VIDEO SURVEILLANCE FOR ROAD TRAFFIC MONITORING

Marc Grau *Ignasi Mas* *Hugo Prol* *Jordi Puyoles*

Autonomous University of Barcelona

{marc.grau, ignasi.masm, hugo.prol, jordi.puyoles}@e-campus.uab.cat

ABSTRACT

During the las decades Road Traffic Monitoring has played a crucial role not only for collecting statistics, but also for managing traffic in real time. Technological devices have allowed people to be informed about the most suitable itineraries in order to avoid traffic jams, maintenance works and other inconveniences easing driving and improving road safety. The present document details a system capable to estimate vehicles speed and evaluate road density in highway using raw video obtained from a standard traffic camera. The project has been developed during March and April, 2018 at the UPC (Universitat Politècnica de Catalunya).

1. MOTIVATION

Road accidents are one of the main causes of death all over the world and the first in people between 18 and 24 years old in the European Union. It's not surprising that governments and associations spend lots of efforts in awareness campaigns and technological improvements to reduce accidents and improve traffic rules. The aim of the system detailed below is to estimate vehicles speed and evaluate road density, using basic computer vision techniques, to increase road safety and avoid traffic jams. The document is structured based on the different stages of the pipeline as follows: Firstly, a brief introduction to related work is considered and data collection is detailed analyzing image characteristics (sections 2 & 3). Then stabilization, in 4.1, is approached in order to reduce the jittering effect produced by the camera. Further stages (4.2, 4.3 & 4.4) are focused on classifying vehicle pixels by means of adaptive background subtraction, remove shadows and apply morphological operations to get uniform objects shape. Later steps in the system are related to tracking the interest objects (subsec. 4.5) and finally estimate the vehicles speed by means of image rectification as detailed in 4.6. The last sections 5 and 6 concern evaluation of the system and a summary of the benefits and weaknesses of our implementation.

2. RELATED WORK

Radar sensors have widely been used in the last years for traffic monitoring [1]. Their limitation arises when analyzing drivers behavior is needed. Newer approaches try to exploit the increase of computational power and analyze more complex features, as for example images taken from cameras rather than unidimensional signals. Methods relying tracking with kalman filters [2] and classical neural networks [3] appeared some years ago, and with the break up of deep neural networks, systems with more capabilities made their appearance [4]. However, these approach rely on having huge datasets and computational power, which make them quite hard to implement. Our method is focused on less demanding algorithms, yet demonstrating remarkable performance, even with the presence of hard shadows, which is a breaking point in comparison with other less demanding methods.

3. DATA COLLECTION

We have worked over a custom sequence recorded March 23rd, 2018 at 16:00 in Mollet del Vallès (coordinates 41.544427, 2.224059).

This scene contains a freeway in both directions, but we need just one for our application. We have considered the direction incoming towards the camera. In particular, we'll focus on the part with a straight trajectory. This direction has two lanes and contains fragments when cars are aligned, so we can evaluate the tracking of cars with parallel trajectories. It contains both straight and dashed road lines. Hard shadows do appear in the scene due to the weather at the time it was recorded. Those shadows have a bigger size than the space between lanes. Therefore, we have to avoid the issue of different cars merging when they are parallel. On the other hand, the scene suffers from some jitter. That could give fake motion, what could lead to false positives in background subtraction.

The original video was recorded at 30 fps and at a resolution of 1920×1080 pixels. Additionally, we applied a sub-sampling of $\frac{1}{6}$ in space, since that would let us work with a resolution similar to the training datasets of Highway and Traffic from ChangeDetection.net dataset [5]. Furthermore, that rescaling avoids that license plates can be seen, so after

that step vehicles are already anonymous. On the other hand, we applied a frame rate subsamplig of $\frac{1}{2}$. Values between it and 1 still keep a natural flow. Otherwise, it would be hard for our tracking algorithm due to the information loss.

4. METHOD

4.1. Stabilization

One of the issues that the dataset presents is the fact that it suffers from jitter. This jitter produces some problems in further stages. The most relevant one is the noise produced on the road granulation, causing a high amount of false positives at the background subtraction, what makes the scene harder to filter in the following stages.

In order to help the rest of the pipeline, a video stabilization step is applied at this point so the points in the real scene remain at the same pixel, so we reduce considerably the noise since the only irregularities remaining are introduced by camera noise or small changes in luminance.

This stabilization step is applied with optical flow through block matching. We have used forward compensation, in which each frame is divided into blocks and then for each block it is found the most correlated patch on the following frame. This way we could ensure that all blocks from the current frame have a motion vector, something which is not accomplished using backward compensation. To do so we have used blocks of 16×16 pixels. Then, from this motion vectors we have computed the overall displacement of the image as the mean of the motion vectors of all blocks. To compensate this translation, we have applied to each frame a similarity transform to translate it with the same value and contrary sign as its motion.

4.2. Background Subtraction

The goal of this stage is to detect the pixels that belong to the foreground. We face the problem that background subtraction models need a ground truth to be trained, which we do not have for the sequences. We can overcome it by using the model and parameters trained on other datasets as a starting point.

Therefore, an adaptive model is applied, since we have validated it against a Gaussian non-adaptive model with the training datasets of Highway and Traffic and it gave higher values of F1 score and Area Under the Curve. Both models compute the probability distribution of each pixel in the training set (assuming a Gaussian behavior) and then at prediction it validates if each of them in a given frame is inside the limits of this Gaussian distribution (if that is the case it considers it as background). The main difference between them is that non-adaptive model once is trained keeps its values (mean and standard deviation) while the adaptive model updates them at each frame. For the parameters α y ρ we have taken as a reference the best ones on the same training datasets and then modified them to fit for the new sequences according to qualitative results, resulting in values for α and ρ of 4 and 0.15 respectively.

4.3. Shadow Removal

Due to the presence of hard shadows in the dataset, we determine to remove them in order to avoid merging several cars into a single blob when passing together. In our approach the technique detailed by Xu et al. in [6] is employed. In this method, classification is applied to the derivatives of the input image to separate hard shadows from the vague ones. Then, color invariance is exploited to distinguish hard shadow edges from material edges and finally image illumination is obtained via solving the standard Poisson Equation.

Since it is difficult to balance the amount of shadows removed and the vehicles shape preservation, we determine applying this algorithm with a noticeable strong effect by adjusting the tolerance for the input and invariant images to 0.00015 and 0.017 respectively. We considered that interest objects area would be improved in further stages of the pipeline as it results crucial to keep only the regions considering vehicles to obtain a feasible tracking.

4.4. Filling & Filtering

The current section details how images must be adapted in order to build uniform shapes able to track in further stages. As mentioned before, the data specified concerns hard shadows, expecting that removing effect will affect, simultaneously, to the interest objects themselves.

Firstly, we apply a morphological dilation with an structuring element of size (5,5). The idea behind is to merge dense pixel regions into bigger blobs while isolated noise increases in a controlled size as observed in Fig. 3a. Next we proceed to fill blank areas inside the objects applying 8-connectivity and filtering noise using area filtering with 4-connectivity (Figs. 3b and 3c). In order to increase the effect of blob filling we determined to employ morphological closing with structuring element (5,5). Its effect is shown in Fig 3d. Even qualitatively the results seem to have no effect, posterior tracking demands of this operation in order to achieve its best performance.

At this point, vehicles have improved their shape resulting in consistent blobs. In pursuance to ease things for tracking, we succeed on applying a morphological opening with structuring element with size (14,14). The consequence is obtaining more homogeneous shapes denoting the vehicles and removing residual artifacts in the regions of interest (Fig. 3e). As counterpart, far objects are also eliminated so tracking area results slightly smaller. Finally, another dilation with structuring element (7,7) is employed in order to increase blob sizes and smooth contours. The output from area filtering and filling is shown in Fig. 3f

4.5. Tracking

Kalman filters [7] is a common algorithm to model dynamical systems under noise conditions. Thus, vehicle tracking is a good candidate to approach with this method. First, because car tracks are evolving over time in a non-deterministic way (eg. change of lane or direction). And second, the observations at this point of our system, corresponding to binary blobs, are subject to noise, introduced in all the previous stages.

We model a car track with the following dynamic equations:

$$p^{(t)} = p^{(t-1)} + (\Delta\, t)v^{(t-1)} + \epsilon \tag{1}$$

$$v^{(t)} = v^{(t-1)} + \xi \tag{2}$$

Where ϵ and ξ are random variables with a Gaussian distribution, superscripts denote time steps, and $p, v \in \mathbb{R}^2$ denote the car location and velocity respectively.

Since our system aims to track multiple vehicles at the same time, the problem of assigning the proper blob to each track arises. We proceed the following way. At initial time $t = 0$, we create a Kalman filter for each blob centroid extracted from the initial frame, and predict the next car location for each filter. Then, for each subsequent time $t = N$, we use the Hungarian algorithm [8] [9] to match the current predictions with the blob centroids extracted for the current frame. We propose a global distance criterion, which is minimized by the Hungarian algorithm:

$$D = \min_{assign} \sum_{(i,j)\in assign} euclid(p_i, c_j) \tag{3}$$

Where D is the global distance, computed as the sum of euclidean distances between assigned pairs of predicted car locations p_i and blob centroids c_j. With the resulting assignment the following steps are performed:

- If a pair of blob centroid and car location prediction has a distance greater than a given threshold, the match is undone. We chose a value of 100 for this threshold.
- When the number of blob centroids and filters does not match or some pairs does not satisfy the above constraint, there will be unpaired elements:
  - For each unpaired blob centroid, a new kalman filter is created.
  - For each unpaired kalman prediction, we increment its *disappeared counter*.

The disappeared counter is initialized to zero during the filter creation. When the counter exceeds a given threshold, we assume either, the vehicle abandoned the scene or the filter tracked a false positive blob. In both cases, the filter is removed. We selected a disappeared threshold of 4.

Once the assignments and postprocessing are done, we update the internal state of each paired kalman filter with the observed blob centroid. Finally, a new prediction for time $t = N + 1$ is performed for all kalman filters kept in $t = N$.

4.6. Speed Estimation

In order to have an estimation of the speed for each car tracked, we follow the theoretical definition given by kinematics under the assumption of no acceleration, taking the quotient between the space traveled under a certain period of time. Following this approach, since time comes directly from the discrete frame ordering provided by the video, it all comes down to obtain an accurate measurement of the real distance traveled by the car. This measurement will directly be affected by the effect of perspective coming from the camera projection, and also by the possible bounding box displacement coming from a slightly miss detection from the tracker. These facts make that almost impossible to obtain a reliable estimation from a single measurement, being necessary to apply an statistic approach, as averaging or cleaning outliers from a set of estimated values in order to get a reliable estimation. Perspective can be reduced by means of an homography computed with the DLT algorithm [10] thanks to the known geometry of the road. Although the homography almost eliminates the effect of perspective in the plane defined by the road, cars are not well transformed by it, since camera is not far enough to assume that 3d objects will be well mapped by a planar homography. This error will cause a linear variation on the values measured depending if cars are approaching or moving away from the camera. Once perspective has been removed from the road plane, an approximation of the pixel to meter ratio is done by manually measuring the length of road lines. For estimating the distance traveled for each tracked object, detected centroids are transformed into the warped space, where we compute the euclidean norm between consecutive measurements. Finally, if not known, we assume 30 fps for the sequences.

4.7. Additional applications

Besides speed estimation, some additional metrics were implemented. In particular, leveraging on the tracking system, we introduced counters for the current and total vehicles passed in a road, as well as an estimation of the traffic in terms of vehicles per minute. It illustrates the potential of a computer vision system, which offers a cheap yet powerful replace to traditional solutions based on inductive loops or equivalent hardware, more difficult to install and maintain.

The system allows to define the detection area, where the traffic statistics are extracted. Thus, information can be particularized to specific roads captured by a camera with a wider field of view, as shown in the examples of Fig. 4.

5. EVALUATION

Since the dataset employed has no ground truth, evaluation is done qualitatively. In general, our system present a good performance in car detection and tracking (Fig. 4, a-c). There are some related issues with detection of distant vehicles, since the morphology applied tends to remove them in a trade-off to avoid most of the false positive blobs (Fig. 4, e-f).

In addition, it has been observed that some adjacent vehicles are merged during the morphology stage in a unique blob, provoking a unique track being kept in subsequent frames (Fig. 4d). This issue causes huge variations in speed estimation, as illustrated in Fig. 5, with plots for vehicles 1, 5, 6 and 16 presenting really high spikes. In order to avoid these extreme values in the final result for the speed, we opt for a statistical filtering, taking only values at maximum 1.2 times the standard deviation of the data.

It is worthing to note the good results avoiding to track the hard shadows present in the dataset sequences, as observed in all the images of Fig. 4. The free shadow mask obtained from an original input image is shown in Fig. 2a where some isolated noise is appreciated. This mask is applied to the original background subtraction prediction to provide the material to the filling and filtering stage (Fig. 2b).

6. CONCLUSIONS

We have presented a computer vision system for traffic monitoring, with applicability to vehicle counting, tracking and speed estimation. If offers a powerful replace to other solutions more expensive and difficult to install and maintain, as inductive loops.

The system has properly dealt with hard shadows although adjusting parameters results tedious in order to correctly preserve objects shape. Morphological operations have overcome this limitation but fail on keeping distant vehicles.

We have seen Kalman filters-based tracking presents a good trade-off between simplicity and performance, although adjacent objects remain as a challenging problem with room for further improvement.

Speed estimation should be adjusted since it has been proved to suffer from miss detections due to relying on the distance given by the centroids of the bounding boxes. A trade-off between the two values could be employed in order to detect wrong values in the estimations.

7. FURTHER WORK

The developed tools could be introduced into more complex applications such as collision avoidance or traffic jams prediction. Both fields would have a direct effect in safety as we have mentioned in Sec. 1. Furthermore, it could also be applied to infractions detection, such as driving through the left lane while the right one is free or overtaking vehicles on the left lane through the right lane.

8. REFERENCES

[1] D. Felguera-Martin, J. T. Gonzalez-Partida, P. Almorox-Gonzalez, and M. Burgos-García. Vehicular traffic surveillance and road lane detection using radar interferometry. *IEEE Transactions on Vehicular Technology*, 61(3):959–970, March 2012.

[2] C. M. J. Tampere and L. H. Immers. An extended kalman filter application for traffic state estimation using ctm with implicit mode switching and dynamic parameters. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 209–216, Sept 2007.

[3] A. Koutsia, T. Semertzidis, K. Dimitropoulos, N. Grammalidis, and K. Georgouleas. Intelligent traffic monitoring and surveillance with multiple cameras. In *2008 International Workshop on Content-Based Multimedia Indexing*, pages 125–132, June 2008.

[4] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret. Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE Access*, 5:18042–18050, 2017.

[5] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 393–400, Washington, DC, USA, 2014. IEEE Computer Society.

[6] Li Xu, Feihu Qi, and Renjie Jiang. Shadow removal from a single image. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 2, pages 1049–1054. IEEE, 2006.

[7] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[8] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[9] Harold W. Kuhn. Variants of the Hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3:253–258, 1956.

[10] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

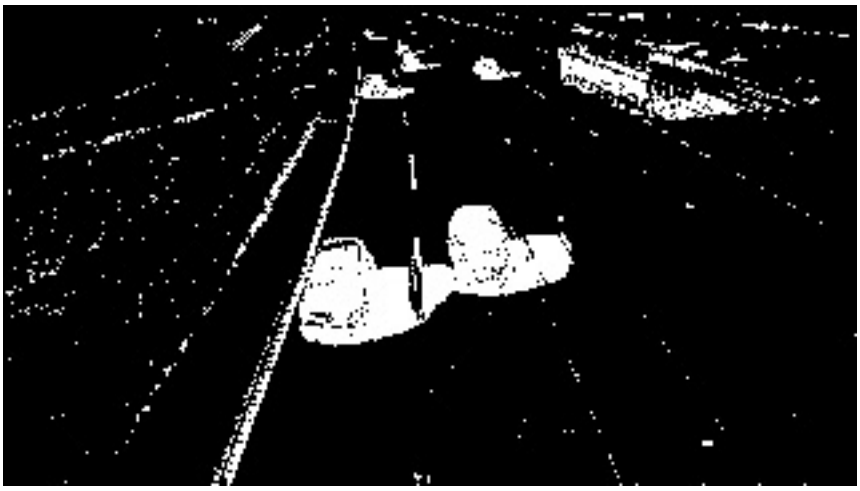
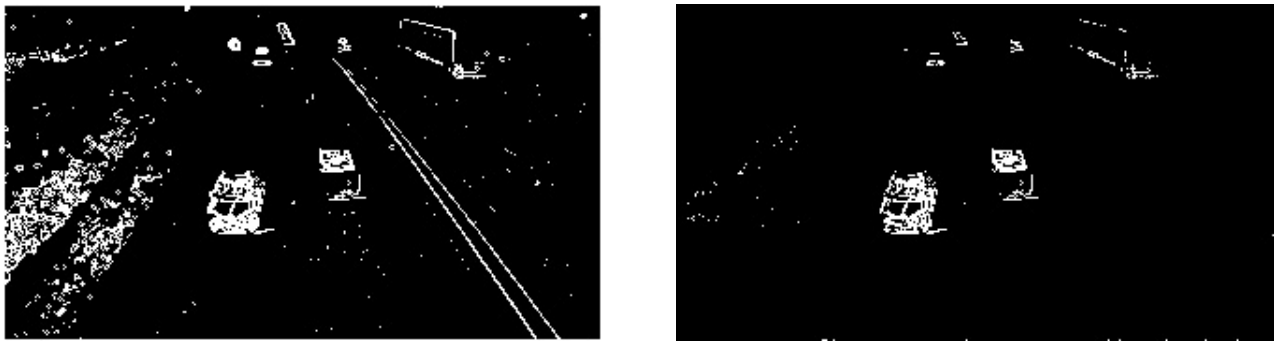


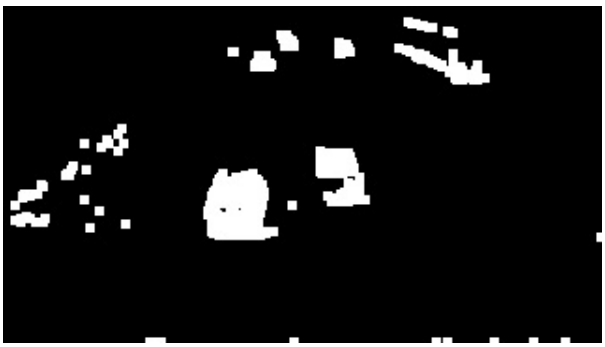
Fig. 1: Original Prediction



(a) Shadow mask obtained from original input image

(b) Prediction masked with shadow free image

Fig. 2: Shadow removal stage



(a) Dilation with st. element (5,5)

(b) Area Filling (8 connectivity)

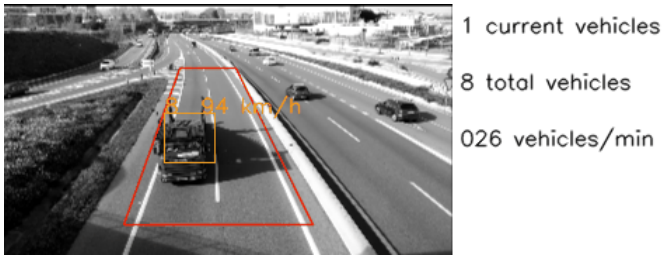
(c) Area Filtering (4 connectivity)

(d) Closing with st. element (5,5)

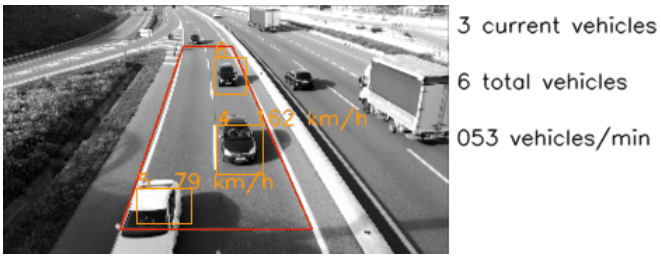
(e) Opening with st. element (14,14)

(f) Dilation with st. element (7,7)

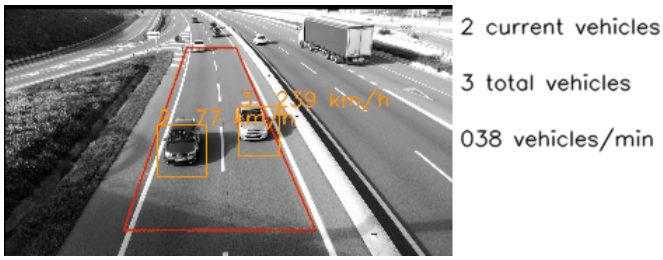
Fig. 3: Area Filling, Filtering & Morphology Stages



(a) The truck is correctly tracked

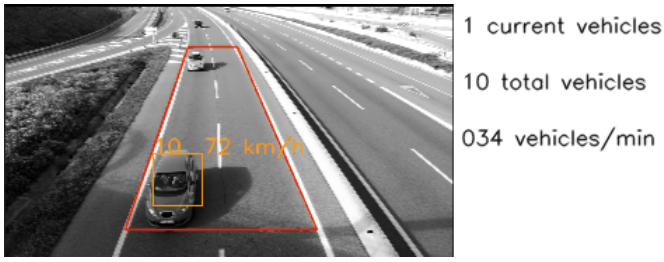


(b) All vehicles are correctly tracked

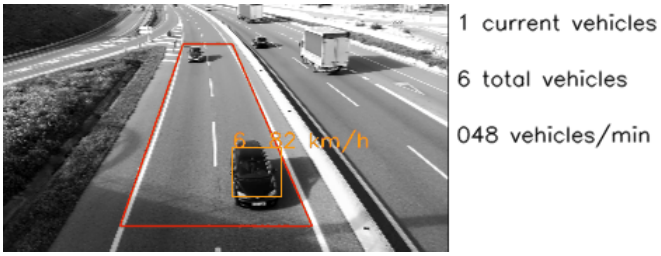


(c) Even parallel vehicles are correctly tracked

(d) Parallel vehicles can merge



(e) Nearby vehicles are tracked while distant ones still remain untracked



(f) Another case of untracked distant vehicles

Fig. 4: Results on the final application

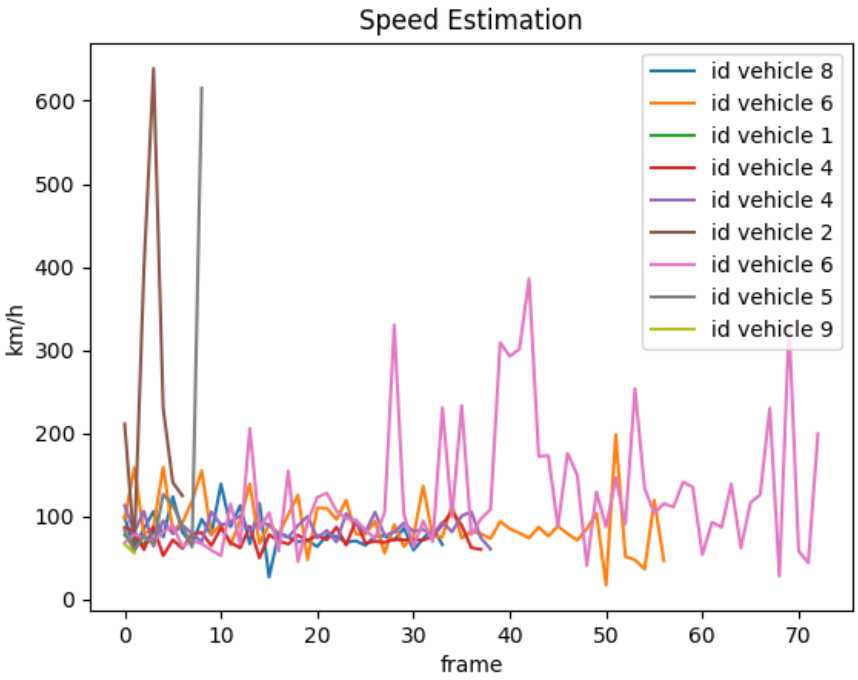


Fig. 5: Speed values for the highway sequence