Mechanism Analysis and Mitigation of Visual Navigation System Vulnerability

Yawei Zhai, Yuanwen Fu, Shizhuang Wang, and Xingqun Zhan(✉)

Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai, China
xqzhan@sjtu.edu.cn

Abstract. Autonomous systems have attracted significant interest from academia and industry in recent years. To achieve accurate and reliable positioning, and to improve navigation robustness in complex operational scenarios, visual navigation has become indispensable. However, visual systems are vulnerable to many influencing factors. Therefore, this paper introduces the "vulnerability" concept for visual navigation, which originates from satellite navigation. We analyze the underlying causes of visual vulnerability and propose mitigation methods accordingly. Following the pipeline of visual navigation systems, several internal and external vulnerability factors are revealed: image blur, object motion, incorrect feature matching, biased depth recovery, repeated texture, etc. The causing and influencing mechanisms are analyzed, based on which we propose some mitigation methods. Using the well-known open-source datasets (e.g., TartanAir, EuRoc), it is proved that visual navigation is highly and frequently vulnerable due to the factors above. Some typical examples are offered to illustrate the vulnerability factors. Besides, experiments are carried out to validate the proposed vulnerability mitigation approaches, and the results show their effectiveness.

Keywords: Vulnerability · Visual navigation · Autonomous systems · reliability

1 Introduction

Future autonomous systems are expected to bring significant convenience for people's daily life. To ensure operation safety, their corresponding navigation systems must continuously provide position solutions with high reliability, while meeting the accuracy requirement. In addition, because those systems will be applied over various scenarios, the navigation systems should also provide high robustness against operational environment changes [1]. Therefore, providing high reliability and robustness for autonomous systems is one of the key research challenges in navigation field today.

The particular interest of this paper is on Visual Navigation (VN), which has been identified as one of the main navigation sensors for autonomous applications [2]. It aims at estimating the ego-motion of an agent by extracting information from raw images, and the key is to find the projections of spatial points (correspondences of pixels) in the consecutive frames. Typical VN approaches include feature-based approach [3, 4], direct approach [5], and a hybrid of feature-based and direct approaches [6]. For its prevalence,

C. Yang and J. Xie (Eds.): *China Satellite Navigation Conference (CSNC 2021) Proceedings*, LNEE 773, pp. 515–524, 2021.
https://doi.org/10.1007/978-981-16-3142-9_49

this work takes feature-based stereo visual navigation as an example. But the proposed methods are also applicable to other VN systems with appropriate modification. The baseline workflow of feature-based stereo VN is presented in Fig. 1 [7–9], where the main steps include pre-processing, feature extraction, feature matching, outlier rejection, and the final state estimation.

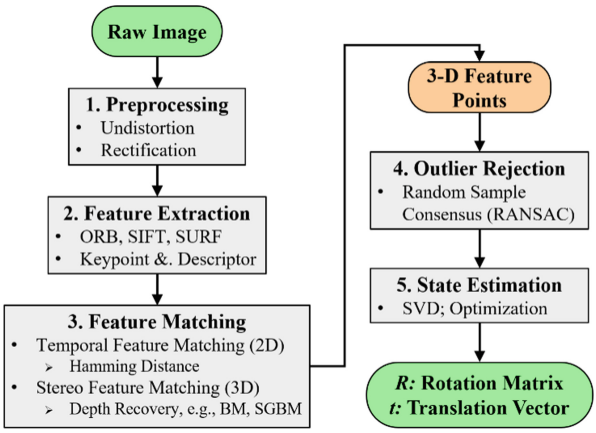


Fig. 1. Baseline workflow of feature-based stereo visual navigation

Comparing to radio navigation and inertial navigation, VN is more vulnerable to landmark uncertainty and disturbances. To quantitatively analyse its impact, this work introduces the VN "vulnerability" concept, which is inspired by the same concept from Global Navigation Satellite Systems (GNSS) field [10–12]. Regardless of the fact that there has been enormous work on improving VN accuracy and stability [13–15], it is desired to improve the reliability of VN systems under various scenarios. Although there is usually an outlier rejection step in a typical VN system (step 4 of Fig. 1), it cannot meet the reliability requirements to support autonomous applications, e.g., driverless cars and unmanned aerial vehicles. Analysing the vulnerability factors and developing mitigation techniques for VN are critical to improve navigation accuracy and reliability, and it is also necessary for assessing VN quality in real-time.

The rest of this paper is organized as follows. Section 2 provides the definition of VN vulnerability and analyses its root causes. Then, Sect. 3 presents mitigation approaches against VN vulnerability. Dataset-based experiments are carried out in Sect. 4 to illustrate VN vulnerability and validate the effectiveness of the proposed methods. Finally, Sect. 5 concludes this paper.

2 Definition and Root Cause Analysis of VN Vulnerability

VN systems are frequently subject to various threats and challenges due to its working principle. In this work, this natural fact is named "VN vulnerability", as inspired by GNSS vulnerability. For GNSS, its vulnerability is mainly because GNSS signals are

very weak during long-distance transmission. Meanwhile, there are a lot of unintentional and intentional signal interferences. Similar vulnerability analysis methods apply to VN system. In this section, we will analyze the root causes of VN vulnerability based on a baseline workflow of feature-based stereo visual navigation systems. As mentioned above, the general principles are also applicable to other VN systems with slight modifications. Comparing to Fig. 1, Fig. 2 provides a more detailed work process of VN, which lays the foundation for the following analyses.

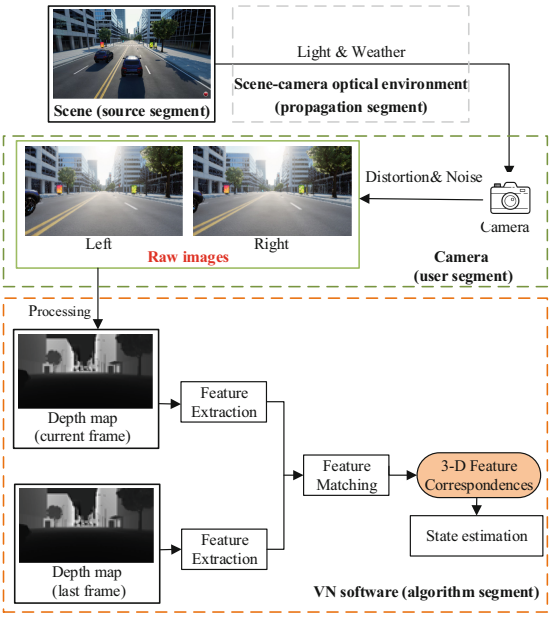


Fig. 2. All-in-loop workflow of feature-based stereo visual navigation

As shown in Fig. 2, there are four connected segments in a standard VN workflow: scene (source segment), scene-camera optical environment (propagation segment), camera (user segment), and VN software (algorithm segment). This division is an analogy to GNSS, which is divided into constellation (space segment), atmosphere (propagation segment), receiver (user segment), and user algorithm (including signal processing and state estimation). Figure 1 only addresses the algorithm segment, while the errors and outliers of visual measurements are mostly introduced in the first three segments. To capture all the sources of VN vulnerability, we provide a qualitative analysis on the first three segments (also named as "measurement segment") as follows.

2.1 Scene (Source Segment)

Just like a GNSS constellation, scene determines the richness and quality of the information in the raw images. Figures 3a and 3b show two different scenarios for an exemplary illustration. Obviously, Fig. 3a has richer information, because its textures are clear

with almost no repetition. In contrast, Fig. 3b only conveys little information, which is attached with less and repetitive texture. Therefore, extracting high-quality feature points is easier for Fig. 3a. Besides, as shown in Fig. 3c, when there are moving objects in the image, VN systems may output large error. This is because the assumption that the landmarks are static is the basis for visual navigation. To sum up, from the perspective of VN vulnerability, the scene itself is the most important factor. Scenes lacking texture, with repeated texture, or with non-static objects, are one of the main sources of vulnerability.

Fig. 3. Three typical visual scenes

2.2 Scene-Camera Optical Environment (Propagation Segment)

Similar to the atmospheric propagation errors and multipath effects in GNSS, for VN, the optical environment from the scene to the camera will also introduce disturbances to the visual measurement. Common optical environmental characteristics include light, air humidity, and weather, etc. For example, over a day period, the varying light conditions will cause different gray levels of the images. Besides, for the same scene, the images taken on rainy days and sunny days will show significant difference, given that the light from the scene to the camera may be refracted in rainy days. The ideal assumptions for VN are (a) there is sufficient and stable illumination, and (b) the light travels in a straight line from the scene to the camera. However, the actual optical environment does not necessarily meet such assumptions, which is another cause of VN vulnerability.

2.3 Camera (User Segment)

Camera is a sensor used in VN to generate raw images. Generally, the pinhole camera model is regarded as an ideal camera model. The pinhole camera model describes the relationship between a real-world spatial point $P_c = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$ (expressed in the camera coordinate system) and its image $P^{'} = \begin{bmatrix} u & v \end{bmatrix}^{\mathrm{T}}$ in a 2-D phase plane, as shown in the following equation.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \triangleq \frac{1}{z} K \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

where f_x and f_y are the focal lengths in pixels; c_x and c_y are the coordinates of the camera's optical center in the pixel plane in pixels. f_x, f_y, c_x and c_y are the intrinsics, and K is

called the camera's internal reference matrix, which is a deterministic and time-invariant parameter.

Then, let us consider the pose of the camera (called the external parameters of the camera), i.e., the rotation matrix R and translation vector t of the camera. Therefore, for a stationary point P_w in the world coordinate system, we have:

$$P_c = R \cdot P_w + t \tag{2}$$

Substituting (1) into (2), we get:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} K (R \cdot P_w + t) \tag{3}$$

Equation (3) describes the transformation relationship between a point in the scene and the coordinates of its phase plane, which is the basis for VN.

However, the imaging process is easily affected by distortions and noise, which introduces errors in Eq. (3). Generally, distortions are deterministic that can be corrected. But noise is random, and its presence can make (a) the projection relation inaccurate or (b) the grayscale of a pixel no longer accurate. The grayscale disturbance of pixels will directly affect the feature matching process. In addition, there may be motion blur, lens fogging, and lens obscuration during camera imaging, which significantly reduces the image quality. To sum up, noise, blur, and occlusion are the common sources of VN vulnerability in user segment.

2.4 VN Software (Algorithm Segment)

The goal of the VN algorithm is to determine the pose of the camera based on the original image (i.e., R and t). As shown in Figs. 1 and 2, typical process of an VN algorithm includes binocular depth recovery, feature extraction, feature matching, and state estimation. The algorithm will not introduce additional errors, but only propagate the error from measurement to state estimation. In other words, the VN algorithm maps the vulnerability factors in the first 3 segments to navigation performance.

2.5 Summary of VN Vulnerability Factors and Their Impact

This subsection summarizes the sources of VN vulnerability and their effects, as shown in Table 1. The vulnerability factors in the measurement segment mainly affect (a) the projection relationships (i.e., u and v) in Eq. (3) and (b) the gray value of each pixel. From Eq. (3), (a) directly affects the estimation of the agent's navigation state. In contrast, (b) may lead to large navigation errors by causing incorrect feature matching and inaccurate depth recovery.

3 VN Vulnerability Detection and Mitigation Techniques

To prevent the users from being threatened by large visual navigation errors, it is imperative to design a real-time VN vulnerability detection mechanism and incorporate vulnerability mitigation techniques into VN algorithms.

Table 1. Visual navigation vulnerability factors and their impact

Segment	Vulnerability factor	Negative impact
Scene	Lacking textures	Lacking features
	Repetitive textures	Incorrect feature matching & incorrect depth recovery
	Non-static objects	Incorrect state estimation
Scene-camera	Illumination change	Incorrect feature matching & incorrect depth recovery
	Rainy day	Incorrect state estimation
	Foggy day	Lacking features
Camera	Distortion	Little effect after de-distortion
	Noise	Inaccurate state estimation
	Blur	Incorrect state estimation

3.1 Real-Time Vulnerability Detection

Three vulnerability detectors are developed to alert the users when (a) the number of tracked features is not adequate, (b) there are a significant number of repetitive textures, or (c) the image quality is poor. These three factors can directly influence the availability of VN.

Similar to the "Dilution of Precision (DOP)" concept in GNSS, the number of tracked features and their spatial distribution will impact VN accuracy. In our prior work [16], we established the analytic relationship between the VN error covariance and the distribution of the feature correspondences. Accordingly, the first vulnerability detector is to alert the users when the navigation accuracy cannot meet the requirement. This detector can be expressed as follows:

$$availability = \begin{cases} 0, \ if \ \mathcal{F}(FP) > accuracy \ requirement \\ 1, \ if \ \mathcal{F}(FP) \leq accuracy \ requirement \end{cases} \tag{4}$$

where $\mathcal{F}(FP)$ denotes the estimated navigation accuracy with the matched feature point set FP.

The second detector to avoid the occurrence of severe incorrect feature matching events. At some scenes (e.g., bushes, floors), there might be a large number of repetitive textures. This will potentially lead to the event that incorrectly-matched features account for a large proportion of all the pairs. To avoid this, we define the effective feature subset as:

$$S_F = \left\{ p_i \middle| \left\lceil d(p_i) - d\left(p_j\right) \right\rceil > T, \forall j \neq i \right\} \tag{5}$$

where $\left\lceil d(p_i) - d\left(p_j\right) \right\rceil$ represents the Hamming distance between two feature points p_i, p_j; and T is the threshold for feature matching. Then, the number of effective features and their distribution will determine the availability of the VN system through Eq. (4).

The final detector is designed to filter out bad-quality images. In some cases, the image may be blurred or unclear, which means the image cannot be directly used for

visual navigation. There are existing algorithms to assess the quality of an image [17, 18], and we can set a threshold for the image quality. If an image cannot pass the three detectors above after image augmentation (e.g., deblur [19]), the VN system should send an unavailable flag.

3.2 VN Vulnerability Mitigation

In this subsection, two vulnerability mitigation techniques are introduced to mitigate the effects of non-static objects and incorrect depth recovery. It is worth mentioning that although RANdom SAmple Consensus (RANSAC) is often used for outlier rejection, there are a considerable number of outliers remaining in the "inlier set" after RANSAC. Therefore, we design the following two approaches as a supplement to remove the outliers.

Employing the real-time object detection techniques, e.g., YOLO [20], is the first technique for vulnerability mitigation. For the objects that are very likely non-static, e.g., cars, pedestrian, birds, the associated features should be considered unreliable, and thus their weights are set to zero in the process of pose estimation. This technique is useful in removing the outliers caused by non-static objects.

The principle of another approach is residual-based consistency check. The residual vector of a feature correspondence is defined by:

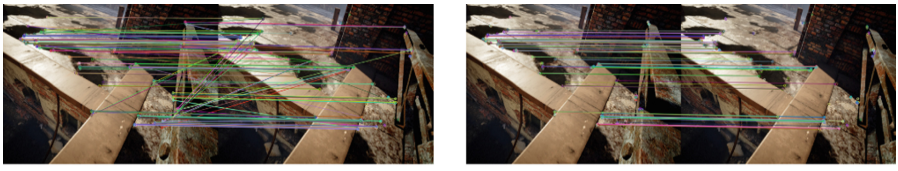
$$\boldsymbol{r}_i = \widehat{R} \cdot P_{c1} + \hat{t} - P_{c2} \tag{6}$$

where P_{c1} and P_{c2} are the coordinates of the same physical point in two camera coordinate systems (at two epochs); $\widehat{R}$ and $\hat{t}$ are the estimated rotational and translational parameters of the camera between the two epochs. If the residual vector of a feature point is larger than a preset threshold in any direction, then this feature should be removed from the pose estimation process. This technique can be used to exclude the outliers caused by non-static objects and incorrect depth recovery.

4 Experimental Results and Discussion

4.1 Examples of VN Vulnerability

Using open-source dataset, this subsection shows typical VN vulnerability examples: incorrect feature matching, non-static objects, and inaccurate depth recovery.



(a) Original feature matching result (b) Truth feature matching result

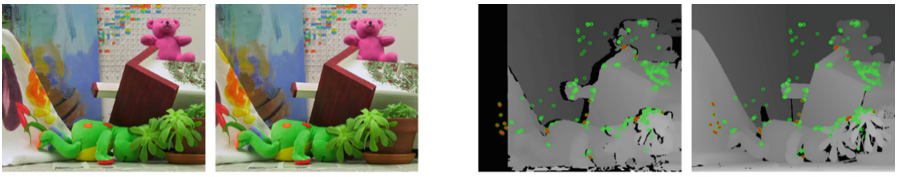
Fig. 4. An example of incorrect feature matching

Figure 4 shows an example of incorrect feature matching, where the feature extraction is based on ORB. Figure 4a is the original matching result using conventional algorithms, whereas Fig. 4b is the truth obtained by manual check. As shown in this figure, there are many incorrect feature matching events, which is caused due to close descriptors.

Figure 5 demonstrates a scenario where there are non-static objects in the image. Many extracted features (labeled by circles) locate in the passing car, which could potentially lead to large navigation errors. For vehicular navigation, this scenario is very common, and thus it is one of the major causes of VN vulnerability.

Fig. 5. An example of non-static object

Figure 6 shows a typical example for inaccurate depth recovery, where the depth estimation is realized based on Semi-Global Matching (SGBM). Figure 6a presents the raw images, and Fig. 6b shows the estimated and the true depth map. Besides, Fig. 6b labels the features with large depth errors (over 1 m) in red in the estimated depth map. For some features, the errors are significantly large, which is mainly caused by the disturbance on the pixel grayscale.



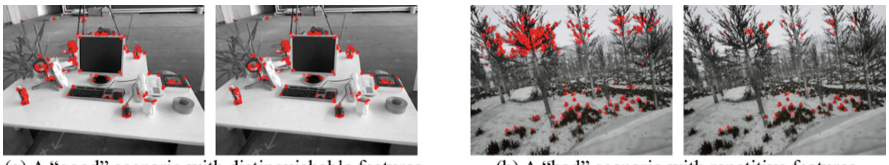
(a) Raw images (left and right) (b) Depth map (estimation and truth)

Fig. 6. An example for inaccurate depth recovery

4.2 Validation of Detection and Mitigation Techniques

Dataset-based experiments are carried out in this subsection to further illustrate and validate the proposed VN vulnerability detection and mitigation techniques. First, Fig. 7 illustrates the detector that is used to limit severe incorrect feature matching events. After removing the features with similar descriptors using Eq. (5), most features remain in Fig. 7a, while there are only a few features left in Fig. 7b. Therefore, the scenario shown in Fig. 7b is more vulnerable.

Figure 8 demonstrates the effect of object detection on removing features extracted from non-static objects. This figure shows the object detection result output by YOLO-v4, and it shows that this technique can effectively identify the objects that are likely



(a) A "good" scenario with distinguishable features (b) A "bad" scenario with repetitive features

Fig. 7. Effective analysis result of the detector design for two scenarios

Fig. 8. Object detection result output by YOLO-v4

to be non-static. Taking this scenario as an example, the VN algorithm should set the weights of the features extracted from the cars to zero in the state estimation process.

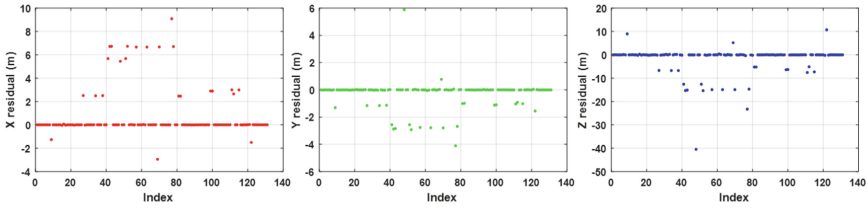


Fig. 9. Residuals of each feature correspondence in three directions

Finally, Fig. 9 shows the residuals of each feature correspondence for the scenario of Fig. 5. As shown in this figure, the residuals of some features are larger than the preset threshold, which indicates they are likely to be faulted. And by manual check, it is proved that most of these features suffer from incorrect feature matching, non-static object, or inaccurate depth recovery. Therefore, the experiment result suggests the effectiveness of the residual-based vulnerability mitigation technique.

5 Conclusion

This paper proposes vulnerability concept for Visual Navigation (VN) system, and develops corresponding mitigation techniques. Based on the baseline workflow of VN, this work points out the internal and external vulnerability factors in image generation, depth restoration, feature matching, scenario itself, etc. Besides, we analyze the generating and influencing mechanism of typical vulnerability events, such as movement, illumination changes, mismatches, and depth estimation errors. Experiments are carried out to analyze the impact of vulnerability on VN performance, and to validate the proposed vulnerability detection and mitigation methods. It is shown that incorrect feature matching,

non-static objects, and inaccurate depth recovery are three main causes of VN vulnerability. And employing object detection and residual based detection can effectively mitigate their impact.

References

1. Luo, Y., Yu, Y., Jin Z., Zhao, H.: Environment-centric safety requirements for autonomous unmanned systems. In: 2019 IEEE 27th International Requirements Engineering Conference (RE), pp. 410–415 (2019)
2. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. Found. Trends Robot. **4**(1), 1–104 (2015)
3. Gonzalez, R., Rodriguez, F., Guzman, J., et al.: Combined visual odometry and visual compass for off-road mobile robots localization. Robotica **30**(6), 865–878 (2012)
4. Cumani, A.: Feature localization refinement for improved visual odometry accuracy. Int. J. Circuits Syst. Signal Process. **5**(2), 151–158 (2011)
5. Naroditsky, O., Zhou, X., Gallier, J., et al.: Two efcient solutions for visual odometry using directional correspondence. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 818–824 (2012)
6. Scaramuzza, D., Siegwart, R.: Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. IEEE Trans. Robot. **24**(5), 1015–1026 (2008)
7. Jiang, Y., Xu, Y., Liu, Y.: Performance evaluation of feature detection and matching in stereo visual odometry. Neurocomputing. **120**, 380–390 (2013)
8. Parra, I., Sotelo, M., Llorca, D., et al.: Robust visual odometry for vehicle localization in urban environments. Robotica **28**(3), 441–452 (2010)
9. Arun, K., Huang, T., Blostein, S.: Least-squares fitting of two 3-d point sets. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-9**(5), 689-700 (1987)
10. Jing, S., Zhan, X., Liu, X., Feng. S.: GNSS vulnerability assessment method based on application suitability. In: Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014), pp. 2291–2298 (2014)
11. Hsu, L.-T.: Analysis and modeling GPS NLOS effect in highly urbanized area. GPS Solutions **22**(1), 1–12 (2017). https://doi.org/10.1007/s10291-017-0667-9
12. Zhao, X., Zhan, X., Yan, K.: GNSS vulnerabilities: simulation, verification, and mitigation platform design. Geo-spatial Inf. Sci. **16**(3), 149–154 (2013)
13. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of International Conference Computer Vision and Pattern Recognition, pp. 652–659 (2004)
14. Frahm, J.-M., Georgel, P., Gallup, D., et al.: Building rome on a cloudless day. In: Proceedings of European Conference on Computer Vision, pp. 368–381 (2010)
15. Scaramuzza, D., Fraundorfer, F.: Tutorial: visual odometry. IEEE Robot Autom. Mag. **18**(4), 80–92 (2011)
16. Wang, S., Zhan, X., Fu, Y., Zhai, Y.: Feature-based visual navigation integrity monitoring for urban autonomous platforms. Aerosp. Syst. **3**(3), 167–179 (2020). https://doi.org/10.1007/s42401-020-00057-8
17. Weken, D.V., Nachtegael, M., Kerre, E.E.: Using similarity measures and homogeneity for the comparison of images. Image Vis. Comput. **22**, 695–702 (2004)
18. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers **2**, 1398–1402 (2003)
19. Chen, T., Ma, K., Chen, L.: Tri-state median filter for image denoising. IEEE Trans. Image Process. **9**(12), 1834–1838 (1999)
20. Bochkovskiy, A., Wang, C., Liao, H., "YOLOv4: Optimal Speed and Accuracy of Object Detection". 2020, arXiv:2004.10934.