those moving objects inside the picture so that most matching features are static, select two pictures more close in spatial (also in time), so it improves the accuracy of matching. To sum up, this article proposes an innovative solution to the geographic coordinates of traffic signs, which provides a low-cost and high-efficiency solution for high-precision map production and automatic driving. We believe that our work shall be valuable to the related experts working in the fields by providing with promising direction for future research.

#### References

- R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, 2015.
- [2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 1271–1278.
- [3] H.-S. Kim, K.-B. Seo, M. Kang, G. S. Ryu, and M. Hong, "Design and implementation of luo-kuan recognition application," *Journal of Internet Computing and Services*, vol. 19, no. 1, pp. 97–103, 2018.
- [4] T. X. Gao, "Fourteen lectures on visual slam: From theory to practice," *Publishing House of Electronics Industry.*
- "Opencv," https://docs.opencv.org/2.4/modules/calib3d/doc/camera\_calibr ation\_and\_3d\_reconstruction.html?highlightradial%20distortion.
- [6] "Feature detection," https://en.wikipedia.org/wiki/Feature\_detection\_(com puter\_vision).
- [7] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in 2011 International conference on computer vision. Ieee, 2011, pp. 2548–2555.
- [8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [9] G. P. Huang, L. I. Xiao-Yong, and G. Q. Qin, "Test-range calibration of digital camera," *Journal of Institute of Surveying and Mapping*, 2005.
- [10] Q. L. Guo-hang Shan, Shuang-cheng Jia, "Carcorder camera calibration method of external parameters based on lane line," *ICISCE 2020 7th International Conference on Information Science and Control Engineering*, 2020.
- [11] "Triangulation," https://en.wikipedia.org/wiki/Triangulation.