Accurate Automatic Camera Calibration on Low-Quality CCTV Traffic Video Streams

Audhav Durai
Thomas Jefferson High School for Science and Technology
Alexandria, United States
2023adurai@tjhsst.edu

Sankarshanaram Vempati
Thomas Jefferson High School for Science and Technology
Alexandria, United States
2024svempati@tjhsst.edu

Laurent Bindschaedler
Computer Science and AI Lab
Massachusetts Institute of Technology
Cambridge, United States
bindscha@csail.mit.edu

***Abstract*—This paper focuses on extracting vehicle trajectory from low-quality, uncalibrated traffic cameras via fully automatic camera calibration and homography transformations. We analyze video streams from 511 traffic surveillance cameras in varying areas and arbitrary relative locations and orientations to roads. We propose a novel automatic calibration methodology that is used to transform the video streams to a top-down perspective from a minute of the video stream. Using this perspective, we can extract both lane changes and vehicle speed. Our automatic camera calibration algorithm is inspired by two vanishing point based automatic camera calibration methods. We implement more accurate and robust methods for the detection of both vanishing points. We show that YOLOv4 combined with DeepSORT is 12.5 times faster than the leading vehicle detection model and achieves a 7.81% higher average precision and 6.04% higher mean average precision in low quality traffic surveillance cameras. We use this model for a more robust estimation of the first vanishing point and more robust object tracking. Furthermore, we propose a fast, novel "guess and check" algorithm for the detection of the second vanishing point to ensure accurate second vanishing point detection. We show that our camera calibration algorithm produces fewer inaccuracies than the state-of-the-art automatic camera calibration methodology in [1] through qualitative results.**

Keywords—camera calibration, traffic surveillance, computer vision, vanishing point detection

I. Introduction

Live traffic surveillance cameras are becoming increasingly more abundant [2]. This presents large opportunities for traffic-based computational applications through constant surveillance of such cameras. These applications include:

- Speed Limit: Estimation of vehicle speed from traffic video can be used by law enforcement for increased traffic safety [3].
- Accident Detection: The detection of traffic anomalies and collisions in traffic cameras can alert law enforcement when accidents occur [4], [5].
- Multi-Camera Fusion: Traffic Cameras that overlook different scenes in close areas can be fused together. Additionally, the same car can be tracked through multiple cameras [6].

Computationally analyzing these cameras presents its own set of challenges. Traffic cameras are mounted at arbitrary orientations and locations and possess parameters that are specific to each camera. These parameters are necessary in extracting useful vehicle measurements in video, such as speed, acceleration, and real-world position. Camera calibration is the process of extracting such parameters of a camera from its video and/or images. In this work, we focus on using camera calibration to extract both speed and vehicle trajectory from the video streams without manual input.

[7] is a review of the current camera calibration methods. It shows that many camera calibration methodologies require manual interference in the form of physical landmarks or measurements, such as annotated lane markings [8] or camera position [9], to aid in the extraction of its parameters. However, many of these measurements are challenging to find, especially over a large number of cameras. Automatic camera calibration is camera calibration with no manual involvement. However, most traffic cameras have a low resolution and frame rate. Even so, many current camera calibration algorithms rely on unrealistically high-quality traffic cameras [7].

In this paper, we focus on a fully automatic camera calibration methodology, which needs no user input or scene measurements, for vehicle speed and trajectory extraction in low-quality traffic video. This algorithm allows for various camera resolutions, frame rates, and positions (orientation and location) and calibrates in about a minute of video stream. The only assumption we make is that the roads do not contain any sharp turns, which is rare for highway roads.

Our methodology improves upon fully automatic traffic camera calibration algorithms that utilize the detection of two vanishing points, originally proposed in [10]. Since low-quality traffic camera data with camera calibration annotations are not currently available, we produce qualitative results of our camera calibration algorithm in comparison to a state-of-the-art algorithm on the same data. Our algorithm makes the following contributions:

- We show that a YOLOv4 model [11] combined with DeepSORT [12] performs better than the leading traffic surveillance detection model on realistic, low-quality video streams
- We improve on the detection of the first vanishing point (VP1) by proposing a more accurate and faster vehicle detection algorithm. We add vehicle tracking to extract the vehicle trajectory of each vehicle.

- We propose a guess and check second vanishing point (VP2) detection algorithm to prevent incorrect estimations and inaccuracies.

II. Related Works

A. Camera Calibration

An integral aspect of camera calibration is the detection of vanishing points. [14]-[20] rely on manual measurements on the road plane in some form to extract these vanishing points. However, algorithms to detect vanishing points without manual measurements also exist. One methodology for finding the vanishing points is by estimating the intersection of lane markings [21]-[23]. Another common methodology utilizes vehicle motion. [24] uses an activity map of the moving vehicles to detect the lane boundaries and finds the first vanishing point using the intersection of the lane boundaries. The second vanishing point is found by finding the intersection between lines in the direction of the bottom edge of each of the vehicles. However, they rely on manual measurements to determine the scene scale.

[1], [10] begin with the detection of two vanishing points, and unlike [24], is fully automatic. Vehicles are tracked and the motion of the vehicles is passed into a Diamond Space [25]. The Diamond Space transforms the motion of the vehicles into a Hough Space and takes the global maximum as the first vanishing point. The second vanishing point is detected using the Diamond Space as well. However, instead of using the motion of the vehicles, strong edges that are not in the direction of the first vanishing point are passed into the Diamond Space and the maximum is taken as the second vanishing point. Using these two values, the focal point and the third vanishing point are both calculated. 3D bounding boxes are created using the three vanishing points and the mean dimensions of the vehicles in the video streams are compared to statistical data of mean vehicle dimensions in the country to determine the scene scale. Though this calibration algorithm works on some video streams, it often results in incorrect vanishing point estimations that lead to distorted birds-eye view transformations.

[13] uses vanishing points to obtain a birds-eye view of the road plane. They detect vanishing points in the same manner as [1]. As shown in Figure 1, they use ground truth masks of the road and generate 4 lines, each starting at either the VP1 (dotted red) or VP2 (solid blue) and tangent to the road mask. The intersection points of these lines are used to find a perspective transformation matrix and transform the road to a top down view. We adopt this methodology and improve upon it by creating a more robust automatic camera calibration algorithm.

B. Vehicle Detection and Tracking

For the camera calibration algorithms that utilize the detection of vehicles for vanishing point estimation, their accuracy is highly dependent on the accuracy of their vehicle detection algorithms. Many of these algorithms opt for simple tracking algorithms: [10] utilizes a Kanade–Lucas–Tomasi feature tracker [26] and [13] relies on Kalman filters [27]. However, such simple tracking algorithms have shown to perform poorly in complex situations [28]. On the other hand, deep learning approaches have shown much more promise. Limited data exists on vehicle detection on traffic surveillance cameras. The most common dataset is the UA-DETRAC dataset (University at Albany DEtection and TRACking) [28], which provides over 10 hours of image sequences at 24 different locations with different camera angles, camera positions, and lighting conditions. [29] achieves the highest mean average precision (maP) score on the UADETRAC dataset. It utilizes a CenterNet architecture [30] to perform both vehicle detection and segmentation. However, it does not perform vehicle tracking, which is imperative for common computational traffic surveillance tasks, such as speed estimation and multicamera fusion. Although intersection over union (IOU) [31] based tracking algorithms can sometimes be used to circumvent this issue, they are ineffective on traffic surveillance videos with slow frame rates or frequent frame drops.

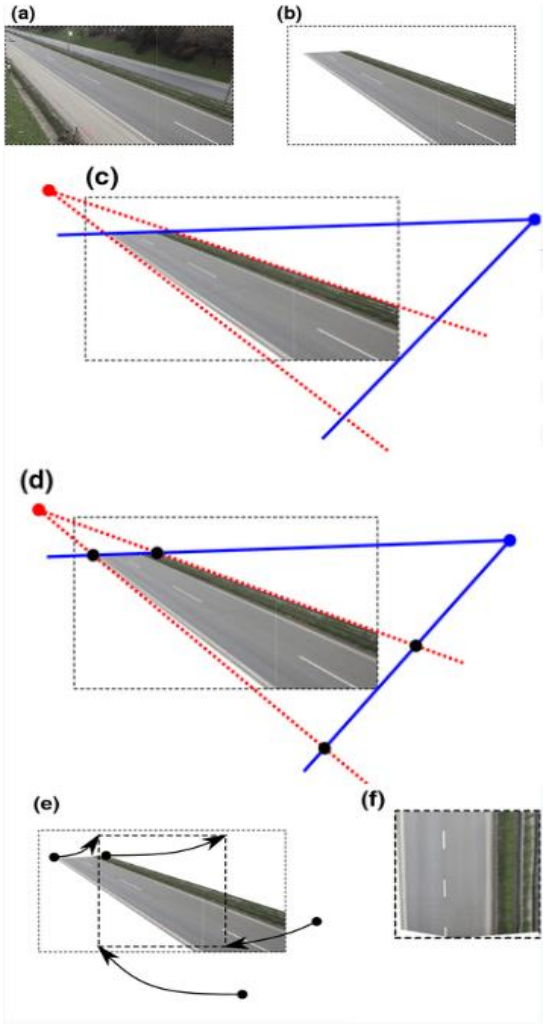

Fig. 1. The process of the construction of the perspective transformation using the first and second vanishing points [13]

III. Methodology

A. Vehicle Detection

Our vanishing point estimation algorithm relies on the detection of vehicles (see Section II-B and Section II-C). For the purpose of this study, we decide to focus on the UA-DETRAC dataset [28]. However the UA-DETRAC dataset is recorded at 25 frames per seconds (fps), with the JPEG image resolution of

960 × 540 pixels [28]. Such quality is unrealistic for the majority of traffic cameras, such as the 511 cameras used for this study [32], which are recorded at 15 frames per second with a resolution of 320 x 240 pixels [33] and include frequent frame drops. Thus, we lower the resolution, lower the frame rate, and incorporate frame drops in the UA-DETRAC dataset for our evaluation of the models. This provides a more realistic evaluation for how the model will perform on traffic surveillance cameras.

We choose to use a YOLOv4 [11] model with DeepSORT [12] to perform both vehicle detection and tracking. The object detection output from YOLOv4 is passed into the DeepSORT algorithm, which uses both Kalman filters and a re-identification model to assign IDs to each vehicle. Table I shows the evaluation of this model and the state-of-the-art SpotNet [29] model on our modified UA-DETRAC dataset.

TABLE I. COMPARISION OF OUR YOLOV4-DEEPSORT AND SPOTNET [29] ON OUR MODIFIED UA-DETRAC DATASET WITH AVERAGE PRECISION (AP), MEAN AVERAGE PRECISION (MAP), AND FRAMES PER SECOND

Model	AP	maP	FPS
SpotNet [29]	70.23%	50.34%	2 FPS
YOLOv4- DeepSORT	**78.04%**	**56.38%**	**25 FPS**

B. *First Vanishing Point*

We adopt the VP1 detection proposed by [10] and improve it by implementing our more accurate YOLOv4-DeepSORT vehicle tracking algorithm. We track cars using YOLOv4-DeepSORT and fit a line to the path of each distinct vehicle. As shown in Figure 2, the VP1 (filled blue circle) corresponds to the estimation of the intersection of all the vehicle paths. To estimate the intersection of the vehicle paths, we use the RANSAC [34] model proposed by [35] as it is robust against outlier paths, such as lane changes or video processing issues, that are not in the direction of the true VP1. The blue lines represent the vehicle paths extended to the VP1 while the red lines represent the detected vehicle paths.

Fig. 2. RANSAC algorithm used on vehicle paths to estimation VP1. VP1 represented with the intersection of blue lines

C. *Second Vanishing Point*

[10] shows that the VP2 is in the direction of many of the strong edges on the vehicles. Thus, we implement a probabilistic Hough line algorithm [36] on vehicles detected from our YOLOv4-DeepSORT model to find such edges, shown in Figure 3. Then, we pass these edges into a Diamond Space [25] that transforms all the detected edges on the road plane into a finite Hough space. [1], [10] take the global maximum in this space as the VP2. However, we find that, especially in low accuracy images, both the edge detection implemented by [1], [10] and the probabilistic Hough line are often too inaccurate to simply take the global maximum in the Diamond Space as the VP2.

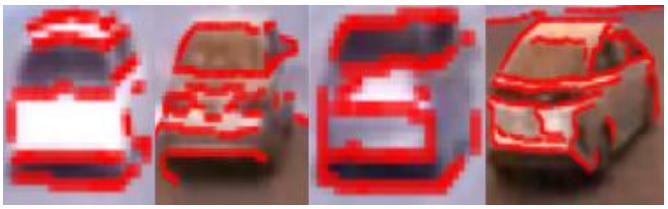

Fig. 3. Examples of edges from probabilistic Hough line algorithm on detected vehicles

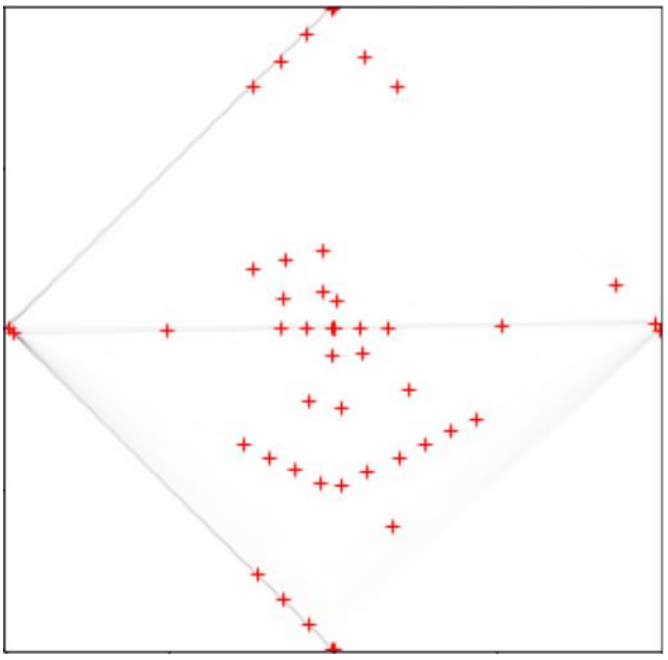

Fig. 4. Diamond space accumulator for VP2 estimation. Red crosses represent VP2 guesses in the diamond space

We propose a guess and check methodology for VP2 estimation. Since the VP1 estimation is very accurate, we hold that as constant. During the VP1 estimation, we also generate a mask of the road using the detected 2D vehicle bounding boxes. We then use the probabilistic Hough lines to pass in the vehicle edges into the Diamond Space, weighted by the strength of the edges. We find local maximums in the Diamond Space and use these as VP2 guesses; an example of the VP2 guesses in the Diamond Space is shown in Figure 4.

For each VP2 guess, we use the road mask and VP1 to generate a top down view of the road plane using the process summarized in Figure 1. A birds-eye view transformed video stream using the true VP2 will show the vehicles and lane lines in parallel and vertical, as they are in the real world scene, because the birds eye view transformation effectively transforms the (u, v) image pixel positions to 2D (x, y) real world coordinates. However, many of the VP2 guesses are incorrect and will produce distorted images after the birds-eye view transformation (example in Figure 5). Thus, for each perspective transformation, we find the standard deviation of the slopes of the fitted lines for the motion of each vehicle. We choose the VP2 guess that generates the smallest standard deviation to be the most accurate VP2. An example of a final, correct birds-eye view transformation using the VP2 estimation with our algorithm is shown in Figure 6, where the cars move in parallel.

Fig. 5. Example of VP2 guess that produces an incorrect perspective transformation

Fig. 6. Final perspective transformation with the chosen, correct VP2 estimation.

D. Traffic Information Extraction

After obtaining the final VP1 and VP2 estimation values, we compute the final birds-eye view transformation. We compare the average speed of the vehicles in the calibrated video stream to the average speed of vehicles on the road that the camera overlooks to obtain the scene scale.

IV. Results

There are limited camera calibration datasets on low quality traffic surveillance streams. Thus, we show the effectiveness of our automatic camera calibration algorithm by comparing it with the state-of-the-art algorithm presented in [1]. We take the detected vanishing points of each algorithm on the same video stream and compare the resulting birds-eye view images after applying the algorithm shown in Figure 1. As stated previously, accurate vanishing point detection should result in a birds eye view transformation with vehicles that are moving in parallel and parallel lane lines. Furthermore, lane widths should be constant as they are in the real world. We evaluate the algorithms on three low-quality, live traffic video streams from the Virginia Department of Transportation Database [32] and two high-quality traffic video streams from the original UA-DETRAC dataset.

Cameras a, b, and e are on standard 511 traffic surveillance streams [32] and cameras c and d are high quality streams from the UA-DETRAC dataset. We evaluate the camera calibration algorithms by comparing the transformed images to real world coordinates. As stated previously, this means that lanes lines and vehicles motion should be parallel as they are on the road plane. Our automatic camera calibration methodology is able to correctly estimate vanishing points in all five video streams, since all of the transformed images in the middle column have parallel lane lines and vehicle motion. On the other hand, the automatic camera calibration algorithm in [1] does not work on the lower quality 511 streams. We observe that the lane lines in the rightmost column of rows a, b, and e converge to a singular point, which is not how they are in the real world, showing that the camera calibration was ineffective. In contrast, both our calibration algorithm and the calibration algorithm in [1] perform well on the high quality UA-DETRAC stream, shown in row c and d. We conclude that our automatic camera calibration algorithm is more accurate and more robust to lower quality video streams than the state-of-the-art automatic camera calibration algorithm in [1], which is only accurate on high-quality streams, while our algorithm is effective on any quality of traffic video stream.

Our camera calibration output on row b shows that, while our algorithm does extract a correct birds-eye view perspective (with parallel lane lines), the transformed image does not cover the whole road. The limitation results from the mask generation as the vehicles that are far away from the camera go undetected. A road mask generation algorithm that does not require detection of vehicles would be more effective in this situation.

V. Conclusions

This paper presented a novel automatic camera calibration algorithm that increases the accuracy of the first and second vanishing point estimation in low quality traffic surveillance cameras. We implemented a YOLOv4 model combined with a DeepSORT model and show that it has a 7.81% higher average precision and 6.04% higher mean average precision than the leading vehicle detection model, SpotNet [29], on low-quality traffic surveillance video. We used this model to improve upon

the motion-based VP1 estimation introduced in [10] and also performed robust vehicle tracking using the DeepSORT re-identification model. We also introduced a novel ”guess and check” VP2 estimation algorithm that evaluated each VP2 guess and chooses the best estimate as the final VP2. We showed that our automatic camera calibration algorithm is more robust to low-quality video streams and produces significantly less inaccuracies than the state-of-the-art camera calibration algorithm in [1] through qualitative results on a variety of roads.

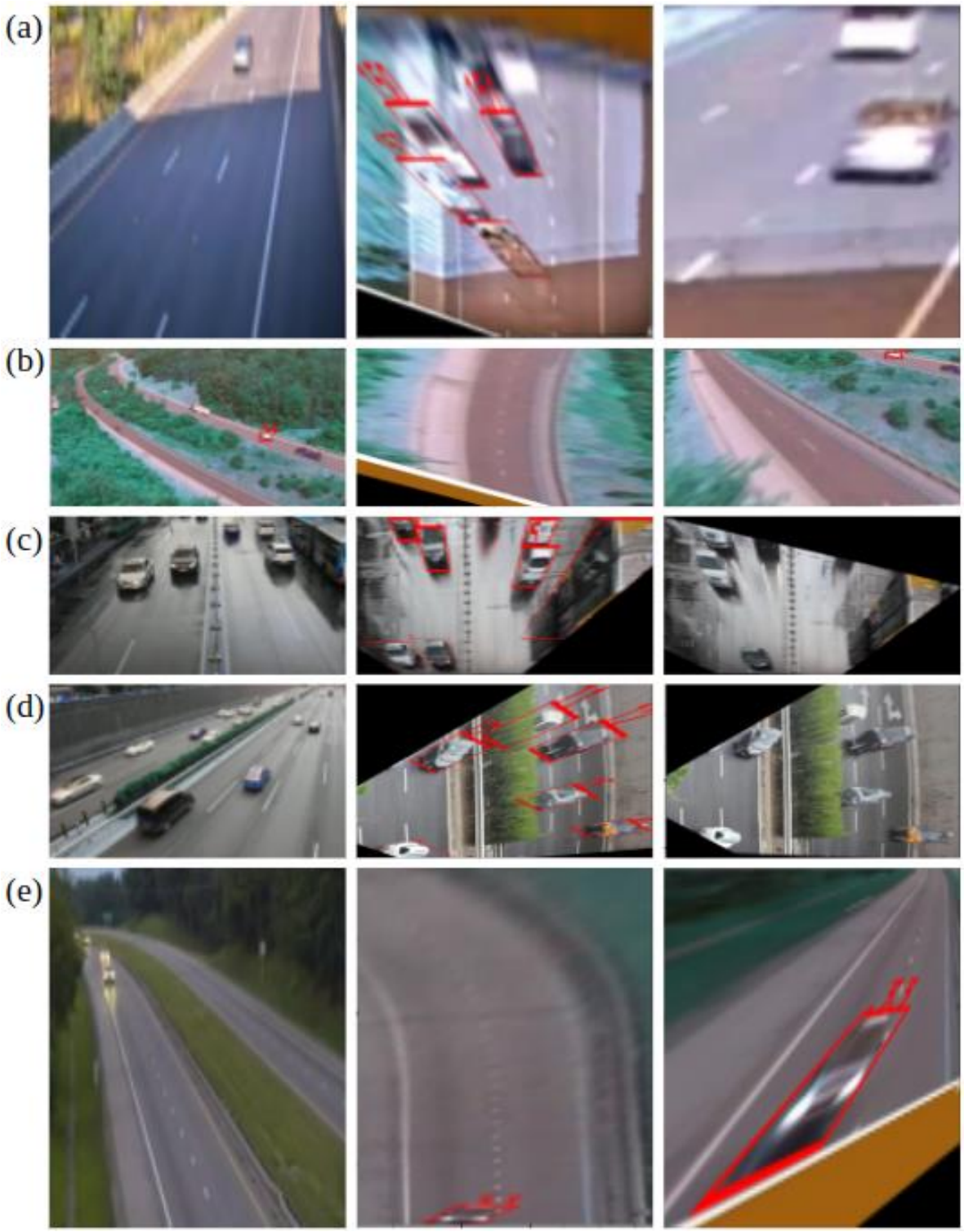

Fig. 7. Automatic camera calibration run on traffic surveillance video streams, varying in features. The left column is the original camera perspective. The middle column contains the results of our automatic camera calibration algorithm. The rightmost column has the results of the camera calibration algorithm in [1].

REFERENCES

[1] *Sochor, J., Juránek, R., Herout, A. (2017). Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement. Computer Vision and Image Understanding, 161, 87-98.*

[2] Kurdi, H. A. (2014). Review of closed circuit television (cctv) techniques for vehicles traffic management. International Journal of Computer Science Information Technology, 6(2), 199.

[3] Huang, T. (2018). Traffic speed estimation from surveillance video data. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 161-165).

[4] Vasu, L. (2010). An effective step to real-time implementation of accident detection system using image processing. Oklahoma State University.

[5] Ferreirinha, A. M. M. (2016). Fully automatic roadside video camera calibration and traffic analysis (Doctoral dissertation, Universidade de Coimbra)..

[6] Zhang, Z., Scanlon, A. W., Yin, W., Yu, L., Venetianer, P. L. (2009). Video Surveillance Using a Multi-Camera Tracking and Fusion System

[7] Sochor, J., Juránek, R., Špaňhel, J., Maršík, L., Široký, A., Herout, A., Zemčík, P. (2018). Comprehensive data set for automatic single camera visual speed measurement. IEEE Transactions on Intelligent Transportation Systems, 20(5), 1633-1643.

[8] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei-Yue Wang. Research on lane-marking line based camera calibration. In International Conference on Vehicular Electronics and Safety, ICVES, 2007. doi: 10.1109/ICVES.2007.4456361

[9] Tun-Wen Pai, Wen-Jung Juang, and Lee-Jyi Wang. An adaptive windowing prediction algorithm for vehicle speed estimation. In IEEE Intelligent Transportation Systems, 2001. doi: 10.1109/ITSC.2001.948780.

[10] M. Dubska, A. Herout, and J. Sochor. Automatic camera calibration for traffic understanding. In BMVC, volume 4, page 8, 2014. 1, 2

[11] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[12] Wojke, N., Bewley, A., Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.

[13] Kocur, V., Ftáčnik, M. (2020). Detection of 3D bounding boxes of vehicles using perspective transformation for accurate speed measurement. Machine Vision and Applications, 31(7), 1-15.

[14] C. Maduro, K. Batista, P. Peixoto, and J. Batista, "Estimation of vehicle velocity and traffic intensity using rectified images," in Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on, Oct 2008, pp. 777–780.

[15] Nurhadiyatna A, Hardjono B, Wibisono A, Sina I, Jatmiko W, Ma'sum M, Mursanto P. Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. In: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on. 2013. p. 451–6.

[16] Sina I, Wibisono A, Nurhadiyatna A, Hardjono B, Jatmiko W, Mursanto P. Vehicle counting and speed measurement using headlight detection. In: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on. 2013. p. 149–54.

[17] Luvizon D, Nassu B, Minetto R. Vehicle speed estimation by license plate detection and tracking. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. 2014. p. 6563–7.

[18] Luvizon DC, Nassu BT, Minetto R. A video-based system for vehicle speed measurement in urban roadways. IEEE Transactions on Intelligent Transportation Systems 2016, PP(99):1–12.

[19] Do VH, Nghiem LH, Thi NP, Ngoc NP. A simple camera calibration method for vehicle velocity estimation. In: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on. 2015. p. 1–5

[20] Lan J, Li J, Hu G, Ran B, Wang L. Vehicle speed measurement based on gray constraint optical flow algorithm. Optik - International Journal for Light and Electron Optics 2014;125(1):289 –95.

[21] Cathey, F. W., Dailey, D. J. (2005, June). A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In IEEE Proceedings. Intelligent Vehicles Symposium, 2005. (pp. 777-782). IEEE.

[22] He, X. C., Yung, N. H. (2007, February). A novel algorithm for estimating vehicle speed from two consecutive images. In 2007 IEEE Workshop on Applications of Computer Vision (WACV'07) (pp. 12-12). IEEE.

[23] Grammatikopoulos, L., Karras, G., Petsa, E. (2005, November). Automatic estimation of vehicle speed from uncalibrated video sequences. In Proceedings of International Symposium on Modern Technologies, Education and Profeesional Practice in Geodesy and Related Fields (pp. 332-338).

[24] T. Schoepflin and D. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," Intelligent Transportation Systems, IEEE Transactions on, vol. 4, no. 2, pp. 90–98, June 2003

[25] M. Dubska and A. Herout, "Real projective plane mapping for detection of orthogonal vanishing points," in Proceedings of BMVC 2013. The British Machine Vision Association and Society for Pattern Recognition, 2013, pp. 1–10

[26] Suhr, J. K. (2009). Kanade-lucas-tomasi (klt) feature tracker. Computer Vision (EEE6503), 9-18

[27] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[28] Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M. C., Qi, H., ... Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding, 193, 102907.

[29] Perreault, H., Bilodeau, G. A., Saunier, N., Héritier, M. (2020, May). Spotnet: Self-attention multi-task network for object detection. In 2020 17th Conference on Computer and Robot Vision (CRV) (pp. 230-237). IEEE.

[30] X. Zhou, D. Wang, and P. Krahenb¨uhl, "Objects as points," arXiv preprint arXiv:1904.07850, 2019.

[31] Bochinski, E., Senst, T., Sikora, T. (2018, November). Extending IOU based multi-object tracking by visual information. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.

[32] Virginia Department of Transportation (2012). Virginia 511 Web. https://www.511virginia.org/

[33] Virginia Department of Transportation (2012). VDOT's New 511 Traffic Information Program. https://www.virginiadot.org/news/resources/511_video/VDOT_-_Media_Day_Info_Session.pdf

[34] Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. Image Rochester NY, 4(1), 2-3.

[35] Chaudhury, K., DiVerdi, S., Ioffe, S. (2014, October). Autorectification of user photos. In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 3479-3483). IEEE.

[36] Matas, J., Galambos, C., Kittler, J. (2000). Robust detection of lines using the progressive probabilistic hough transform. Computer vision and image understanding, 78(1), 119-137.