YOLO Multi-Camera Object Detection and Distance Estimation

Bojan Strbac, Marko Gostovic, Zeljko Lukac
Department for Computer Engineering and Communications
Faculty of Technical Sciences, University of Novi Sad
Novi Sad, Serbia
@rt-rk.uns.ac.rs

Dragan Samardzija
Nokia Bell Labs
Holmdel, the USA
dragan.samardzija@nokia-bell-labs.com

***Abstract* — Due to challenging requirements in the competitive market and significant software expansion in automotive industry, there is a need and opportunity to develop and implement new algorithms and solutions, which can further enhance performances, features and quality of this fast growing and constantly changing industry. The aim of this paper is to present the possibility of using cameras instead of LIDARs for distance estimation. The proposed solution is based on the YOLO deep neural network and principles of stereoscopy. This solution uses two slightly moved cameras which obtain two pictures, which goes through algorithm for stereoscopy-based measurement. And estimate distance to detected objects.**

***Keywords* — Deep learning, YOLO deep neural network, object detection, stereoscopic vision.**

I. Introduction

Modern vehicles contain a number of driver assistance systems, such as a park assist, driver monitoring and active lane assist. Those solutions are leading to a fully automated vehicles. However, the main goal is to achieve a completely autonomous vehicle which can guide itself without human involvement. Correspondingly, modern vehicles integrate a variety of different sensors: camera, LIDAR, ultrasound sensor, etc. Cameras are known to provide the most detailed information about the vehicle's surrounding, as in Figure 1. Therefore camera-based recordings can be used for many algorithms. Furthermore, the price is one of the key aspects that has to be considered in today's industry. In favor of this, and in terms of distance estimation, which is the main topic of this paper, cameras are much cheaper and therefore more profitable solution in comparison with LIDARs.

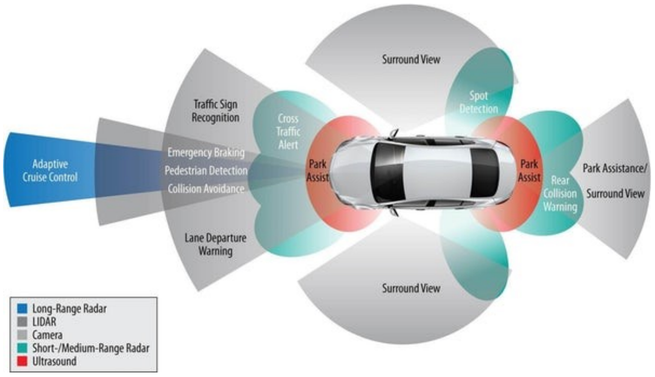

Figure 1. Sensors used by the ADAS (Advanced Driver-Assistance System) [1].

For example, a multi-camera object detection can be applied such that that two cameras are installed in the front mask of the car. Along with the appropriate algorithm, this system of cameras can substitute LIDAR for distance estimation. It can be used for the Adaptive Cruise Control (ACC) as well, which is one of the most significant features of autonomous vehicles.

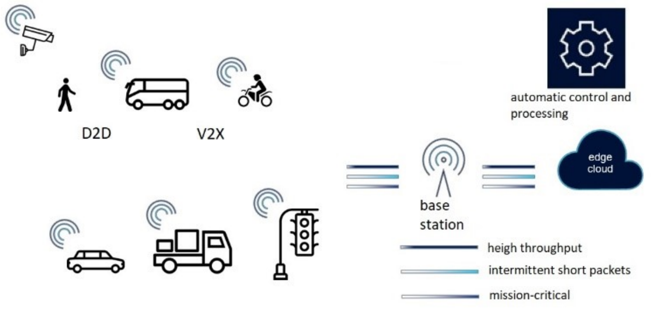

Figure 2. Integrated transportation system.

An additional and highly useful way of using multi-camera object detection is through the connectivity between vehicles and infrastructure, as depicted in Figure 2. It illustrates a system where the Device-to-Device (D2D), Vehicle-to-Everything (V2X) connectivity and edge computing implement a hyper-connected cyber-physical environment. D2D and V2X are radio-access technologies standardized by 3GPP in its Release 14 with further enhancements expected in Release 16. For example, cameras that are placed on lampposts, traffic-lights and buildings are distributing video information to neighboring traffic participant. This would enable vehicles to detect presence of potential obstacles even before they can directly see them. It is expected to lead to a much safer and efficient transportations system.

II. Algorithm

A. *Stereoscopy*

Stereoscopy (stereo recording) is a technique used to create or improve illusion of picture depth [2]. It is the essential part of the algorithm that is presented in this paper. A stereoscopy-based method for measuring the distance of objects belongs to a group of passive solutions, meaning that it does not involve the emission of any signals toward objects. This method is moderately computationally complex.

978-1-7281-8259-9/20/$31.00 ©2020 IEEE

The biggest challenge was to calibrate the cameras appropriately and to precisely choose pixels for the calculation. In order to get pictures that can be used afterwards, cameras need to be accurately installed (Figure 3(a)), otherwise (as illustrated in Figure 3(b))) large deviations can occur, possibly endangering traffic safety. Additionally, if the object, whose distance is being estimated, is far away, the estimation error will be larger.

Considering mobility, video frames taken by each camera should ideally correspond to the same time instance. The best scenario would be the one in which objects are not moving, however, the reality is quite different, and cameras must be properly synchronized in order to get adequate results.

Also, the key thing for the accurate estimation is to find pixels in both pictures, which show the same detail of the object. Performing accurate estimation is not easy due to the fact that objects are not in the same position in both pictures, because these objects are detected from different angles.

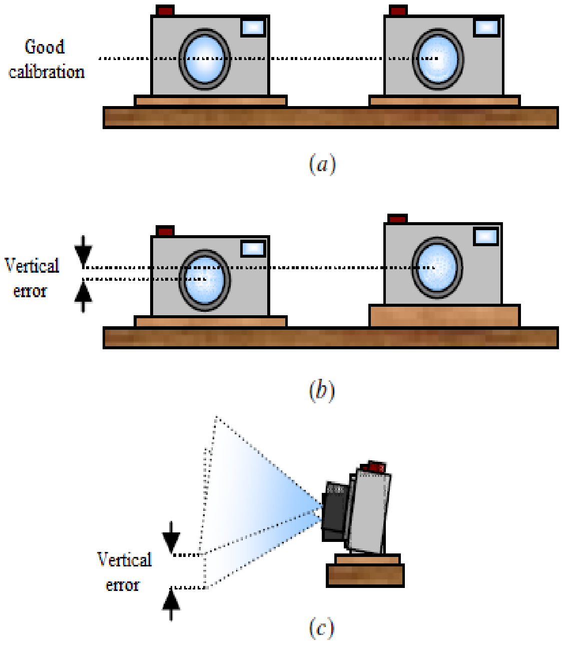

Figure 3. Well-calibrated cameras (a) and the most common calibration errors (b, c) [3].

Figure 4 depicts parameters important for stereoscopy-based measurements. **S$_L$** and **S$_R$** are representing two cameras which are at distance B from each other. φ$_0$ represents cameras Field of View (FoV). Distance to the object (the car) D can be expressed through geometrical derivations resulting in the following expression

$$D = \frac{B}{\tan\varphi_1 + \tan\varphi_2}, \qquad (1)$$

where φ$_1$ and φ$_2$ are the angles between the axis of the camera lens and the direction to the object.

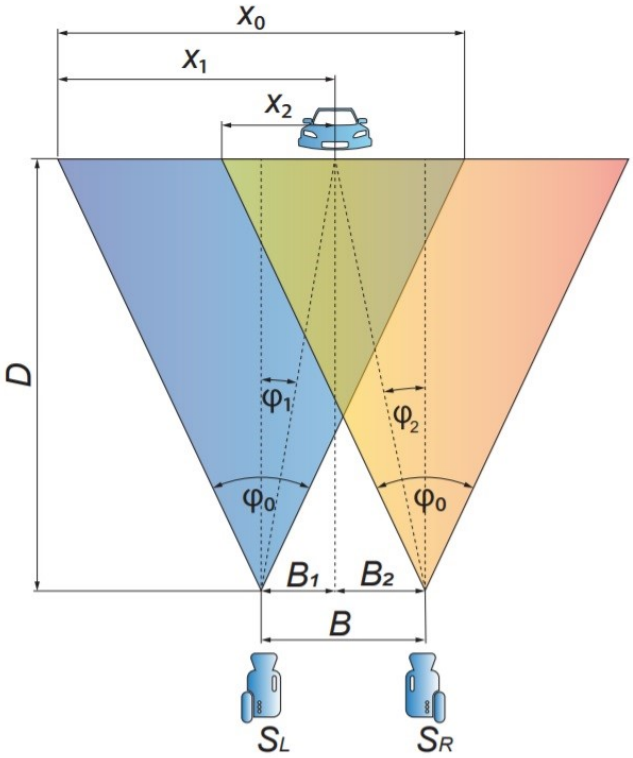

Figure 4. Parameters for the stereoscopy measurements.

Three important parameters, for further derivation, are X_0, X_1 and X_2. X_0 represents the number of horizontal pixels of the pictures, X_1 and X_2 are the numbers of pixels between the midpoint of the object's boundary box horizontal edge and left edge of the picture (X_1 is for the left picture and X_2 is for the right one).

$$D = \frac{B * X_0}{2\tan\left(\frac{\varphi_0}{2}\right)(X_1 - X_2)}, \qquad (2)$$

Therefore, we can estimate the distance to any object which appears in both pictures if we know the distance between cameras (B), number of horizontal pixels of the photo (X_0), FoV of cameras (φ$_0$) and horizontal difference between the same object in both pictures (X_1-X_2).

B. *YOLO V3 Deep Neural Network*

The YOLO V3 neural network [4] is the third generation of the YOLO neural networks and it presents a set of different components for detection and classification of objects in a picture. This neural network is able to recognize more than one object in the same picture, which belongs to the same or a different class. The majority of algorithms for object detection and classification consists of two neural sub-networks. One for the detection and one for the classification, which makes these algorithms too slow for real time automotive systems. On the other side, YOLO [7] uses a different approach. There is only one neural network which predicts multiple boundary boxes at the same time as well as the probabilities that there is an object inside them. The main advantage of YOLO V3 is its speed of execution. It is able to recognize objects very fast, but in terms of precision, it still lags behind of the top-performing object detection algorithms

[5]. YOLO V3 performs detection on three levels [6], where each is using a different grid (13x13, 26x26 and 52x52) for predicting boundary boxes.

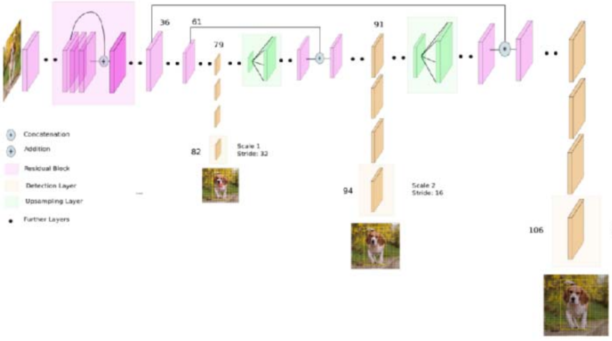

Figure 5. YOLO V3 neural network architecture overview. [4]

An arbitrary size image can be brought to the input of YOLO, because at the beginning of the algorithm the image is scaled to $N{\times}N$ pixels. In the case of YOLO V3, $N = 416$. After that, scaled picture is being divided to $S{\times}S$ cells, in the first step $S = 13$, but in later steps of the algorithm $S = 26$ and $S = 52$. It is desirable for the parameter S to be an odd number, to form only one particular cell, which is in the middle of the grid map, responsible for the object that is in the picture center. Every cell predicts M boundary boxes and confidence that there is an object inside them. In a case that the cell does not contain object, the confidence parameter is 0. But if there is an object, then confidence parameter is intersection over union between the predicted box and ground truth [4].

The YOLO algorithm outputs the coordinates of boundary boxes of all detected objects. This information is very useful for obtaining the parameters X_1 and X_2 which are then used in the mathematical expression for the stereoscopy-based distance estimation.

C. *Distance Estimation*

The algorithm for distance estimation has three significant steps. The first one is to prepare both pictures for YOLO object detection and perform classification of the objects on those pictures. After this step, the output are two arrays with the detections from both pictures.

The second step is to find objects which appear in both picture detection arrays, and to precisely determine X_1 and X_2, which are explained in section two. This step is the most complicated one of the entire algorithm. For the object detected in the left picture, the algorithm searches for the same object in the detection array of the right picture. The following criteria must be met.

1. Both objects must be from the same class (e.g., car, pedestrian, truck, etc.).
2. The object from the right picture must be closer to the left edge of the picture than the same object found in the left picture.
3. As the object is photographed with the cameras that are shifted horizontally, vertical edges of the boundary boxes from the same object on different pictures should not be different. Furthermore, in terms of horizontal edges, deviations could be minimal (less than 5% of the horizontal edge). The algorithm is searching for two boundary boxes that belong to the two pictures, with the smallest differences that fulfill the two previous criteria. The minimum mean square error algorithm is used, and it is represented as

$$mmse = P\,(w_L - w_R)^2 + (1-P)\,(h_L - h_R)^2. \quad (3)$$

Where w_L and w_R are the widths of boundary box of recognized objects in the left and right picture, h_L and h_R are the heights of the same boundary box, while P represents the ponder. Higher weight is assigned to the vertical edges.

The last step of the algorithm is to calculate the distance. At this point, all necessary parameters are already obtained, and the distance is calculated as in expression (2).

III. Implementation

The initial goal of this project was to realize a real-time implementation of the algorithm as depicted in Figure 6. Two video streams are processed independently by YOLO. In our implementation, YOLO is executed sequentially, once for each video input.

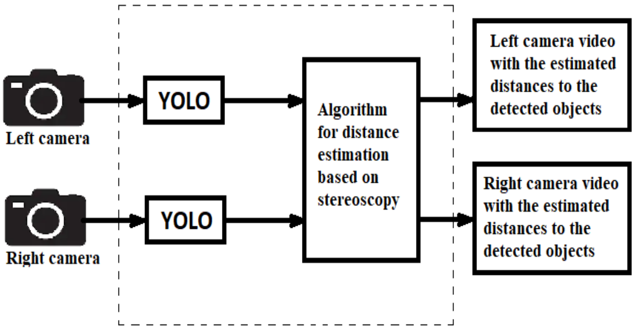
Left camera
Right camera
YOLO
YOLO
Algorithm for distance estimation based on stereoscopy
Left camera video with the estimated distances to the detected objects
Right camera video with the estimated distances to the detected objects

Figure 6. Two-camera/two-YOLO real time system.

For the implementation of this algorithm a personal computer with 16 GB of RAM, Intel i7-3930K processor and NVIDIA GeForce GTX 1070Ti graphic card was used. The YOLO V3 neural network inference is implemented on the NVIDIA graphic card applying the CUDA toolkit, including the cuDNN library.

The application-development Darknet framework [8] was used since optimized for neural network implementations, supporting usage of the C programming language, CUDA toolkit and its libraries, Darknet works directly with the grafic card resulting in a significant improvement in the speed of execution.

IV. Performance Evaluation

The performance is evaluated in two scenarios. First one is with objects that are close to the cameras (up to 2 m), and the second one is for objects that are relatively far (10 m, 20 m, 30 m). Furthermore, it was evaluated for different distances between the cameras (parameter B).

A. *Objects close to cameras (up to 2 meters)*

This type of evaluation was used to confirm that the algorithm provides accurate results in the idealized conditions with all prerequisites satisfied. The algorithm gives very good results during this test scenario. YOLO outputs very precise boundary boxes, which results in very good distance estimation. Two model cars were set up such that one is closer to the cameras than another one. The distances were measured manually, 0.84 m for the green car and 1.04 m for the yellow one. The algorithm has the average error of 4%. This is depicted in Figure 7. Furthermore, in certain situations, if the same object is not detected in both pictures, the algorithm cannot estimate its distance and the output will be “no information”. In our case chair and plant are detected only in the right picture (a lower photo Figure 7).

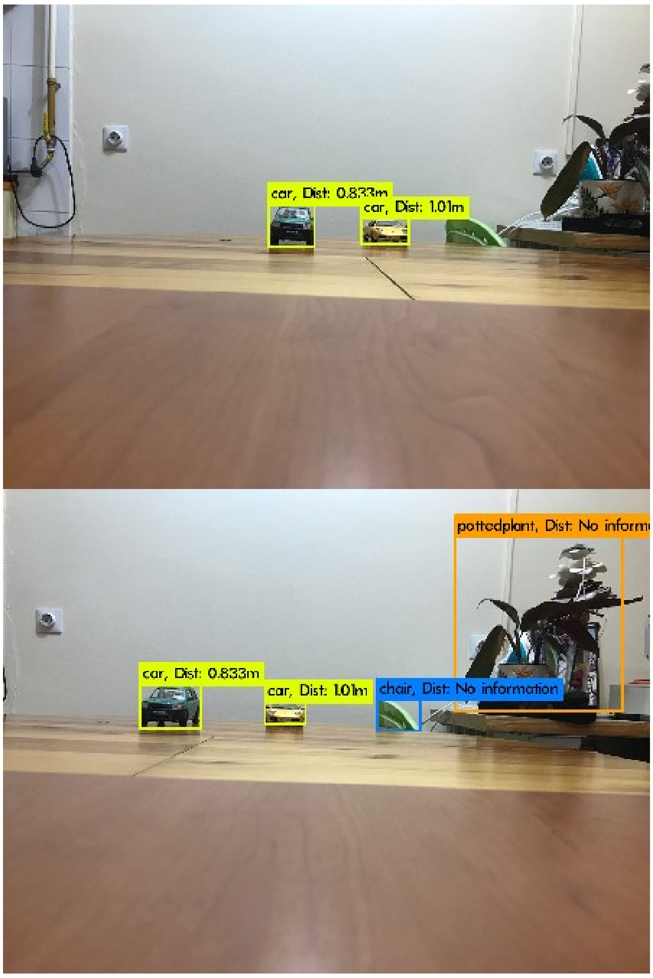

Figure 7. Output of the algorithm when testing the objects that are close to the cameras.

B. *Objects away from cameras*

The aim of this test scenario was to evaluate the algorithm performance in real-world conditions. Real cars and real traffic situations were considered. The car was positioned at three different distances (10 m, 20 m and 30m). Also, the distance between cameras was tested, with the goal to determine the optimal distance *B*. Results of the measurements are given in the following table and these are the actual outcomes (distance in meters) when the algorithm was used. Sign X in the table means that the YOLO algorithm did not detect the object in at least one picture or that the error of the estimation was greater than 50%.

Table 1. Test results in meters for different distances between cameras (*B*) as well as for different distances between object and cameras (*D*)

D[m] B[cm]	10	20	30
10	11.6	20.9	26.1
20	10.4	22.4	X
30	11.2	18.8	X
40	13.2	16.8	X

From the table it can be concluded that YOLO is not reliable for long distances and for small object. It gives very imprecise boundary boxes, that particularly noticeable when car is 30 meters away. As a prove for that error is more that 50% in the majority of cases. Unlike in the 30 meter case, for 10 and 20 meters, results are acceptable. In the best case, when parameter B was 20 cm and distance D was 10 m, error was only 4% which is considered as a successful outcome.

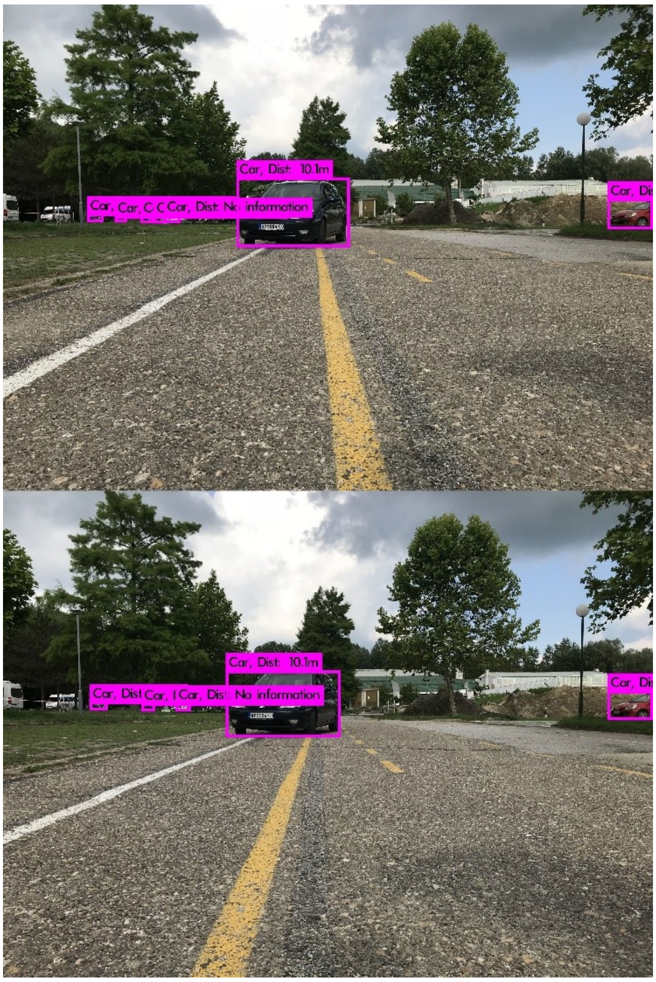
Car, Dist: 10.1m
Car, Car, C C Car, Dist: N information
Car, Di
Car, Dist: 10.1m
Car, Dist Car, I Car, Dist: No information
Car, Di

Figure 8. Output of the algorithm when testing the objects that are in real traffic situations.

Considering this test case, the YOLO algorithm in combination with stereoscopy resulted with reliable and highly useful outcomes where the distances between detected objects and cameras was below 20 m. Contrary to that, the results for 30 m and above did not satisfied expectations.

V. Conclusions

In this paper, a solution for distance estimation based on stereoscopy and the YOLO algorithm is presented. The algorithm is based on two pictures or two video streams which are obtained from the two horizontally separated cameras (stereoscopy-based). Extensive testing confirm that our algorithm could be useful for distance estimation within 20 meters. This algorithm could be used in many ADAS application, such as applications for adaptive cruise control, automatic parking, etc. It was shown that cameras, as the main sensor, could replace LIDAR in some occasions.

Regarding test scenarios when the object was more than 20 meters away from the cameras, problems occur mostly because of unprecise YOLO boundary boxes. Every pixel of error causes significant deviations in terms of the measurement accuracy.

There are several ideas and approaches how to improve the presented solution leading to more accurate estimation. For example, problems caused by the YOLO detector could be eliminate with the usage of a detector which is more precise or with the neural network that is trained differently than the one used in this work.

References

[1] "ADAS and the System Engineering Challenge." [Online]. Available: https://www.mentor.com/products/fv/resources/overview/adas-and-the-systems-engineering-challenge-ea1f3799-bf5f-49d7-a97c-86aca05165ee. [Accessed: 30-Sept-2019]

[2] "Stereoscopy." [Online]. Available: https://www.britannica.com/technology/stereoscopy. [Accessed: 30-Sept-2019].

[3] Jernej Mrovlje & Damir Vrancic (2008), Distance measuring based on stereoscopic pictures

[4] Kathuria, A., (2018), *What's new in Yolo v3?* Available on: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b (accessed in october 2019.)

[5] R.B. Girshick. Fast R-CNN. *CoRR,* abs/1504.08083, 2015. 2,5,6,7

[6] Redmon J., Farhadi A. (2018), Yolov3: An incremental improvements

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016), You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[8] "Darknet neural network framework, Github repository." [Online]. Available: https://github.com/pjreddie/darknet. [Accessed: 3-April-2019.]