Proc. of the 9th Student Computing Research Symposium (SCORES'23), Koper, Slovenia, October 5, 2023

Real-Time Vehicle Speed Estimation from Video

Mitko Nikov
mitko.nikov@student.um.si
Faculty of Electrical Engineering and Computer Science, University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

Mitja Žalik
mitja.zalik@um.si
Faculty of Electrical Engineering and Computer Science, University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

Domen Mongus
domen.mongus@um.si
Faculty of Electrical Engineering and Computer Science, University of Maribor
Koroška cesta 46
SI-2000 Maribor, Slovenia

ABSTRACT

Vehicle tracking and speed estimation are essential in many intelligent traffic management systems. In this paper, we propose a new real-time architecture for estimating the speed of the vehicles from monocular video streams. While following traditional steps, including vehicle detection, its tracking, and mapping of coordinates from pixel to real-world space, notable progress beyond state-of-the-art is achieved by introducing a new filtering schema with a camera calibration that allows for speed estimations from arbitrary viewing angles. This enables monitoring to be performed from moving platforms, such as Unmanned Aerial Vehicle (UAV), while also observing multiple vehicles at once. As confirmed by the results, even in these circumstances, the proposed approach achieves comparable results to state-of-the-art techniques applied on stationary video streams.

KEYWORDS

vehicle speed estimation, video stream processing, unmanned aerial vehicle, traffic management systems

1 INTRODUCTION

With the increase in traffic density, real-time vehicle speed estimation is actively used in traffic-aware route planning and forecasting [11], speed limit enforcement [13] and other intelligent traffic management systems' services. Traditionally, methods used for vehicle speed estimation from video streams, adopt a so-called tracking-by-detection framework [20], that consists of vehicle detection, vehicle tracking and speed estimation steps [14, 15, 19, 25, 26].

In general, vehicles are detected and annotated with bounding boxes on a frame-by-frame basis. In pipelines of older work [8, 27, 29] this was achieved using histograms, thresholds, and other means of separating moving objects from static backgrounds. Accordingly, these methods are susceptible to noise and need extensive manual calibrations for specific environments and setups. Nevertheless, deep learning approaches proved to be efficient in dealing with these issues, and state-of-the-art today rely almost exclusively on variations of Convolutional Neural Networks (CNNs) such as YOLO[1] [23], Faster-RCNN[2] [7] and Mask-RCNN[2] [10] for detection as well as classification of vehicles [9, 15, 18, 25, 26, 30].

Speed estimation, however, requires vehicle tracking over a set of successive frames even in those cases when occlusions occur. Predicting their movements is, therefore, inevitably required for successfully associating their detected positions. The Simple Online Realtime Tracking (SORT) algorithm [2] and different variations of the Kalman filter are commonly used for this purpose [20]. In order to enhance vehicle matching robustness across frames, the recently introduced DeepSORT further extends the SORT algorithm by extracting vehicle features using a pre-trained CNN [18, 28], while increased accuracy of vehicle tracking and the extraction of movement parameters using optical flow was also examined [4, 14, 19, 22]. In all of these cases, however, movement vectors are estimated in pixel-space. Their mapping into real-world coordinates is, therefore, addressed next.

Mapping of movement vectors inevitably requires camera calibration that establishes a correlation between the pixel and the real-world displacement of the vehicle. In most cases, calibration approaches are based either on manually defined coefficients [14, 19, 22, 24, 27, 29] and intrusion lines [16] or linear image transformations [15, 18]. However, these approaches require extensive environment-specific parameter calibrations and measurements, due to factors such as lighting conditions, camera angles, occlusions or car-specific parameters [19, 29]. Therefore, they often remain limited to the video-streams from stationary cameras. On the other hand, some recently introduced approaches address these issues by using CNNs and optical flow to estimate the average speed of traffic rather than individual vehicles. [5, 31]

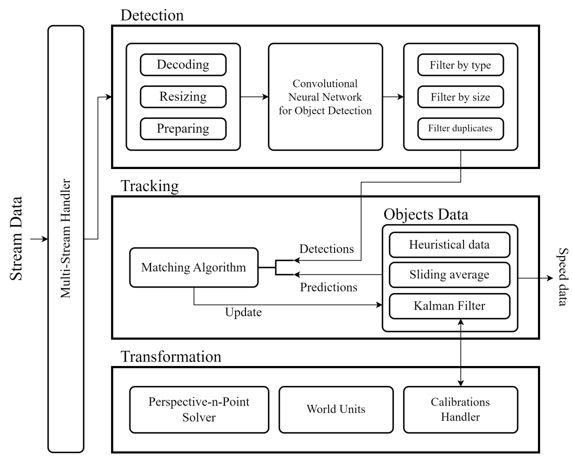


Figure 1: High-level architecture of the proposed vehicle speed estimation method.

Accordingly, while vehicle detection, tracking and even average speed estimation of traffic are already well-researched, estimating

[1]YOLO: You Only Look Once
[2]RCNN: Region-based Convolutional Neural Network

DOI: https://doi.org/10.26493/scores23.14

speed of individual vehicles poses a greater challenge. The non-linear effects stemming from the perspective projection of real-world objects onto the 2D image space, coupled with the intricate terrain configurations, further aggravates the problem, preventing applications of the discussed methods for processing video-streams from moving platforms, such as Unmanned Aerial Vehicle (UAV).

In order to address these issues, we propose a new camera calibration technique that is based on an iterative solver for the Perspective-n-Point (PnP) problem [6], together with several improvements made upon the known movement vector estimation with ray tracing and Kalman filtering [17]. Details on the proposed method are explained in the next section. Section 3 provides the achieved results, while conclusions are presented in section 4.

2 METHOD

Following the pattern of tracking-by-detection frameworks, the processing pipeline of the proposed method consists of detection, tracking and space transformation steps, as shown in Figure 1.

2.1 Vehicle Detection

In the detection step, the stream data is first decoded, resized and preprocessed before being fed into the CNN for object detection. Processing the image in the CNN is the most computationally intensive task of the proposed pipeline. For the real-time object detection, we obtained a YOLOv4 model [3], pretrained on the ImageNet and MS COCO datasets. Since the mentioned datasets contain mostly images of different objects from the ground level, the pretrained model performed badly in our high elevation and high meter-to-pixel domain, where objects are captured from above and appear relatively small. Therefore, we improved the model via transfer learning using custom created dataset. The training dataset consists of approximately 61500 images and 550000 annotated vehicles. Vehicle annotation was performed manually over 6 month period and aided by general object tracking [1, 12, 21] in order to obtain multiple samples of vehicles from different angles. The model reached convergence after being trained on 35000 batches of 64 images in order to fine tune its weights and, thus, transfer the knowledge to the required domain.

Further improvements on object detection are achieved by applying several filters for removal of duplicate objects and objects that are irrelevant for speed estimation because of their class or size. Still, this does not solve the commonly present shift of bounding-boxes over consecutive frames, as indicated by Hua et al. [14]. Nevertheless, the integration of the Kalman filter substantially mitigates this error during vehicle tracking, as discussed.

2.2 Vehicle Tracking

The proposed implementation of tracking is based on the SORT algorithm [2] for Multi-Object Tracking (MOT). Given the information about prior movement of objects $[\dot{x}, \dot{y}]^T$, the SORT algorithm uses a Kalman filter with the following state vector:

$$\mathbf{x} = [x, y, s, r, \dot{x}, \dot{y}, \dot{s}]^T, \tag{1}$$

where the point (x, y) is the center of the bounding box, s is the area and r - the aspect ratio of the bounding box. Moreover, the displacement vector $[\dot{x}, \dot{y}]^T$ in 2D space can be directly extracted from the state. However, the perspective projection of the 3D world makes the problem of object tracking in the 2D image space non-linear. Therefore, with the use of the mapping function R that projects image coordinates onto a real-world plane, formally defined in section 2.3, we are able to model the problem directly in 3D space with the following Kalman filter state:

$$\mathbf{x}_{\text{ours}} = \left[R_0(c)_x, R_0(c)_y, s, r, \dot{R_0}(c)_x, \dot{R_0}(c)_y, \dot{s}\right]^T, \tag{2}$$

where c is a corner of the bounding box whose projection on the ground plane is the closest to the camera. The vehicle velocity is directly described by the vector $v = \left[\dot{R_0}(c)_x, \dot{R_0}(c)_y\right]^T$. Moreover, the inclusion of the z coordinate from the 3D domain in the Kalman filter state is unnecessary and will always be 0 (ground plane).

Following the prediction of the Kalman filter and re-projection of the object's position in the current frame, we proceed to match the predicted bounding box with a bounding box obtained by object detection, prioritizing the highest Intersection over Union (IOU) metric. We also impose a IOU_{min} threshold that restricts the matching algorithm from linking bounding boxes that have a really low IOU correspondence. Furthermore, we implement an object age heuristic, which assigns priority to older objects in cases where a single bounding box exhibits similar Intersection over Union (IOU) correspondence with two or more tracked objects.

2.3 Space Transformations

As already mentioned in the previous section, space transformation from the 2D image space to the 3D world captured by the camera, is necessary for tracking as well as the actual vehicle speed estimation. Additionally, to project the ground plane and bounding box predictions from 3D to 2D space, a reverse transformation is also required.

We propose an effective methodology that requires only 6 pairs of image-to-world correlation coordinates. With the use of PnP iterative solver, we manually provide 4 image-to-world correlation coordinate pairs on the ground plane ($z = 0$) and 2 more on the $z = 1$ plane. This results in translation and rotation vectors describing the relative 3D space. We use landmarks with approximately equal length, width and height to ensure uniformity across all units in the 3D space (x, y and z). Although subpixel precision is required for extremely precise calibration, we found the Kalman filter to be sufficiently flexible, delivering comparable results even in its absence.

Using the camera matrix, translation and rotation vectors provided by our camera calibration, it is possible to define a function $R_z : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that maps the 2D image space to some predefined z plane in the 3D space. It is implemented using a ray-plane intersection. By defining the R function in regards to a fixed z plane, we can show that R_z is a bijectional mapping from the 2D image space to the predefined plane in the 3D world. Let us assume that all vehicles are driving on the ground plane where $z = 0$ and thus only be interested in R_0.

Nevertheless, establishing a correlation between the 2D and 3D space remains incomplete until the equivalence of one unit in our 3D space to a specific distance in meters is determined. Using our forementioned functions, we can introduce a routine to calibrate the "world units coefficient" as follows: pick two points p_1 and p_2

in the 2D image space for which, you know the real-world distance between their projections in the $z = 0$ plane. The 3D world unit coefficient is the ratio of the known distance between the two points in the real world and the Euclidean distance of the projected points $R_0(p_1)$ and $R_0(p_2)$.

At last, using time data and the world unit coefficient, we can convert the mentioned velocity vector $v = \left[\dot{R_0}(c)_x, \dot{R_0}(c)_y\right]^T$ to a speed measurement.

3 RESULTS

This research is a part of the development of the GeMMa Fusion Machine Learning (GFML) application written in C++, where all of our tests were also conducted. For the given results, we evaluated the pipeline on Intel Core i7-9700K @ 3.6GHz with 32GB RAM and NVIDIA GeForce RTX2070 SUPER GPU.

3.1 Test Data

(a) (b) (c)

Figure 3: Test (a) Stream 1, (b) Stream 2, and (c) Stream 3 with estimated vehicle speed.

A DJI Mini 3 Pro and a DJI Phantom 4 Advanced drones were used to record most of our test footage. The footage was initially recorded in 4K resolution, with 25 frames per second (FPS) D-Cinelike 8-bit color profile, then resized for comparison to 1920 × 1080 px and 720 × 576 px resolution without applying a color lookup table (LUT). The ground truth data is calculated using a Global Positioning System (GPS) measurement device with 30 connected GPS satellites and synced to the footage using time codes.

Amongst these, three different scenes were selected for testing. Stream 1 (shown in Fig. 3a) monitors vehicle during acceleration, Stream 2 (shown in Fig. 3b) monitors vehicle during driving with constant speed, while Stream 3 (shown in Fig. 3c) monitors individual vehicle in heavy traffic.

3.2 Time complexity

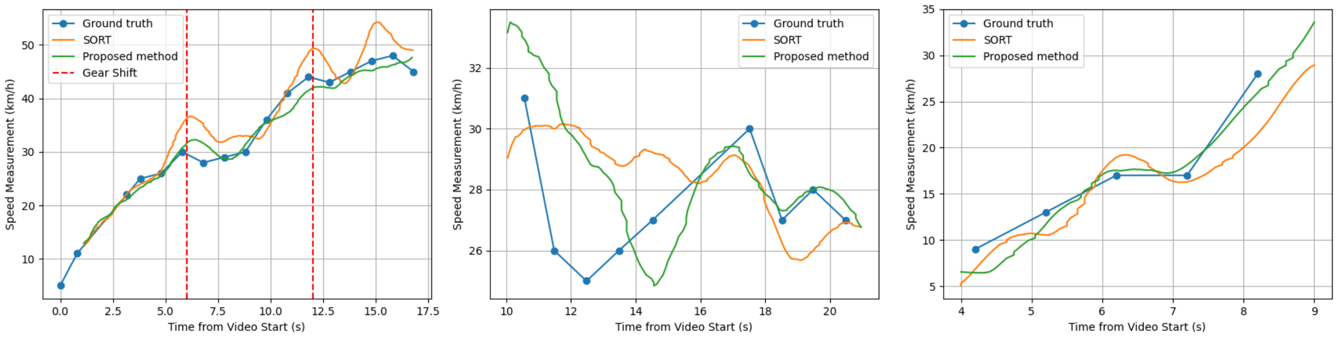
With our custom dataset, a pipeline integrating the YOLOv4 model, as well as its reduced version YOLOv4-tiny, was examined. As presented in Table 1, we achieved speed estimation of 11-18 frames per second (FPS). Most of the computing time is spent in the YOLO CNN, requiring 20-50 milliseconds, as seen in the OD column in Table 1, depending on the model and the input image size. There were no significant time differences with and without the use of our modified Kalman state.

Table 1: FPS rates and object detection times (OD) achieved on tested video streams with 1920 × 1080 px and 720 × 576 px resolutions and two different CNNs, namelly YOLOv4 and YOLOv4-tiny.

Stream size [px]	CNN	Stream 1		Stream 2		Stream 3	
		FPS	OD [ms]	FPS	OD [ms]	FPS	OD [ms]
1920× 1080	YOLO v4	9	28	9	29	9	33
	YOLO v4-tiny	11	26	11	26	10.5	27
720× 576	YOLO v4	11	22	11	22	14	47
	YOLO v4-tiny	18	36	17	33	17	33

3.3 Accuracy

In Figure 2a, we can observe a sample measurement of a vehicle acceleration from aerial footage in Stream 1, in order to compare the initial SORT state [2] of the Kalman filter with our modified 3D-based state. The accuracy of the ground truth and measurement data, is further corroborated by the fact that one can notice the two initial gear shifts. A second sample of measurement of a very



(a) (b) (c)

Figure 2: Comparison of the results achieved using SORT and the proposed method with ground truth data on (a) Stream 1, (b) Stream 2, and (c) Stream 3.

small speed variation across a 10 second period can be seen on Figure 2b. When the SORT's 2D-based Kalman state is used, because of its linear nature, we found that it will quite often overshoot, especially when the camera's elevation angle is relatively small. Nevertheless, we can see that even with imprecise camera and world unit calibration, one can achieve comparable results to the ground truth. Furthermore, as shown by Fig. 2c, comparable results are achieved also when observing an individual vehicle within a heavy traffic. It should, however, be noted that inaccuracies in the GPS and time data, camera and world unit calibration and bounding box detections are unavoidable.

4 CONCLUSION

In this paper, we presented our pipeline for vision-based vehicle speed estimation with many improvements of the algorithms used, both in object tracking and space transformations. We can conclude that camera and world unit calibration combined with modifications of the Kalman state, can yield results comparable to state-of-the-art custom space transformation methodologies. We note however, that due to the specificity of the problem, the lack of proper datasets and the amount of external factors involved, fully testing and comparing our method to closed-source state-of-the-art is practically impossible.

In the future, we also plan to work on adaptive camera calibration for airborne vehicles with KLV gyroscopic sensor streams and fully automatic intrinsic and extrinsic parameter camera calibration using convolutional neural networks.

ACKNOWLEDGMENTS

This work was supported by DARS d.d. as a part of the project "System for short and long-term traffic prediction with the use of artificial intelligence" and funded by Slovenian Research Agency grant number P2-0041 and L7-2633.

REFERENCES

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. 2009. Visual tracking with online Multiple Instance Learning. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 983–990. https://doi.org/10.1109/CVPR.2009.5206737

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple Online and Realtime Tracking. *arXiv* (Feb. 2016). https://doi.org/10.1109/ICIP.2016.7533003 arXiv:1602.00763

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* (April 2020). https://doi.org/10.48550/arXiv.2004.10934 arXiv:2004.10934

[4] Sedat Doğan, Mahir Serhan Temiz, and Sıtkı Külür. 2010. Real Time Speed Estimation of Moving Vehicles from Side View Images from an Uncalibrated Video Camera. *Sensors (Basel, Switzerland)* 10, 5 (May 2010), 4805–24. https://doi.org/10.3390/s100504805

[5] Huanan Dong, Ming Wen, and Zhouwang Yang. 2019. Vehicle Speed Estimation Based on 3D ConvNets and Non-Local Blocks. *Future Internet* 11, 6 (May 2019), 123. https://doi.org/10.3390/fi11060123

[6] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (June 1981), 381–395. https://doi.org/10.1145/358669.358692

[7] Ross Girshick. 2015. Fast R-CNN. , 1440–1448 pages.

[8] Lazaros Grammatikopoulos, George Karras, and Elli Petsa. 2005. Automatic estimation of vehicle speed from uncalibrated video sequences. *ResearchGate* (Jan. 2005).

[9] Alexander Grents, Vitalii Varkentin, and Nikolay Goryaev. 2020. Determining vehicle speed based on video using convolutional neural network. *Transp. Res. Procedia* 50 (Jan. 2020), 192–200. https://doi.org/10.1016/j.trpro.2020.10.024

[10] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask R-CNN. , 2961–2969 pages.

[11] Zongjian He, Jiannong Cao, and Tao Li. 2012. MICE: A Real-time Traffic Estimation Based Vehicular Path Planning Solution Using VANETs. In *2012 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 172–178. https://doi.org/10.1109/ICCVE.2012.39

[12] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2012. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *Computer Vision – ECCV 2012*. Springer, Berlin, Germany, 702–715. https://doi.org/10.1007/978-3-642-33765-9_50

[13] Stephane Hess. 2004. Analysis of the Effects of Speed Limit Enforcement Cameras: Differentiation by Road Type and Catchment Area. *Transp. Res. Rec.* 1865, 1 (Jan. 2004), 28–34. https://doi.org/10.3141/1865-05

[14] Shuai Hua, Manika Kapoor, and David C. Anastasiu. 2018. Vehicle Tracking and Speed Estimation from Traffic Videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 153–1537. https://doi.org/10.1109/CVPRW.2018.00028

[15] Tingting Huang. 2018. Traffic Speed Estimation From Surveillance Video Data. , 161–165 pages.

[16] Saleh Javadi, Mattias Dahl, and Mats I. Pettersson. 2019. Vehicle speed measurement model for video-based systems. *Comput. Electr. Eng.* 76 (June 2019), 238–248. https://doi.org/10.1016/j.compeleceng.2019.04.001

[17] Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82, Series D (1960), 35–45.

[18] Amit Kumar, Pirazh Khorramshahi, Wei-An Lin, Prithviraj Dhar, Jun-Cheng Chen, and Rama Chellappa. 2018. A Semi-Automatic 2D Solution for Vehicle Speed Estimation From Monocular Videos. , 137–144 pages.

[19] Jing Li, Shuo Chen, Fangbing Zhang, Erkang Li, Tao Yang, and Zhaoyang Lu. 2019. An Adaptive Framework for Multi-Vehicle Ground Speed Estimation in Airborne Videos. *Remote Sens.* 11, 10 (May 2019), 1241. https://doi.org/10.3390/rs11101241

[20] David Fernández Llorca, Antonio Hernández Martínez, and Iván García Daza. 2021. Vision-based vehicle speed estimation: A survey. *IET Intel. Transport Syst.* 15, 8 (Aug. 2021), 987–1005. https://doi.org/10.1049/itr2.12079

[21] Alan Lukežič, Tomáš Vojíř, Luka Čehovin Zajc, Jiří Matas, and Matej Kristan. 2018. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *Int. J. Comput. Vision* 126, 7 (July 2018), 671–688. https://doi.org/10.1007/s11263-017-1061-3

[22] Diogo Carbonera Luvizon, Bogdan Tomoyuki Nassu, and Rodrigo Minetto. 2016. A Video-Based System for Vehicle Speed Measurement in Urban Roadways. *IEEE Trans. Intell. Transp. Syst.* 18, 6 (Sept. 2016), 1393–1404. https://doi.org/10.1109/TITS.2016.2606369

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2015. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* (June 2015). https://doi.org/10.48550/arXiv.1506.02640 arXiv:1506.02640

[24] Divya Sharma, Shilpa Sharma, and Vaibhav Bhatnagar. 2022. Automated Vehicle speed Estimation and License Plate Detection for Smart Cities Development. In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*. IEEE, 378–383. https://doi.org/10.1109/AIC55036.2022.9848890

[25] Zheng Tang, Gaoang Wang, Tao Liu, Young-Gun Lee, and Jenq-Neng Hwang. 2017. Multiple-Kernel Based Vehicle Tracking Using 3D Deformable Model and Camera Self-Calibration. *ResearchGate* (Aug. 2017).

[26] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. 2018. Single-Camera and Inter-Camera Vehicle Tracking and 3D Speed Estimation Based on Fusion of Visual and Semantic Features. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 108–1087. https://doi.org/10.1109/CVPRW.2018.00022

[27] Jin-xiang Wang. 2016. Research of vehicle speed detection algorithm in video surveillance. In *2016 International Conference on Audio, Language and Image Processing (ICALIP)*. IEEE, 349–352. https://doi.org/10.1109/ICALIP.2016.7846573

[28] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv* (March 2017). https://doi.org/10.48550/arXiv.1703.07402 arXiv:1703.07402

[29] Jianping Wu, Zhaobin Liu, Jinxiang Li, Caidong Gu, Maoxin Si, and Fangyong Tan. 2009. An algorithm for automatic vehicle speed detection using video camera. In *2009 4th International Conference on Computer Science & Education*. IEEE, 193–196. https://doi.org/10.1109/ICCSE.2009.5228496

[30] Wencheng Wu, Vladimir Kozitsky, Martin E. Hoover, Robert Loce, and D. M. Todd Jackson. 2015. Vehicle speed estimation using a monocular camera. In *Proceedings Volume 9407, Video Surveillance and Transportation Imaging Applications 2015*. Vol. 9407. SPIE, 17–30. https://doi.org/10.1117/12.2083394

[31] Xiaodong Yu, Ping Xue, Lingyu Duan, and Qi Tian. 2007. An algorithm to estimate mean vehicle speed from MPEG Skycam video. *Multimed. Tools Appl.* 34, 1 (July 2007), 85–105. https://doi.org/10.1007/s11042-006-0073-8