Improved ORB-SLAM2 Algorithm with Image deblurring

Liu, Peiqi

(出版者 / Publisher)
法政大学大学院理工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学研究科編
(巻 / Volume)
63
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2022-03-24
(URL)
https://doi.org/10.15002/00025366

Improved ORB-SLAM2 Algorithm with Image deblurring

Liu Peiqi

Applied Informatics Major, Graduate School of Science and Engineering,

Hosei University,

Supervisor: Professor Jinjia Zhou

***Abstract*— ORB-SLAM2 is a feature-based sparse visual simultaneous localization and mapping (SLAM) system. After obtaining images and related data from sensors and cameras, it can extract useful information from them, and perform real-time positioning and map construction. This technology is now widely used in the fields of unmanned driving, smart home and factory operations, and it also provides a relatively stable foundation for the creation and development of many new SLAM systems. However, the current ORB-SLAM2 system can only work stably and normally when the camera is moving steadily and slowly. When the camera moves too fast or violently bumps, the image taken by the camera will inevitably be blurred, which will easily cause the "Track Lost" of the ORB-SLAM2 system. In order to solve this problem, this thesis combines the ORB-SLAM2 system with a deblurring algorithm based on deep learning, and uses the SLAM-oriented data set to optimize the deep learning network. This thesis also adds a blur detection module to ORB-SLAM2, which enables it to automatically distinguish blurry images. The experimental results demonstrated that this work detects and deblurs the blurred image, successfully avoiding part of the "Track Lost", and improving the accuracy of ORB-SLAM2 mapping.**

Keywords: Visual SLAM; ORB-SLAM2; blur detection; deblurring; DeblurGAN-v2.

I. INTRODUCTION

With the rapid development of science and technology, high-tech technologies such as drones, autonomous driving, and intelligent robots have gradually attracted more and more attention in the past two decades. At the same time, however, these technologies are facing a problem that needs to be solved urgently. In order to ensure that the robot can accurately and quickly go to the destination designated by the human, we usually need to prepare an extremely accurate and reliable map for it, so that it clearly knows its own location and the location of the destination during the movement. However, in practical applications, the environment in which the robot is located is usually "unknown" in many cases. Therefore, we need the robot to locate its own location based on the information returned by the sensor without any prior knowledge, analyze the data by itself, and finally complete the mapping of the unknown environment. Real-time positioning and map construction technology has thus become one of the hot topics of researchers.

The basic concept of SLAM technology is that we equip a robot or robot car with some specific sensors, and then put it into an unknown environment. When the robot moves in that unknown environment, although it cannot directly obtain its own position coordinates, it can obtain some indirect physical quantities from the sensor, such as the angular velocity and acceleration of the movement, and the image of the surrounding environment. We hope that after obtaining these data, the robot can analyze and calculate its own position independently, and record it. Finally, by repeatedly observing the unknown environment, recording relevant data, analyzing and calculating, a complete map of the environment will be constructed.

When the sensor is set as a camera, we call this system visual SLAM (V-SLAM). The visual SLAM camera will shoot the surrounding environment at a certain rate and obtain the image stream. ORB-SLAM only supports monocular cameras, while ORB-SLAM2 supports both binocular cameras and RGB-D cameras on this basis.

According to the different ways of processing input data, we divide V-SLAM into feature-based SLAM and direct SLAM. Feature-based SLAM will perform feature extraction on the input image, and then only process the feature data.

The ORB-SLAM used in this thesis is a sparse SLAM system based on feature extraction. The feature point extraction method it uses is the oriented FAST and rotated BRIEF (ORB) algorithm. Because ORB-SLAM does not process all the elements in the input image, but only extracts some of the data with obvious characteristics for subsequent operations, which results in it being more robust than direct SLAM. However, its shortcomings are also very obvious: because the construction of the map completely depends on the extraction and matching of feature points, when errors occur in these two steps, it is easy to make the ORB-SLAM system not work properly.

So far, there is currently no suitable solution to the problem of "the rapid movement of the robot or the violent turbulence caused motion blur in the captured pictures, so that sufficient feature points cannot be extracted". Therefore, this work has carried out research and discussion on this issue. By combining deep learning technology with ORB-SLAM2, the accuracy of mapping has been improved.

II. METHODS

A. The overall framework

The overall framework of this work is shown in Figure 1. After getting the image from the monocular camera, before ORB-SLAM2 performs image preprocessing, we added a blur detection module and a deblurring module. In the blur detection module, we will detect the input image to determine whether it is "blur". If it is detected that the image contains motion blur, the image will be sent to the deblurring module for deblurring, and then the clear image after deblurring will be sent to the tracking module of ORB-SLAM2. If the detected image is clearer, the image will be directly sent to the tracking module of ORB-SLAM2 to save calculations and reduce running time.

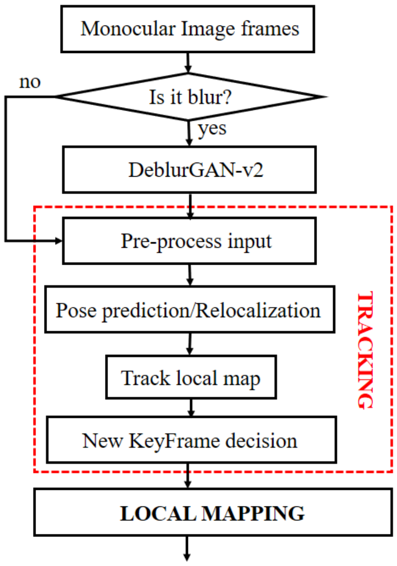


Fig 1. The overall flow framework of our work

B. The blur detection module

In order to ensure the real-time performance of the ORB-SLAM2 system, we have chosen a simpler method to detect the image blur. In this method, the blurriness of the image is only represented by a floating-point value. After obtaining the floating-point value, we will compare it with a set threshold to determine whether the image is blurry. The overall process framework of the blur detection part is shown in Figure 2.

As can be seen from the figure, the blur detection part is roughly divided into four steps: First, we convert the obtained image into a grayscale image, and then we convolve it with the Laplacian operator, and then calculate its variance. Finally, we compare the variance obtained with the threshold set at the beginning. The threshold represents the minimum "information" required for an image to be judged as "clear". If the variance is greater than the set threshold, it means that the blur degree of the image is small, and it will be determined as "clear". If the variance is less than the set threshold, it means that the image is blurred and the image will be determined to be "blurred".

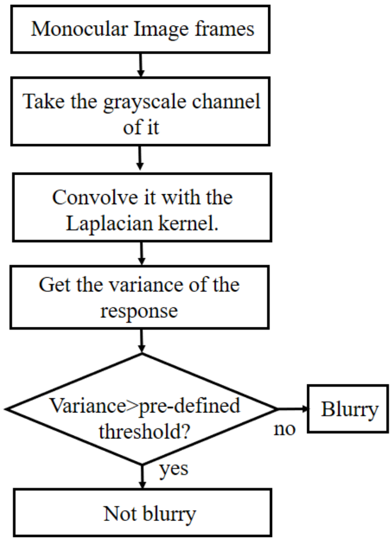


Fig 2. The overall framework of the blur detection model

The blur detection method chosen in this work expresses the "blur degree" of the image with a floating-point value, which is very convenient and simple, but the selection of the threshold is very important. When the threshold is set too low, if the image is not clear, it will be incorrectly marked as blurred. If the threshold is too high, an image that is actually blurred will not be marked as blurred.

The selected threshold is closely related to the "data" or "data set" used in the blur detection experiment, and there is a clear gap between the optimal thresholds in different situations. Therefore, according to the data set used in the experiment of this thesis, it is very important to select a targeted preset threshold that is more suitable for the ORB-SLAM2 system.

In order to select a more suitable blur detection threshold for the ORB-SLAM2 system, we used some data from EuRoC MAV Dataset to construct a new test data set V-test. This data set contains 2182 gray-scale images, of which about 300 blur images cannot be extracted by ORB-SLAM2 and cause the "Track Lost" phenomenon.

We set the progressive value to 0.1, and tested the number of blurred images from 1.0 to 2.0 in turn. In this case, the number of blurred images measured by the blur detection program is shown in Table 1.

Table 1. Threshold and corresponding blurred images

Threshold	Blur img	Threshold	Blur img
1.0	146	1.6	245
1.1	179	1.7	**295**
1.2	191	1.8	443
1.3	199	1.9	752
1.4	199	2.0	1155
1.5	223		

It can be seen from Table 1 that when the threshold is set to 1.7, the number of blurred images detected by the program is closest to 300, and when the threshold is greater than 1.7, the number of blurred images detected suddenly increases sharply. And after the comparison, it is found that the image detected by the blur detection program and the blur image in the original data set basically coincide. This shows that when the threshold is set to 1.7, it is most suitable for the ORB-SLAM2 system.

C. Image deblurring module

To ensure the real-time performance of the ORB-SLAM2 system, we intend to find a lightweight, fast, and low-computing deblurring algorithm. DeblurGAN-v2 is an end-to-end generative adversarial network (GAN) which oriented to deblurring blurred images caused by fast motion. So this work finally decided to use DeblurGAN-v2 for deblurring.

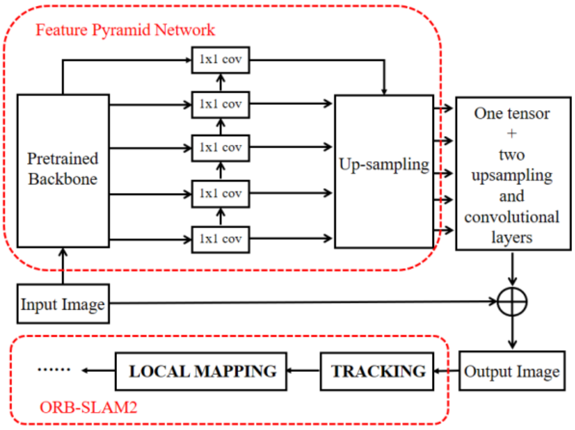


Fig 3. The overall framework of Image deblurring module

The overall framework of the image deblurring module which using DeblurGAN-v2 is shown in Figure 3. The main body of the structure is a feature pyramid network (FPN), which supports a variety of backbone options, and here they used Inception-ResNet-v. The FPN designed by O. Kupyn et al. contains five-scale feature output, which is up-sampled to a quarter of the original input size and stitched as a whole (it contains multi-scale information). After that, two up-sampling modules and convolutional layers were added to restore the original image size and reduce artifacts.

When training DeblurGAN-v2, O. Kupyn et al. formed their training data set by selecting every second frame from the GoPro and DVD data sets, and selecting each tenth frame from the NFS data set. Correspondingly, we also select images from the EuRoC MAV dataset to form a training data set that is more suitable for the SLAM system. The DeblurGAN-v2 network trained with such a data set will be more suitable for SLAM system deblurring, thereby improving the accuracy of building maps.

III. RESULTS

The evaluation work of this experiment was conducted on Ubuntu 16.04 and ROS Kinetic with Intel® Core™ i7-9700 CPU @ 3.00GHz with 8 cores and RAM 32GB. The graphics is llvmpipe (LLVM 6.0, 256 bits), and the operating system type is 64 bits.

A. Implementation of real-time ORB-SLAM2 system

We used the RealSense D435i RGB-D camera to make ORB-SLAM2 run in real time in a real environment. Its running state is shown in Figure 4. As shown in the figure, the environment taken by the camera is an ordinary room. ORB-SLAM2 successfully detected the feature points on the image and constructed the map in real time.

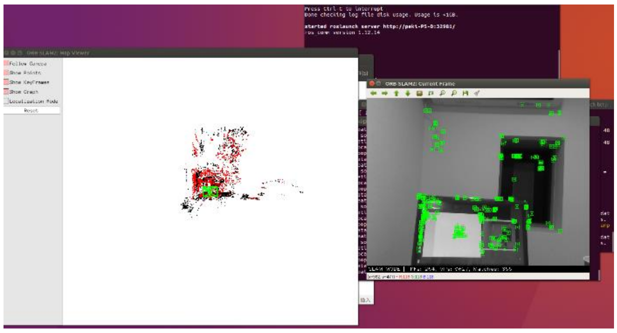


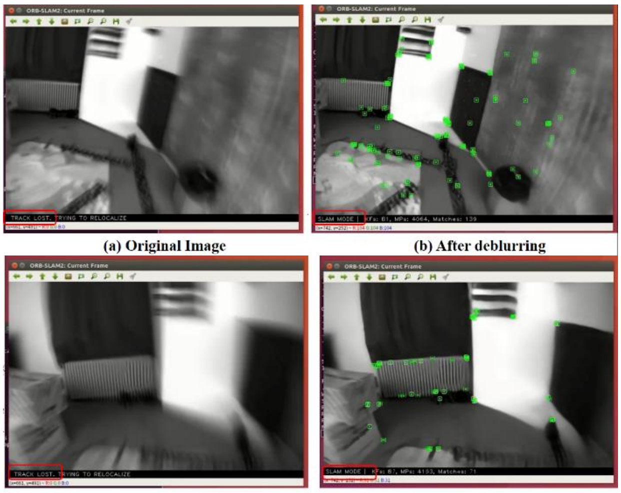
Fig 4. ORB-SLAM2 run in real time in a real environment

We also used 4WD robot substrate and RPLIDAR A1 360° laser scanning radar to build a highly integrated ROS robot to cooperate with the experiment, as shown in Figure 5. After loading the D435i RGB-D camera and laptop on the robot, we can remotely control the car to move in the real environment. At the same time, ORB-SLAM2 will process the image sent by the RGB-D camera and construct map in real time.

Fig 5. The highly integrated ROS robot

B. Avoidance of the "Track Lost" phenomenon

We used the original ORB-SLAM2 and the deblurring ORB-SLAM2 proposed in this work to run the same data set for comparison experiments. Figure 6 respectively shows two sets of contrast images of ORB-SLAM2 extracted feature points before and after deblurring. Among them, the image (a) is the running image of the original ORB-SLAM2. We can see that because the camera moved too fast and produced strong motion blur, ORB-SLAM2 did not extract any feature points from the image. This directly leads to the "Track Lost" phenomenon of the system, and the information bar in the lower left corner also prompts "TRACK LOST, TRYING TO RELOCALIZE". When running the same data set with the deblurring ORB-SLAM2 proposed in this work, a part of the feature points can be successfully extracted from the image, and the track lost phenomenon of ORB-SLAM2 can be successfully avoided, as shown in image (b).



(a) Original Image (b) After deblurring

(a) Original Image (b) After deblurring

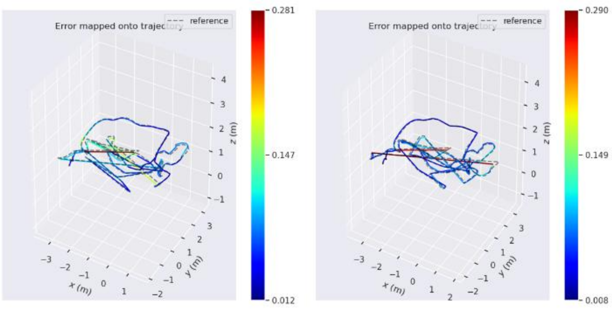
Fig 6. The highly integrated ROS robot

C. Error comparison of constructing map

We use APE(absolute pose error) to evaluate the error of constructing the map. It can be seen from Table 2 that after deblurring, the average value of the error of ORB-SLAM2's map construction on the V203 data set is reduced. In addition, the variance and root mean square error of APE also decreased. This shows that after deblurring, the accuracy of ORB-SLAM2's map construction is improved.

Table 2. Comparison of APE

	ORB-SLAM2	This work
max	0.281186	0.289539
mean	0.083180	**0.069858**
median	0.077765	0.068798
min	0.012458	0.008460
rmse	0.091723	**0.077995**
sse	1.447070	1.076718
std	0.038656	0.034685



(a) Original Image (b) After deblurring

Fig 7. Error mapped onto trajectory before and after image deblurring

The "Error mapped onto trajectory" in both cases is shown in Figure 7. We can see from the figure that the dotted line is ground truth, and the colored line is the map path drawn by ORB-SLAM2. Although the error of the deblurring ORB-SLAM2 designed in this work is greater than the original ORB-SLAM2 in some straight paths, the overall composition error is smaller than the original ORB-SLAM2.

IV. DISCUSSION

This thesis first introduces the background knowledge of SLAM technology and ORB-SLAM2 system. The basic concept of SLAM technology is to put a robot equipped with sensors into an unknown environment. Record the physical quantity it obtains from the sensor during the movement, and then analyze it autonomously, calculate its position, and record it. Finally, through repeated observations, it will construct a map of the unknown environment.

However, in actual experiments, the robot car often encounters rapid turns or shaking during its travel. This will cause the image captured by the camera to contain strong "motion blur". This work mainly improves the ORB-SLAM2 algorithm to solve this problem. Although we have achieved some results, this work still has a lot of space for optimization. Moreover, we have not yet achieved real-time deblurring of images, but are only experimenting with data sets. In subsequent research, this will also become an important direction for researchers to carry out follow-up research.

V. CONCLUSION

ORB-SLAM2 is a sparse visual SLAM system based on feature extraction that supports monocular, binocular and RGB-D cameras. It uses ORB feature extraction method for feature point extraction. Although this algorithm is less time-consuming and has relatively good scaling and rotation invariance, its shortcomings are also obvious: it cannot extract enough feature points from images with inconspicuous features. Therefore, whether the image captured by the camera has motion blur or not has a great influence on ORB feature extraction.

In order to solve this problem, this work improves the original ORB-SLAM2 algorithm, and adds a blur detection module and a deblurring module to its original framework. The blur detection module uses the Laplacian operator to calculate the degree of blur of the image, so as to judge whether the image is blurry. If the image is blurred, it will be sent to the deblurring module. If the image is clear, it will be sent directly to the ORB-SLAM2 system for analysis.

The results of the experiment show that this work successfully avoided some of the "TRACK LOST" of ORB-SLAM2 and improved the accuracy of its map construction.

ACKNOWLEDGEMENT

First of all, I would like to extend my most sincere thanks to my supervisor, Professor Zhou Jinjia. She gave me meticulous help and patience at every stage of the completion of this thesis. When I am confused or frustrated, she always encourages me with the gentlest words, and I am really grateful. At the same time, I would also like to express my gratitude to Professor Kazuo Yana for his care and help after I enrolled in the university. Then I would also like to thank Mr. Xiao Peng for his suggestions on SLAM-related content.

Secondly, my sincere appreciation also goes to all the staff of Hosei University for their help in my life. It is precisely because you have solved many problems I encountered in my daily life that I can focus on my research without any worries. In addition, I also owe my sincere gratitude all the members of the laboratory for their company. Among them, I especially want to thank Minh Man Ho for his guidance in python programming, and Nguyen Tan Ho for his help in deblurring.

Finally, I would also like to appreciate my most beloved parents, thank you for giving me this opportunity to study abroad. Thank you mom for always encouraging me when I encounter setbacks, and thank you dad for supporting the whole family silently. I love you.

REFERENCES

[1] Guillaume Bresson, Zayed Alsayed, Li Yu, Sébastien Glaser, “Simultaneous Localization And Mapping: A Survey of Current Trends in Autonomous Driving,” *IEEE Transactions on Intelligent Vehicles, Institute of Electrical and Electronics Engineers, 2017,* XX, pp.1. 10.1109/TIV.2017.2749181. hal-01615897

[2] M-R.Raul, J. M. M. Montiel and J.D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Transactions on Robotics,* vol. 31, no. 5, pp. 1147-1163, 2015.

[3] M-R.Raul, J.D. Tardós, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” *IEEE Transactions on Robotics,* vol. 33, no. 5, pp. 1255-1262, 2017.

[4] Qin J, Li M, Liao X, Zhong J, “Accumulative Errors Optimization for Visual Odometry of ORB-SLAM2 Based on RGB-D Cameras,” *ISPRS International Journal of Geo-Information,* 2019; 8(12):581.

[5] L. Hao, H. Li, Q. Zhang, X. Hu and J. Cheng, “LMVI-SLAM: Robust Low-Light Monocular Visual-Inertial Simultaneous Localization and Mapping,” *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO),* 2019, pp. 272-277, doi: 10.1109/ROBIO49542.2019.8961635.

[6] B. Bescos, J. M. Fácil, J. Civera and J. Neira, “DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes,” *in IEEE Robotics and Automation Letters,* vol. 3, no. 4, pp. 4076-4083, Oct. 2018, doi: 10.1109/LRA.2018.2860039.

[7] D. Galvez-López and J. D. Tardos, “Bags of Binary Words for Fast Place Recognition in Image Sequences,” *in IEEE Transactions on Robotics,* vol. 28, no. 5, pp. 1188-1197, Oct. 2012, doi: 10.1109/TRO.2012.2197158.

[8] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada, “OpenVSLAM: A Versatile Visual SLAM Framework,” *In Proceedings of the 27th ACM International Conference on Multimedia (MM '19). Association for Computing Machinery,* 2019, New York, NY, USA, 2292–2295. DOI:https://doi.org/10.1145/3343031.3350539

[9] Qin J, Li M, Liao X, Zhong J, “Accumulative Errors Optimization for Visual Odometry of ORB-SLAM2 Based on RGB-D Cameras,” *ISPRS International Journal of Geo-Information,* 2019; 8(12):581.

[10] J. Tang, J. Folkesson, and P. Jensfelt, “Geometric correspondence network for camera motion estimation,” *IEEE Robotics and Automation Letters,* vol. 2, no. 2, pp. 1010–1017, 2018.

[11] J. Tang, L. Ericson, J. Folkesson and P. Jensfelt, “GCNv2: Efficient Correspondence Prediction for Real-Time SLAM,” *in IEEE Robotics and Automation Letters,* vol. 4, no. 4, pp. 3505-3512, Oct. 2019, doi: 10.1109/LRA.2019.2927954.

[12] O. Kupyn, T. Martyniuk, J. Wu, Z. Wang, “DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better,” *Proceedings of the IEEE/CVF International Conference on Computer Vision 2019,* pp. 8878-8887.

[13] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, “DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018,* pp. 8183-8192.

[14] R. Mur-Artal and J. D. Tard´os, “Visual-inertial monocular SLAM with map reuse,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr.2017.

[15] H. Liu, M. Chen, G. Zhang, H. Bao and Y. Bao, "ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2018, pp. 1974-1982, doi: 10.1109/CVPR.2018.00211.

[16] Slinko, Igor et al. “Training Deep SLAM on Single Frames.” ArXiv abs/1912.05405 (2019): n. pag.

[17] H. Qian and P. Ding, "An Improved ORB-SLAM2 in Dynamic Scene with Instance Segmentation," *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS),* 2019, pp. 185-191, doi: 10.1109/REDUAS47371.2019.8999687.

[18] Y. Wang and X. Yang, "Research on omnidirectional ORB-SLAM2 for mobile robots," *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC),* 2018, pp. 1-7, doi: 10.1109/GNCC42960.2018.9018878.

[19] X. Cui, C. Lu and J. Wang, "3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment," in IEEE Access, vol. 8, pp. 67179-67191, 2020, doi: 10.1109/ACCESS.2020.2983488.

[20] Z. Yu and H. Min, "Visual SLAM Algorithm Based on ORB Features and Line Features," *2019 Chinese Automation Congress (CAC),* 2019, pp. 3003-3008, doi: 10.1109/CAC48633.2019.8996373.

[21] H. Rong, A. Ramirez-Serrano, L. Guan and X. Cong, "Robust RGB-D SLAM for Dynamic Environments Based on YOLOv4," *2020 IEEE 92nd Vehicular Technology*

Conference (VTC2020-Fall), 2020, pp. 1-6, doi: 10.1109/VTC2020-Fall49728.2020.9348738.