Vehicle Speed Estimation with 3D Bounding Boxes

Author

MUHAMMAD SAFDAR

Regn Number

203956

Supervisor

DR. HASAN SAJID

Co-Supervisor

DR. M. JAWAD KHAN

R&AI DEPARTMENT

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

AUGUST, 2021

Vehicle Speed Estimation with 3D Bounding Boxes

Author

MUHAMMAD SAFDAR

Reg. Number

203956

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

DR. HASAN SAJID

Thesis Supervisor's Signature: _____

R&AI DEPARTMENT

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

AUGUST, 2021

Declaration

I certify that this research work titled "*Vehicle Speed Estimation with 3D Bounding Boxes"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

MUHAMMAD SAFDAR

2017-NUST-MS-RIME-203956

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

MUHAMMAD SAFDAR

Registration Number

203956

Signature of Supervisor

DR. HASAN SAJID

Copyright Statement

Acknowledgement

All praise is for Allah who is the sustainer of all the worlds.

I am thankful to my Abbu and Ammi for giving me the best of everything.

I am thankful to my beloved wife Javeria and wonderful son Musa for sharing the burden with me.

I would like to thank my supervisor Dr. Hasan Sajid, co-supervisor, GEC members Dr. Jawad Khan and Dr. Karamdad for their valuable input.

Finally, I would like to express my gratitude to all the individuals who have helped me in any way during my thesis and life.

Dedicated to my father, M. Anwar Shad

Abstract

Speed Estimation makes use of computer vision and machine learning algorithms in order to detect, keep track, and estimate speed of vehicle under surveillance. Speed estimation can be integrated with other traffic analysis solutions like vehicle counting, classification and license plate recognition system. My approach uses the vanishing points (VP) geometry and scene scale in the scene to calculate a perspective transformation matrix. This transformation makes us able to simplify the task of detecting 3D bounding boxes to the task of detecting 2D bounding boxes with one extra parameter using an efficient object detector. Additionally, I have proposed algorithm of automatic video mask generation which eliminates this manual step from pipeline. This improved algorithm has improved speed measurement accuracy by reducing mean speed measurement error by 5% and the median speed error to 3%.

Key Words: *3D Bounding box, Vehicle speed estimation, Traffic serveilance, Camera calibration*

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

The modern advancement in commercially available cameras has improved their video quality and reduced their price. This technological advancement has introduced boom in camera usage for traffic surveillance. As the modern research in deep learning methods associated with computer vision has matured, a lot of intelligent traffic surveillance use cases has been emerged to assist the traffic monitoring departments.

These intelligent and automatic traffic surveillance systems aim to provide information about subject vehicle being monitored like its type, make, speed, dimensions, and license plate recognition. These objectives are considered to be important aspects of intelligent transportation system design.

Intelligent traffic sureveillance system relies accurate detection on vehicle, precise camera calibration of video capturing equipment. Vehicle detection is one of the major tasks in the vehicle speed estimation pipeline. Background subtraction, keypoint feature based detection falls in classical computer vision domain and neural network-based detection approached fall in modern deep learning domain. Deep convolutional neural networks CNN are being used to extract features from images, and a supplementary layers of network use extracted features for object detection in the scene. I decided to deploy the object detector, RetinaNet [1] as a base network for vehicle detection because it gives balance between detection accuracy and low inference time.

The complete task of vehicle speed estimation is based on two different steps. First step is known as camera calibration and finding essential parameters of transformation. And the second step is continuous measurement of speed in video. The first step is usually one time in ideal scenario but there are chances of misorientation of camera facing towards road due to unknown reason. That misalignment can result in daviation from camera calibration parameters which ultimately affect measured speed value of vehicle. To handle this problem I have proposed a method to periodically perform the camera calibration task, transformation matrix calculation and automatic video mask generation. This video mask helps in eliminating unwanted parts of road from frame and focuses on region of interest.

1.2 Problem Statement

Although multiple techniques [2] have been developed and being used for vehicle speed estimation in traffic surveillance, but estimation of vehicle using monocular camera using resource-constrained computational unit is still a challenge. The current vision-based speed estimation systems heavily rely on manual camera calibration or assistance of non-vision based technologies like RADAR sensors. The latest vision based systems take use of deep learning techniques to detect, track and measure speed of vehicle using 2D bounding boxes and transformation of viewing screen to 3D roadplane coordinates. So, there is need to utilize 3D bounding boxes for speed estimation as they can serve other surveillance objectives like vehicle fine-grain classification and re-identification.

Chapter 2: Literature Review

Estimation of speed of vehicle captured by a monocular camera is a multi-step pipeline which includes detection of vehicle and tracking of detected vehicles followed by calculation of the distance they travelled using camera calibration parameters. Merging these tasks into a single pipeline is normally difficult so in the last section of this chapter, I have focused on the available methods of measuring the accuracy of the complete pipeline.

2.1 Camera Calibration

Multiple techniques have been proposed in classical computer vision for camera calibration. Most of the times camera calibration is used to rectify distortion from image like curvilinear barrel or pincushion distortion.

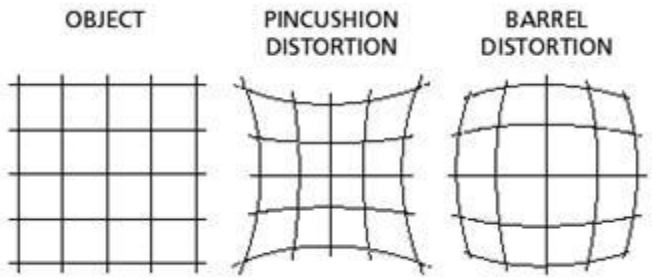

Figure 1: Pincushion and barrel distortion

Camera calibration is essential part for measurement of 3D world distances in 2D captured frames from camera. A detailed review of camera calibration techniques has already been proposed by Sochor et al. [3] In that review the author has discussed that most of the camera calibration methods are less automatic and need manual input like chessboard type pattern [2], drawing other calibration pattern on road [4], using lane marking lines on road [5] [6] or some other physical measurement of real object in frame related to scene. [7] [8]

There have been multiple techniques proposed for automatic and accurate camera calibration. Filipiak et al. [9] presented an automatic technique which was based on evolutionary algorithm, but it used zoomed footage and application was for license plate recognition. But this approach is unusable for current objective which contains multi-lane road view.

In another attempt of automatic camera calibration, Duska et al. [10] proposed a method of finding three orthogonal points which he named as Vanishing Points. The first vanishing point is calculated by intersection of all lines which corresponds to vehicle movement. Second vanishing point is calculated by finding the intersection of all the lines which corresponds to edges of vehicle which do not correspond to first vanishing point. The author has proposed a method of using hough transform on these lines to accumulate them in a diamond space based on parallel coordinates. The third and last vanishing point is just mathematical calculation of orthogonal point to previous two using vector product in homogeneous coordinates.

Scene scale is another parameter in camera calibration which enables measurement of distance on road plane. There are manual, semi-automatic and automatic approaches for scene scale measurement. Dubska et al [10] has proposed taking 3D bounding boxes of detected vehicle and finding height, width and length of detected vehicle and compare it with statistical data based on typical traffic dimension of that specific country. This approach has further been improved by Sochor et al. [3] who tried to fit 3D model of specific vehicle to the detected vehicle in frame.

2.2 Object Detection

With the recent advancement of convolutional neural networks, object detection task has significantly matured. There are two common categories in object detectors, single-stage detector and two-stage detector.

The single stage object detectors work on a structure of anchor boxes as output last layer of neural network. Each anchor box corresponds to a possible bounding box with a classification score to determine to which class this object belongs to. [11] The alignment of bounding box to the object is done by regression. In these detectors, [12] [13] there are more than one anchor boxes at output for one object to rectify this problem non-maximum supperession is used to get only one bounding box for each object.

Two-stage detectors like Faster R-CNN [14] uses a first part of the neural network for region proposal and the later part of neural network does object classification and positioning. This architecture delivers higher accuracy on the cost of inference time and compute power.

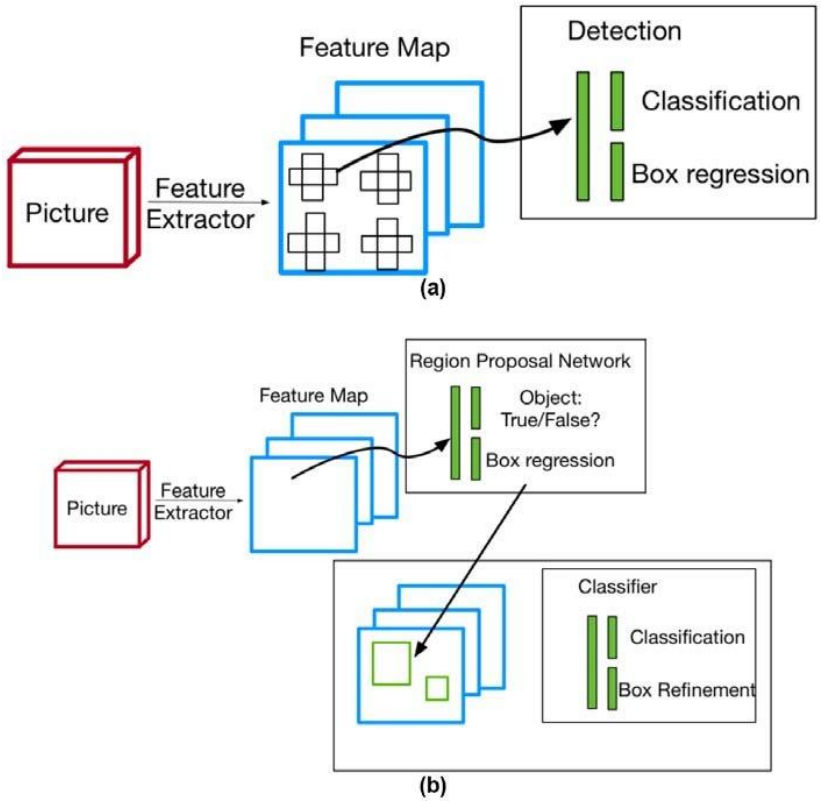

Figure 2: Single stage and Double stage object detector structure

2.3 Vehicle Detection

Most of the cameras deployed on roadside are stationary. Therefore, many techniques of background subtraction have been proposed for vehicle detection. But these methods are subject to lighting conditions and shadows can affect there performance significantly. Moreover, these methods cannot handle occlusion. Elder fit and Corral-Soto [15] a mixture model to distribute vehicle dimensions on data that is labeled. This method is used together with geometry of scene which is known before hand to calculate configuration for blobs of vehicle that were estimated with background subtraction.

2.4 Bounding Box Construction

The technique of 2D bounding box for traffic surveillance is easier than 3D bounding boxes but later has added benefits in other traffic surveillance objectives like fine-grained vehicle classification [16] [17] and vehicle re-identification [18].

Dubska et al. uses object contour detection network [19] after detecting vehicle with Faster R-CNN for vehicles in frame. The countour of vehicle are similar to segmentation output. Tangent lines from vanishing point to this contour are then used for construction of 3D bounuding box. But in this method the order of using vanishing points was important based on the movement angle of vehicle from center line.

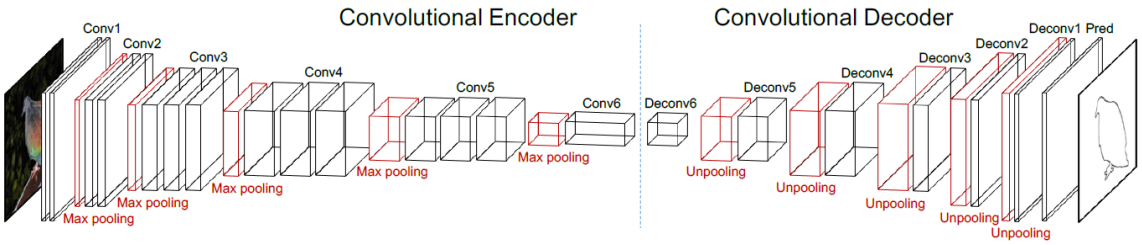

Figure 3: Architecture of the fully convolutional encoder-decoder network

2.5 Vehicle Tracking

Vehicle tracking is crucial part in vehicle counting and speed estimations tasks. Since object detector may fail in some of the frames, a reliable object tracker is required. Kalman filter has been a robust method to handle the task of vehicle tracking. Recent advancement in deep learning methods has led to object tracking networks which use convolutional neural networks together with recurrent neural networks [20] [21]. For my project I have found out that a basic object tracker which is capable of association of vehicle in subsequent frames based on IoU is enough.

2.6 Vehicle Speed Estimation Datasets

Sochor et al in presented a detailed review of evaluation of vehicle speed estimation methods, camera calibration techniques and evaluation datasets. Authors has stated in his review [22] that most of the results that has been published are calculated on relatively smaller datasets and ground truth of speed values is known for very few vehicles. And most of the datasets for previously published research are not available for public. Thus, author presented his own dataset called BrnoCompSpeed [23]. It contains 1-hour long videos for 7 different locations with three different camera angles for each location. The collective video content contains around 20 thousand vehicle instances with ground truth values of speed with the help of laser gates.

There is another dataset from Luvizon et al. [24] which contains 5 hours of video from one intersection alone. The dataset has ground truth speed values measured by inductive loop installed under the intersection. The mapping of vehicle with ground truth is done by license plate recognition for which author has proposed his own pipeline of license plate recognition. The pipeline starts from generating candidate regions around the horizontal lines of moving vehicle. T-HOG descriptor [25] is then used to validate these regions and SVM classifier to classify them. For tracking task, a few feature points within the license plate are selected and pyramidal KLT tracker [26] is used for tracking. For vehicle speed estimation, a manual homography matrix determines the real-world coordinates of those feature points.

Figure 4: Sample image from Luvizon speed dataset

Chapter 3: Proposed Algorithm

The main task of my algorithm is to get 3D bounding box of vehicle using monocular camera. The camera should be placed above the road plan and there should be no curvilinear distortion in the image. Then a common KLT tracker is used to track vehicle and develop associations with previous frames. In the end I use framerate of camera for time difference and frontal bottom center of 3D bounding box as reference point for calculation of speed. I will describe the pipeline as a whole and then I will provide detailed explanation if each part in sections.

The camera calibration process is not the part of main pipeline but this process can be done on daily basis as it is automatic process so it will adopt with any minor movement in camera position or angle. The first part of the pipeline is to apply a mask on viewing window which filters out the unwanted scene from frame like adjacent lane of road or parked vehicles on side of road. In the second step I apply perspective transformation using position of vanishing points which are calculated from camera calibration. Further in pipeline I use a pretrained model RetinaNet [1] object detector to detect vehicles. The output of this step is 2D bounding box with one extra parameter. This output is then goes to vehicle tracking and 3D bounding box construction. IoU matric is used for vehicle tracking in two consecutive frames. After getting 3D bounding box on transformed image, it is then undergoes inverse perspective transform to get 3D bounding box real scene. The center point of 3D bounding box's bottom of frontal edge is then used for interframe distance covered in pixels. The pixel distance then undergoes transformation to get real-world distance covered on road plane.

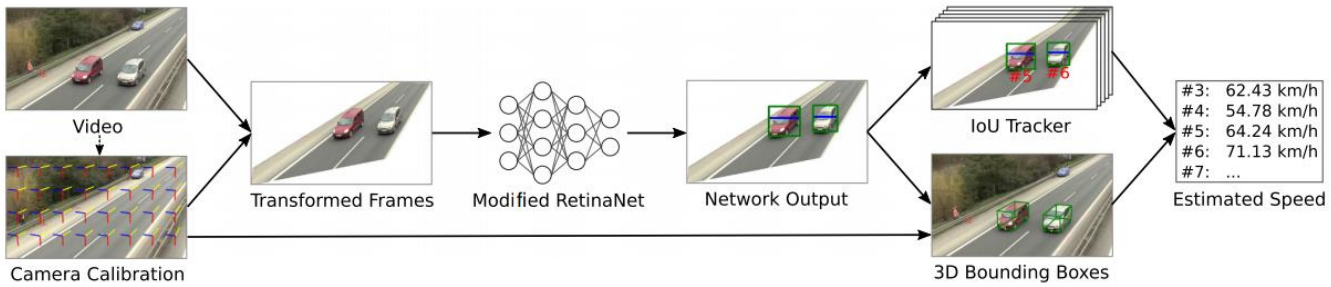

Figure 5: General vehicle speed estimation pipeline structure

3.1 Automatic Construction of Video Mask

In the reference work [27] author has used manual steps to get image mask. This mask is normally used to filter the unwanted vehicles from the scene. Although this step is required only once while installation of camera, but it is prone to errors in longer term use where camera may get disoriented due to any reason which can lead to miscalculations. So, I have proposed an automatic approach for calibrating video image mask frequently to adopt any disorientation in camera. This calibration can be done once in a week to keep image mask updated. Following the algorithm for automatic image mask construction using optical flow, [28]

1. Track feature points via KLT tracker [26] got get vehicle motion lines.
2. Filter out all the lines which are not in the direction of motion.
3. Filter out all the line which have small length.
4. Get convex hull around all these lines left after filtering.
5. Draw tangent lines from VP1 on both sides of convex hull.
6. Draw tangent lies from VP2 to both sides of convex hull.
7. The resulting closed shape after intersection of these four lines will be image mask

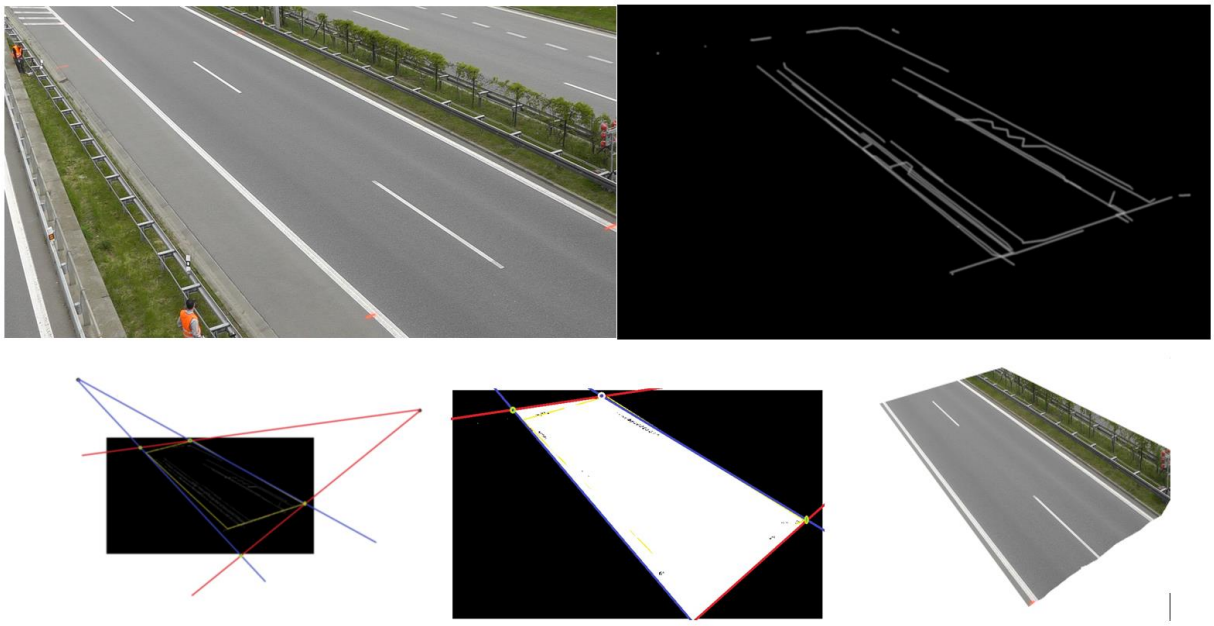

Figure 6: Automatic image mask generation steps (left to right)

3.2 Camera Calibration from vanishing points

The procedure of calculating camera projection matrix **P** from detected vanishing points has been a classical computer vision problem and several solutions are present in literature. Projection matrix **P** transforms a world point $[x_w, y_w, z_w, 1]'$ into point $[x_p, y_p, 1]'$ in the image plane i.e

$$\lambda_p[x_p, y_p, 1]' = \mathbf{P}[x_w, y_w, z_w, 1]'$$

The projection matrix can be decomposed into three matrices, one with intrinsic parameter of camera and two having extrinsic camera parameter known as rotation R and translation T

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}\ \boldsymbol{T}]$$

If we assume of zero skew, and location of principal point is in the center of image plane, aspect ratio is unity, the only parameter left is focal length *f*. by using first two vanishing points we can find third vanishing point and this parameter *f* too. Let say VP1 is **u** and VP2 is **v**, then principal point P and third vanishing point VP3 as **w** can be calculated as

$$\boldsymbol{u} = (u_x, u_y)$$

$$\boldsymbol{v} = (v_x, v_y)$$

$$\boldsymbol{p} = (p_x, p_y)$$

$$f = \sqrt{(U - P).(V - P)}$$

We can use this focal length parameter to represent **u**, **v** and **p** in image plane coordinates in 3D as $\overline{\boldsymbol{u}}$, $\overline{\boldsymbol{v}}$ and $\overline{\boldsymbol{p}}$

$$\overline{\boldsymbol{u}} = (u_x, u_y, f)$$

$$\overline{\boldsymbol{v}} = (v_x, v_y, f)$$

$$\overline{\boldsymbol{p}} = (p_x, p_y, f)$$

$$\overline{\boldsymbol{w}} = \overline{\boldsymbol{u}} \times \overline{\boldsymbol{v}}$$

To get roadplane normal vector

$$n = \frac{\overline{\boldsymbol{w}}}{|\overline{\boldsymbol{w}}|}$$

$$\rho = [n^T, \delta]^T$$

With this known roadplane ρ we can project any point from image plane to real world points.

3.3 Image Transformation

There are some rules for image transformation regarding camera position and orientation. The camera must be placed above the road plane and resultant frame from camera should have no curvilinear effect. One other consideration is that principal point should be in center of the image. After getting vanishing point as in any point in the image can be projected on road place. To measure distances on road place one last parameter of camera calibration is required which is known as scene scale.

The target of image transformation is to get an image so that the lines related to one vanishing point get parallel to one image axis and line related to other vanishing point get parallel to another image axis. And lines related to third vanishing point should be preserved in original direction. For this task I used perspective transformation. The benefit of this perspective transformation is that axis of vehicle that corresponds to those vanishing poing get aligned with image axis and any kind of distortion gets eliminated.

In the reference work [27], the author proposed an algorithm in which he used VP2 and VP3 for perspective transformation and he used complete frame for calculating parameters of transformation. Author also mentioned that this approach sometimes failed in specific camera positions. This failure was due to distorted and very small part of image that was transformed by inadequate transformation. To rectify this issue author has presented a manual approach to crop the frame. Moreover, that approach failed to deliver transformation for the VP1 and VP2 pair.

Now I have proposed an automatic approach which sets the condition byitself to get the maximum information from transformation. First part of the approach is to use mask of concerned road section which covers the speccific lane to be monitored. For automatic construction of mask, I have discussed algorithm in detail in previous section. Luckily the BrnoCompSpeed dataset contain these masks already, so I have used them for consistency of evaluation. Second part of the approach is that I have set condition that atleast 80% of the pixels in transformed image should relate to inside of the mask. I have observed that this small tweak has made the process automatic and improved the speed measurement accuracy.

The steps for transformation are as following:

1. From three possible pairs of three vanishing points select either VP2-VP3 or VP1-VP2.
2. For each of the chosen vanishing point, put two lines which are tangent to mask. This step will output four lines.
3. These four lines will intersect with each other and will give four intersection points.
4. Make pair of each intersection point with the output frame of transformed image such that it maintains the vehicle movement direction.
5. Find perspective transformation from four intersection point to four points of desired transformed frame.
6. Apply the transformation on the masked image. There should be atleast 80% of the area from original mased frame present in transformed image. If this is not the case them crop the mask frame one pixel from bottom and repeat from step 2. If the condition is satisfied then finalize the transformation for use.

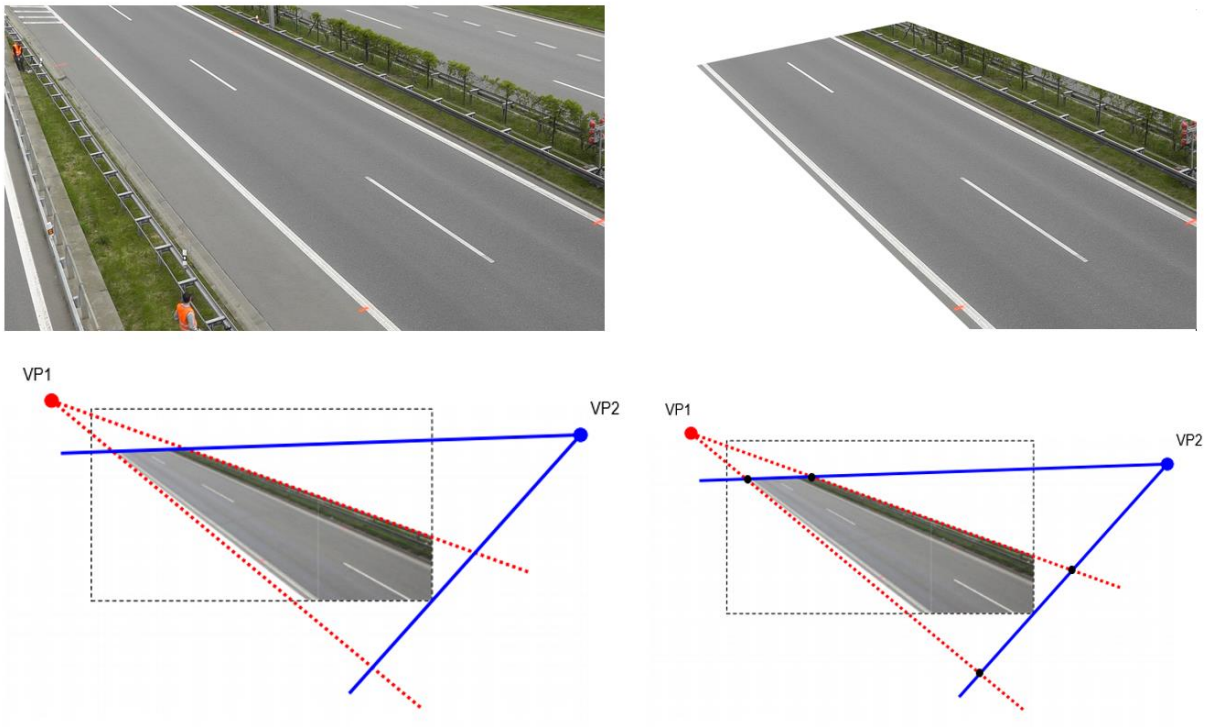

Figure 7: Step 1-4 Perspective transformation algorithm using VP1-VP2

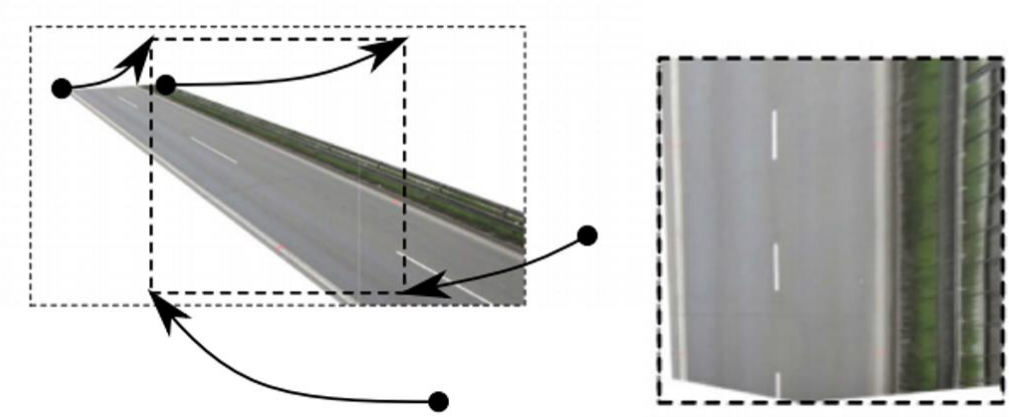

Figure 8: Step 5-6 Perspective transformation algorithm

There is one limitation in this method as suggested in that line intersecting the pair of vanishing point should not intersect the mask section of the frame. In BrnoCompSpeed dataset this limitation is not faced as there is condition of three vanishing point to be orthogonal. The third pair VP1-VP3 which was not suggested in the transformation algorithm was due to fact that line connecting these points always intersect the masked section of frame. It also simplifies the task of 3D bounding box reconstruction.

3.4 Parameterization of 3D bounding box

A 3D bounding box (BB) contains 12 edges and 8 vertices. Since we have aligned 3D bounding box with vanishing points and lines connecting to vanishing point are aligned with image axis after transformation, we get 8 of the 12 edges aligned with image axes. Since 2D BB encloses 3D BB, this makes us able to represent 3D bounding with 2D BB with one extra parameter as c_c which is by definition a relative height of 3D bounding box in 3D bounding box. This parameter is calculated by calculating vertical distance between top front facing edge of 3D BB and top edge of 2D BB. This distance is then divided by total height of 2D bounding box to make the parameter in [0, 1] range for ease in training process.

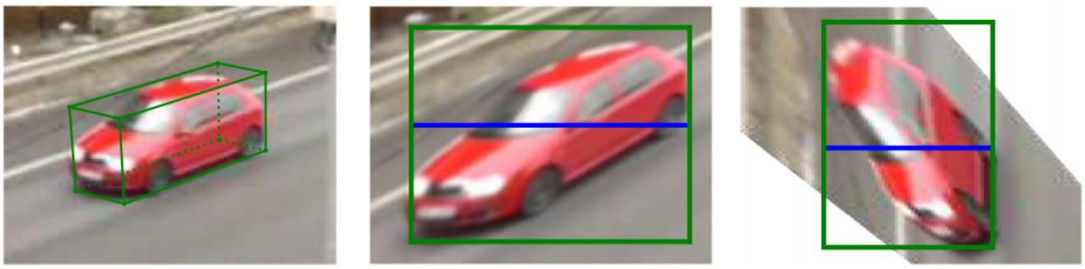

Figure 9: Parameterization of 3D bounding box

3.5 Re-Construction of 3D bounding box

The algorithm to cinstruct3D bounding box from 2D bounding box with one extra parameter is derived from previous work of. But due to improvements in the perspective transformation algorithm, some additional cases have emerged, so I have generalized algorithm for these new cases.

The steps to re-construct follow the check of vanishing point position, specifically the point which was not used in the transformation. To proceed further we need to know the relative position of this vanishing point with respect to transformed 2D bounding box. For this we first apply same perspective transformation of vanishing point let say VP. Because of the geometry of three vanishing points there are two possible relative positions of VP with respect to bounding box. This VP is either below or above the bounding box. Let's first take the case in which VP is above the 2D bounding box.

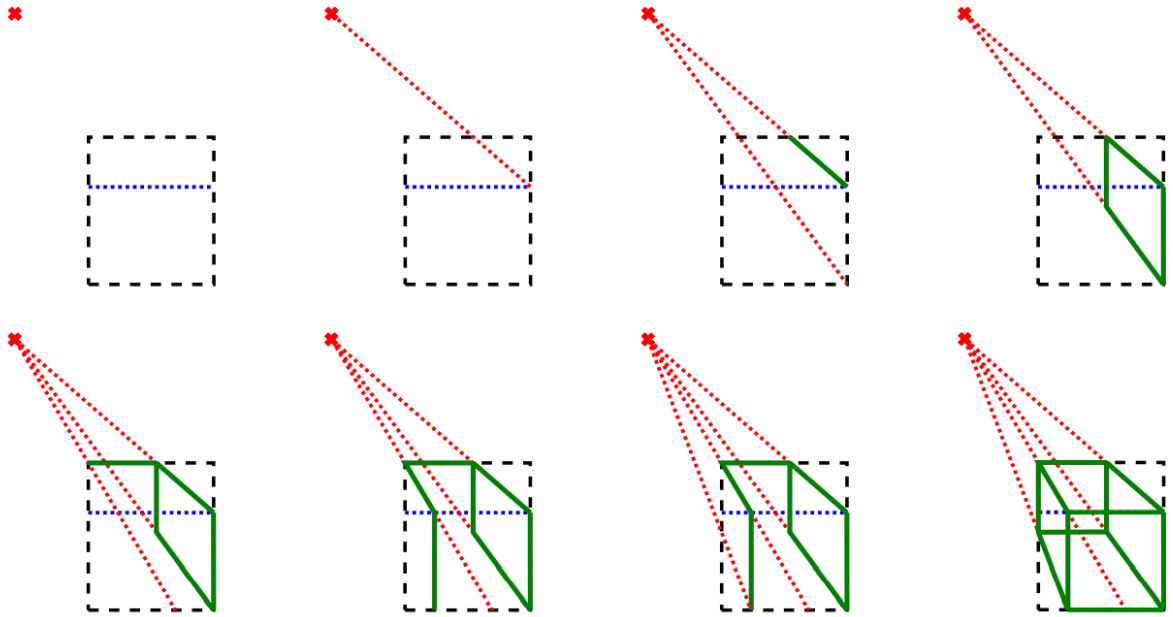

Figure 10: 3D bounding box construction bounding box is on right side of VPU

In this case, horizontal position of VP and 2D BB can be placed in three possible options. If VP is on the right side of 2D BB then the left most end of line corresponding c_c will be vertex of 3D bounding box.

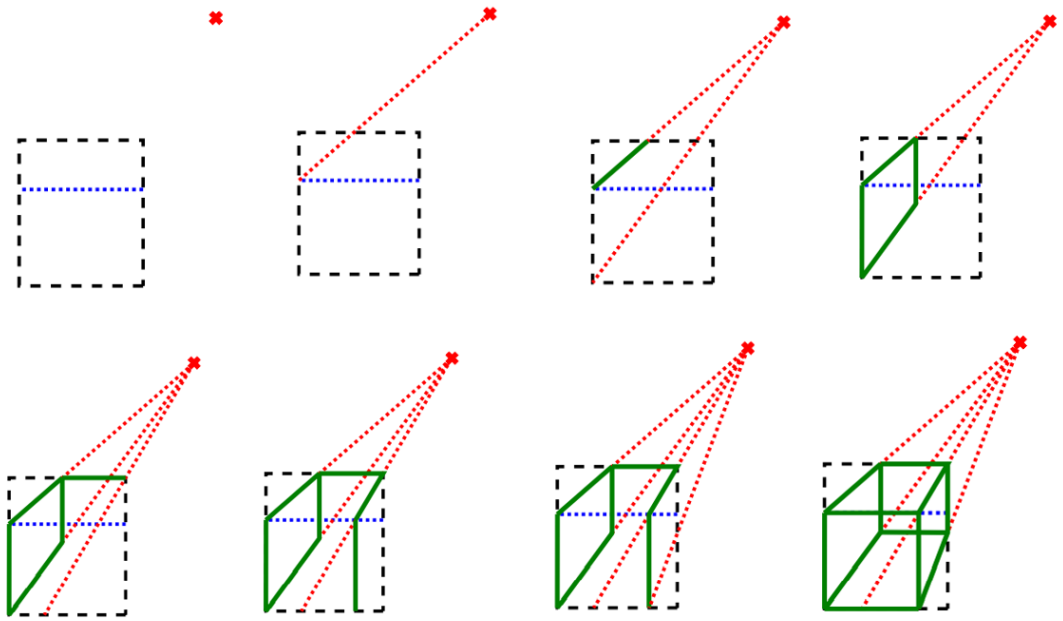

Figure 11: 3D bounding box construction bounding box is on left side of VPU

On the other hand, if the VP is on left side of 2D BB then right most end of the line is used for vertex. The third possible option left is if VP is in the center of 2D BB then any one of the end of the line can be used. The process is shown in figure. In this process VPU is on the left of the 2D BB.

To handle false output from neural network, I have proposed to detect and ignore that detection. This event can be detected when one of the lines of 3D BB goes beyond the area enclosed by 2D BB.

Once the 3D bounding box is constructed, the inverse perspective transformation is used to get real world view of 3D bounding box.

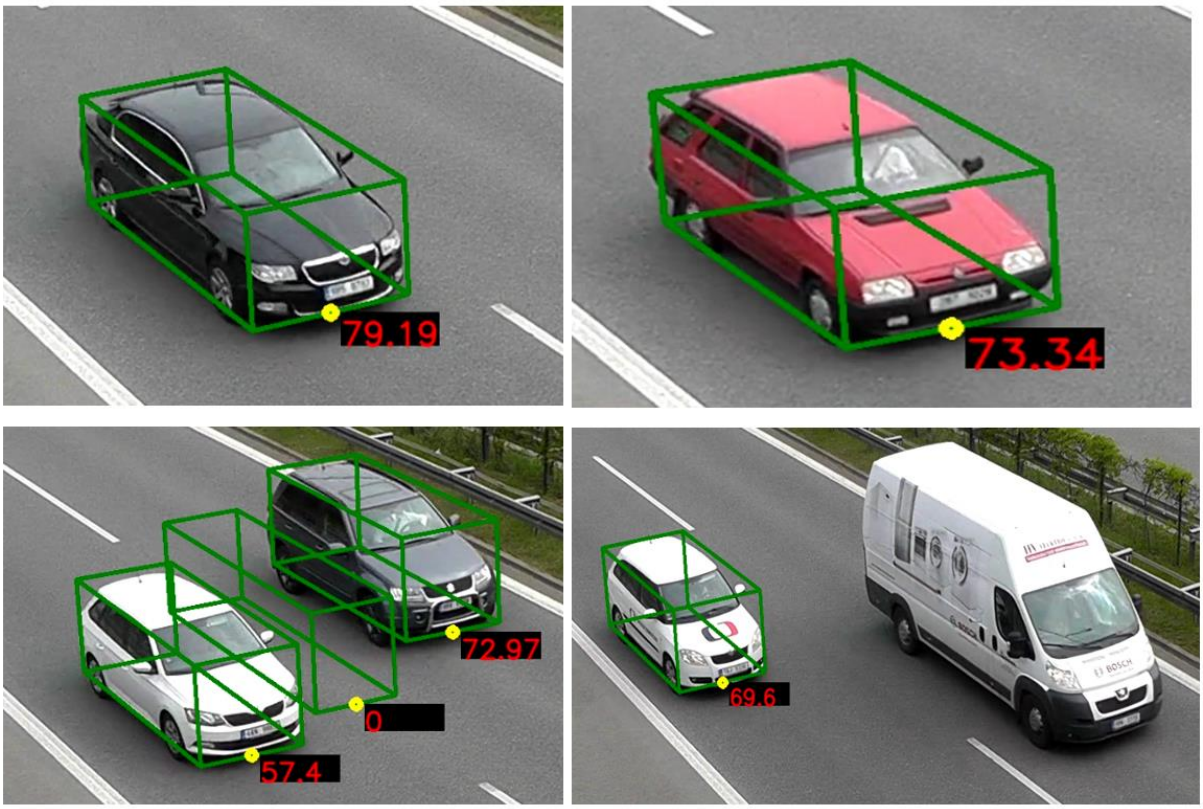

Figure 12: Top: Good vehicle detection, Bottom: Failure cases

3.6 Vehicle Tracking

The output of the vehicle detector is 2D bounding box along with parameter c_c in each frame of inference. To make tracking smooth and robust some rules are defined in tracking process. Once I parse 2D bounding boxes from inference of network, these boxes are then compared with

previous frame based on IoU matric for each track. If the IoU score of a specific detected box is more than 0.2 for atleaset one track, then bounding box with highest IoU is appended to the track. If no track has IoU more than 0.2 then one new track is initiated, and that detection gets associated with it. If any track gets no detection associated with it for last 10 frames then it gets moved to result. I discard the detections in any track which are close to boundary of frame. Algorithm also discards any track which has low age or low distance or in other words which has less than 5 detection and only moved 100 pixels or less.

3.7 Speed Estimation

3D bounding box re-construction is done in parallel to vehicle track accumulation. Only valid detections which pass the checks of tracking are moved further down the pipeline for 3D BB re-construction. This process has been explained in previous sections. For speed estimation task I have choosed frontal bottom center of 3D BB as reference point for calculation od distance covered in two consecutive frames. First this distance is in pixels which cannot be used for real speed estimation. Camera calibration parameters are used to get a transformation from 2D road projection on from to real world road plane.

$$\rho = [n^T, \delta]^T$$
$$\overline{\boldsymbol{p}} = [p_x, p_y, f\ ]^T$$
$$\mathbf{P} = -\frac{1}{\left[\overline{\boldsymbol{p}}^{\boldsymbol{T}}, 0\right].\rho}\overline{\boldsymbol{p}}$$

Since I have selected reference point, which is very close to road plane, so it gives better results as compared to any other reference which are on height from road plane. Since roadplane is at unit distance we need a third camera parameter, λ for real world scaling

$$d = \lambda||P_1 - P_2||$$

To measure the average speed of vehicle in a track I have used same method as in reference work, by calculation median of speed values of interframe detection in complete track. Given a

tracked car with reference point p_i and timestamps t_i where i =1, … N the speed can be calculated by projecting reference points p_i to the ground plane P_i

$$v = \underset{i=1\ldots N-\tau}{\text{median}} \left(\frac{\lambda^*_{reg} \|\mathbf{P}_{i+\tau} - \mathbf{P}_i\|}{t_{i+\tau} - t_i} \right)$$

For the stability of speed and avoid noise in sampling, I have used $\tau = 5$, take next sample for speed measurement after 5 frames.

Chapter 4: Evaluation and Testing

This chapter discusses about evaluation setup and testing results with different setup. I have also discussed and compared evaluation setups of reference work and setups of evaluation dataset being used in this project. In the attached resource CD, I have included two videos with 3D bounding box detection and speed estimation values. These two videos correspond to two different setups, one with VP2-VP3 and other with VP1 and VP2. The false positives of the network are also shown to showcase which were removed at the stage of tracking.

4.1 Speed measurement accuracy

To evaluate modified pipeline and compare it from reference work, I have used same test cases as done by author. The BrnoCompSpeed dataset [23] has a split C which has been used for evaluation. The evaluation results can be referred in table. The evaluation results of the reference work has been provided by author. Which are named as *Previous3D* I have also compared with original method by Dubska et al known as *DubskaAuto*. And its modified and enhanced version by Socho et al. He has implemented two methods: *SochorAuto* and *SochorManual*, which are more accurate approaches.

For evaluation comparison I have named this modified approach as *Modified3D* in table. All the test runs are done on Intel core i7 7th gen 16GB RAM and the GPU Nvidia 1080Ti. The results shows that models with relatively bigger input size give better results on the cost of inference time. Results also show that VP2-VP3 gives better performance as compared with pair VP1-VP2.

Model	VP Set	Input Size	Error Mean	Error Median	Recall
Dubska Auto	-	-	8.22	7.87	90.08
Sochor Auto	-	-	1.10	0.97	83.34
SochorManual	-	-	1.04	0.89	83.34
Previous3D	VP2-VP3	640x360	0.86	0.67	89.32
Modified3D	VP2-VP3	640x360	0.79	0.60	83.32
	VP1-VP2	360x640	1.17	0.88	86.32

Table 1 Comparison of *DubskaAuto, SochorAuto, SochorManual* & proposed method

4.2 Evaluating the influence of recall on accuracy

Since all the approaches in comparion list have used their own tracking algorithm and post processing steps after detection, it is quite possible that finalized tracks will be different in count and ordering. This fact results in variance to recall in the evaluation comparison. A method with better detector and tracker can yield higher recall and a method with weaker detector and tracker will suffer from the lower recall. To eliminate this effect, I have filtered only those ground truth tracks which yielded 100% recall on all methods. In other words which geve proper detection and tracking. Thus 7274 tracks have been marked as ground truth from 13,703 tracks of BrnoCompSpeed's split C dataset and the output results have been shown in table.

Method	VP Pair	Input Size	Mean Error (km/h)	Median Error (km/h)
DubskaAuto	-	-	8.16	8.35
SochorAuto	-	-	1.05	0.90
SochorManual	-	-	1.08	0.89
Previous3D	VP2-VP3	640x360	0.83	0.84
Modified3D	VP2-VP3	640x360	0.79	0.68
	VP1-VP2	360x640	1.02	0.87

Table 2 Comparison of speed measurement error on filtered tracks

4.3 Computational cost

The normal surveillance camera has 25 FPS for real time traffic monitoring. After evaluation for inference time, all my test cases run at more than 30 FPS at mentioned hardware setup which is more than real time. The BrnoCompSpeed dataset contain videos at 50 FPS. If the hardware setup is of lower specs, then one should resample videos at lower FPS to meet the realtime nature of traffic speed estimation task. I donot have FPS information of evaluation setups

I am comparing with but it is clear that *SochorAuto* and *SochorManual* uses Faster R-CNN which is computationally heavy than RetinaNet that is used in this project.

#	Detector Type	Input Dimension	FPS
1	RetinaNet 3D	480x270	120
2	RetinaNet 3D	640x360	70
3	RetinaNet 3D	960x540	30

Table 3 Output FPS with different model sizes on Nvidia GTX1080ti GPU

Chapter 7: Future Work

This chapter presents the discussion about future action item which can further improve speed measurement accuracy

5.1 Better detector

The speed estimation task dependes accurate calculation of distance measured in two consecutive frames. Since 3D bounding boxes are new standards for this task in computer vision domain, it is essential that 3D bounding box should enclose vehicle completely and there should be not extra space be accupied by 3D bounding box. In simple words it should enclose just the vehicle nothing less and nothing extra. For this purpose, an accurate vehicle detector is necessary. Moreover, tracking should be intelligent in cases of vehicle occlusion or large volume of traffic.

5.2 Automatic adoptive calibration

The speed estimation task is prone to error if installed camera gets disoriented or displaced due to unwanted reasons like strong winds, birds' intervention, or any other possible reason. Although I have proposed a method to periodically perform task to calculate parameters of camera calibration, video mask and perspective transformation parameters but there should be a method to detect disorientation and displacement of camera automatically.

Chapter 8: Conclusion

I have proposed several improvements to previously published reference work [27]. My improvements in algorithm of automatic video mask construction eliminates the need of manual video mask construction. The second improvement is related to automatic perspective transformation, and I proposed to use specific rules to eliminate distortion in perspective transformation which resulted in lower recall in reference work.

To gauge the speed measurement accuracy with different configurations, I have extended the analysis of experiment by providing range of different configuration. These configuarations include detector threshold value, tracker IoU threshold parameter and different check to eliminate noise in speed measurement. Based on different setting of these parameter one can get better accuracy depending on which set of parameters work best in that situation. For specific camera scene where there is distortion in detection while vehicle is leaving scene or video mask, we can set condition which will ignore detections which are close to boundary of video mask which is in direction of leaving vehicle. Since a leaving vehicle is not complete rather it is partially visible in couple of frames, the 3D bounding boxes are not perfect, there reference point does not follow the movement of vehicle, so this filtering is necessary.

All detection networks run in real time on commercially available GPUs in market. The inference time depends on model input size, and I have tested on 3 different models all having different input dimensions of frame. There is a trade of in detection accuracy and inference speed, so one must select appropriate input dimenstion for model to get optimal results.

The improved fully automatic led to better speed estimation results on BrnoCompSpeed dataset. Compared to published reference work of author, I have reduced mean speed measurement error by 5% and the median speed error to 3%. In terms of speed values, the mean speed error has been reduced from 0.83 km/h to 0.79km/h and median speed error has been reduced from 0.60 km/h to 0.59 km/h.

References

[1] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., "Focal loss for dense object detection.," *Proceedings of the IEEE International Conference on Computer Vision,* p. 2980–2988, 2017.

[2] Do, V.H., Nghiem, L.H., Thi, N.P., Ngoc, N.P., "A simple camera calibration method for vehicle velocity estimation.," *Proceedings of the 12th International Conference on Electrical Engineering Electronics, Computer, Telecommunications and Information,* pp. 1-5, 2015.

[3] Sochor, J., Juránek, R., Herout,, "Traffic surveillance camera calibration by 3dmodel bounding box alignment for accurate vehicle speed measurement," *Comput. Vis. Image Underst,* vol. 161, pp. 87-98, 2017.

[4] Maduro, C., Batista, K., Peixoto, P., Batista, J., "Estimation of vehicle velocity and traffic intensity using rectified images.," *Proceedings of the 15th IEEE International Conference on Image Processing,* pp. 777-780, 2008.

[5] Cathey, F., Dailey, D., "A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras.," *Proceedings of the IEEE Intelligent Vehicles Symposium,,* pp. 777-782, 2005.

[6] Lan, J., Li, J., Hu, G., Ran, B., Wang, L., "Vehicle speed measurement based on gray constraint optical flowalgorithm," *Optik 125(1),* pp. 289-295, 2014.

[7] Schoepflin, T.N., Dailey,D., "ynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transp. Syst.,* vol. 4, pp. 90-98, 2003.

[8] You, X., Zheng, Y, "An accurate and practical calibration method for roadside camera using two vanishing points.," *Neurocomputing,* vol. 204, pp. 222-230, 2016.

[9] Filipiak, P., Golenko, B., Dolega,, "NSGA-II based autocalibration calibration of automatic number plate recognition camera for vehicle speed measurement," in *European Conference on the Applications of Evolutionary Computation,* 2016.

[10] Dubská, M., Herout, A., Sochor,, "Automatic camera calibration for traffic understanding," *Proceedings of the British Machine Vision Conference,* vol. 4, p. 8, 2014.

[11] Lin, T.Y., Maire, M., Belongie, S., Hays, "Microsoft coco: common objects in context.," *Proceedings of the European Conference on Computer Vision,* pp. 740-755, 2014.

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y.,, "SSD: single shot multibox detector.," *Proceedings of the European Conference on ComputerVision,* pp. 21-37, 2016.

[13] Redmon, J., Divvala, S., Girshick, R., Farhadi,, "You only look once: unified, real-time object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 779-788, 2016.

[14] Ren, S., He, K., Girshick, R., Sun, "Faster R-CNN: towards realtime object detection with region proposal networks," *Advances in Neural Information Processing Systems,* pp. 90-98, 2015.

[15] Corral-Soto, E.R., Elder, J., "Slot cars: 3d modelling for improved visual traffic analytics," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* pp. 16-24, 2017.

[16] Sochor, J., Špaˇnhel, J.,Herout,A, "Boxcars: improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance," *IEEE Trans. Intell. Transp. Syst,* vol. 20, pp. 97-108, 2018.

[17] Zeng, R.,Ge, Z.,Denman, S., Sridharan, S., Fookes, "Geometryconstrained," in *arXiv,* 2019.

[18] Zapletal, D., Herout,, "Vehicle re-identification for automatic video traffic surveillance," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* pp. 25-31, 2016.

[19] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee, Ming-Hsuan Yang, "Object Contour Detection with a Fully Convolutional Encoder-Decoder Network," *Computer Vision and Pattern Recognition,* p. 320, 2016.

[20] S. H. R. A. R. D. I. D. R. A. Milan, "Online multi-target tracking using recurrent neural networks," *AAAI,* vol. 2, p. 4, 2017.

[21] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, "Spatially supervised recurrent convolutional neural networks for visual object tracking.," *In Circuits and Systems (ISCAS),* pp. 1-4, 2017.

[22] Sochor, J., Juránek, R., Herout, "Traffic surveillance camera calibration by 3dmodel bounding box alignment for accurate vehicle speed measurement.," *Comput. Vis. Image Underst.,* pp. 87-98, 2017.

[23] J. J. R. Š. J. L. Š. A. H. Sochor, "Comprehensive data set for automatic single camera visual speed measurement.," *IEEE Trans. Intell. Transp. Syst,* vol. 240, pp. 1633-1643, 2018.

[24] Luvizon, D.C., Nassu, B.T., Minetto, "A video-based system for vehicle speed measurement in urban roadways," IEEE Trans. Intell. Transp. Syst, 2017.

[25] Minetto, R., Thome, N., Cord, M., Leite, N.J., Stolfi,, "T-HOG: an effective gradient-based descriptor for single line text regions," *Pattern Recogn,* vol. 3, pp. 1078-1090, 2013.

[26] Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," *OpenCV Document, Intel, Microprocessor Research Labs,* vol. 1, 2000.

[27] Kocur, V, "Perspective transformation for accurate detection of 3d bounding boxes of vehicles in traffic surveillance," *Proceedings of the 24th Computer Vision Winter Workshop,* pp. 33-41, 2019.

[28] Tomasi, J. Shi and C., "Good features to track," *Proc. IEEE Conf. CVPR,* pp. 593-600, 1994.

[29] Sochor, J., Juránek, R., Herout, A., "Traffic surveillance camera calibration by 3dmodel bounding box alignment for accurate vehicle speed measurement.," *Comput. Vis. Image Underst,* pp. 87-98, 2017.