Speed estimation evaluation on the KITTI benchmark based on motion and monocular depth information

Róbert-Adrian Rill[1,2]

[1]Faculty of Informatics, Eötvös Loránd University, Hungary
[2]Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Romania

Abstract

In this technical report we investigate speed estimation of the ego vehicle on the KITTI benchmark using state-of-the-art deep neural network based optical flow and single-view depth prediction methods. Using a straightforward intuitive approach and approximating a single scale factor, we evaluate several application schemes of the deep networks and formulate meaningful conclusions such as: combining depth information with optical flow improves speed estimation accuracy as opposed to using optical flow alone; the quality of the deep neural network methods influences speed estimation performance; using the depth and optical flow results from smaller crops of wide images degrades performance. With these observations in mind, we achieve a RMSE of less than 1 m/s for vehicle speed estimation using monocular images as input from recordings of the KITTI benchmark. Limitations and possible future directions are discussed as well.

1   Introduction

The significant progress in recent years makes deep learning solutions attractive in automotive industry applications [22]. In intelligent transportation systems, self-driving cars, advanced driver-assistance systems (ADAS) vehicle speed is one of the most important parameters for vehicle control and safety considerations. Velocity can be measured using wheel speed sensor, Inertial Navigation System (INS) or Global Positioning System (GPS). Although these methods can achieve high accuracy, they

also have limitations. Speed sensors involve a trade-off between accuracy and cost. INS suffers from integration drift, i.e. small errors accumulate over time. GPS is prone to signal interference and loss in blocked areas, and may provide unreliable data when accelerating or decelerating. Some researchers also propose model-based approaches (see, e.g., [20]), which, however, may be significantly affected by incorrect parameter estimates.

Other methods for automotive sensing technologies, including speed estimation, are Radio Detection and Ranging (RADAR) or Light Detection and Ranging (LiDAR) systems that use radio frequency or laser signals, respectively [1]. Although both systems are popular in the automotive industry, they have deficiencies. LiDAR systems can identify fine details of the 3D environment, but are greatly affected by unfavorable weather conditions and more importantly they are costly to produce and maintain. In contrast, RADAR systems are more robust, lightweight and cheap, but have much lower accuracy and resolution.

To address the limitations of conventional speed estimation methods, computer vision based approaches have become attractive alternatives in recent years. In this work we present a simple monocular vision-based speed estimation approach that exploits the recent advances in deep learning-based optical flow and monocular depth prediction methods. Optical flow is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer and the scene. Monocular depth estimation aims to obtain a representation of the spatial structure of a scene by determining the distance of objects from a single image. The two problems are fundamental in computer vision and represent highly correlated tasks (see, e.g., [23]). We rely on the intuition that the magnitude of optical flow is positively correlated with the moving speed of the observer and that objects closer to the camera appear to move faster than the more distant ones. We evaluate different schemes of our method on a representative subset of the KITTI dataset [5, 4], and achieve a RMSE of less than 1 m/s.

The rest of the paper is organized as follows. Section 2 presents related works providing a background and motivation for our study. Section 3 describes the KITTI dataset and the deep learning methods used in the present work, and details our speed estimation pipeline. The quantitative and qualitative results are presented in Section 4. Finally, Section 5 discusses the results and limitations, highlights future directions and concludes our work.

2

2 Related work

Vision based approaches represent a promising direction for vehicle speed estimation that may replace or complement traditional methods. Most of the works are concerned with estimating the velocity of the vehicles in traffic using a camera mounted for traffic surveillance. These methods involve different image processing techniques: background extraction [3, 18, 21, 13], image rectification [3, 18, 21, 11], detecting and tracking reference points [3, 18, 13] or centroids [21] over successive frames, converting the velocity displacement vectors from the image to the real-world coordinate system. The state-of-the-art results of deep learning in vision tasks makes object detection and tracking also suitable in the task of speed estimation [11]. Alternatively, locating license plates on the vehicles is another promising direction [13]. One major drawback of these approaches is the use of a stationary camera, which ameliorates the complexity of the problem.

As opposed to traffic surveillance purposes, few studies address the problem from an ADAS perspective, namely estimating speed using monocular images from a camera mounted on the car. Some works estimate the relative speed of other participants in traffic (see, e.g., [16] or [10]). In the present study we are concerned with estimating the absolute speed of the car the camera is mounted on, also called the ego/forward/longitudinal speed.

In [15] the authors used sparse optical flow to track feature points on images from a downward-looking camera mounted on the rear axle of the car and achieved a mean error relative to GPS measurement of 0.121 m/s. However, the method works only in restricted conditions and was evaluated on self-collected data at low speed values. Han [7] used projective geometry concepts to estimate relative and absolute speed in different case studies. Using black box footages, a maximum of 3% difference was reported for higher ego-speed values when compared to GPS measurements. The major limitation of this study is the assumption of known distances between stationary objects such as lane markings. Banerjee et al. [2] used a rather complicated neural network architecture trained on self-collected data and reported an RMSE of 10 mph on the KITTI benchmark [4].

In this work we investigate a simple intuitive approach for ego-speed estimation using deep neural network-based optical flow and monocular depth prediction methods. We achieve a RMSE of less than 1 m/s on recordings from the KITTI dataset.

3 Methods

3.1 The KITTI dataset

The KITTI Vision Benchmark Suite[1] [5, 4] is a real-world dataset consisting of 6 hours of traffic scenario recordings captured while driving in and around a mid-size city. The traffic situations range from highways over rural areas to innercity scenes with many static and dynamic objects.

The recording platform consisted of high resolution stereo camera systems (two color and two grayscale video cameras) mounted on top of the vehicle, Velodyne laser scanner that produces more than one million 3D points per second, a high-precision localization system, and a powerful computer running a real-time database housed in the trunk of the car. The cameras, laser scanner and localization system were calibrated and synchronized, providing accurate timestamped ground truth data. The dataset consists of both the raw and rectified, color and grayscale image sequences, laser scans, object labels in the form of 3D tracklets, and online benchmarks for stereo matching and optical flow estimation, 3D visual odometry, and 3D object detection.

The introductory paper [5] focuses on presenting the benchmark challenges, their creation and use for evaluating state-of-the-art computer vision methods, while the follow-up work [4] provides technical details on the raw data itself, describing the recording platform, the data format and the utilities.

The KITTI raw dataset is further augmented by additional data and benchmarks presented in subsequent papers. The stereo 2015 / flow 2015 / scene flow 2015 benchmark is described in [14]. This is a novel and realistic dataset with semi-dense scene flow ground truth obtained by annotating 400 highly dynamic scenes (200 training and 200 test) from the KITTI raw data collection using detailed 3D models for all vehicles in motion. Reference [19] presents a dataset derived from the KITTI raw dataset, for training and evaluating depth completion and depth prediction techniques, comprising over 94k images annotated with high-quality semi-dense depth ground truth.

3.2 Optical flow estimation: FlowNet2 and PWC-Net

FlowNet2[2] [8] poses the problem of optical flow estimation as an end-to-end supervised learning task and uses convolutional neural networks (CNNs). As a conslidation

---

[1] http://www.cvlibs.net/datasets/kitti
[2] https://github.com/lmb-freiburg/flownet2

of the original FlowNet idea, it inherits the advantages of mastering large displacements, correct estimations of fine details, the potential to learn priors for specific scenarios and fast runtimes. FlowNet2, compared to its initial versions, shows large improvements in quality and speed due to focusing on the schedule of presenting the training data during learning, using a stacked architecture including warping of the second image with intermediate optical flow results, and elaborating on small displacements by introducing a subnetwork specialized on small subpixel motions.

PWC-Net[3] [17] makes significant improvements in model size and accuracy over existing CNN models for optical flow. The method was designed according to simple and well-established principles: using learnable feature pyramids, using the warping operation as a layer to estimate large motions and the use of a cost volume layer. PWC-Net is 17 times smaller in size and easier to train than FlowNet2, and achieves state-of-the results on the KITTI 2015 benchmark [14].

While FlowNet2 achieves impressive performance by stacking basic models into a large capacity model, the much smaller PWC-Net obtains similar or better results by embedding classical principles into the network itself. For more details about the two methods and other optical flow estimation approaches please see the cited works and the references therein.

3.3 Monocular depth estimation: MonoDepth and MegaDepth

MonoDepth[4] [6] innovates beyond existing learning based single image depth estimation methods by replacing the use of large quantities and difficult to obtain quality ground truth training data with easier to obtain binocular stereo footage. The authors propose a CNN network architecture that performs unsupervised depth prediction by posing the task as an image reconstruction problem and using a novel training objective that enforces left-right consistency between the disparities of the left and right color images from a calibrated stereo pair. To obtain better accuracy, at test time MonoDepth requires a single input image and computes disparity for its horizontally flipped counterpart too. MonoDepth outperforms supervised methods on the KITTI 2015 benchmark [14].

MegaDepth[5] [12] refers to a large depth dataset generated via modern structure-from-motion and multi-view stereo methods from multi-view Internet photo collections. The authors propose data cleaning methods and augmentation of the data

---

[3]https://github.com/sniklaus/pytorch-pwc
[4]https://github.com/mrharicot/monodepth
[5]https://github.com/lixx2938/MegaDepth

with ordinal depth relations using semantic segmentation, and a corresponding loss function for training. The models trained on MegaDepth exhibit high accuracy and strong generalization ability to novel scenes, including the KITTI dataset.

One important note to make is that MonoDepth predicts disparity values, which can be converted to depth by multiplying its inverse with the given focal length and baseline between the stereo cameras. On the other hand, the MegaDepth models do not predict metric depth, but ordinal depth defined up to a scale factor. For more details about the two methods and related single-view depth estimation approaches please see the cited works and the references therein.

3.4 Speed estimation pipeline

To estimate ego-speed from monocular images of a camera mounted on the car our approach employs a multistage process. In a first step we run one optical flow and one depth estimation algorithm on a given image frame. Then, the results at valid pixels are considered from a predefined crop of the original image, and the mean of the magnitude of optical flow vectors (denoted by OF in the following) is divided by the mean of the disparity (DISP) values. The valid pixels are obtained by imposing thresholds: $OF > 0.2$ and $DISP > 0.01$. The next step is the concatenation of the scaled speed estimates over the temporal dimension, i.e. over frames of a video. The aggregated vectors are temporally smoothed using a 1D convolution of size 25 with equal weights. Finally, the resulting smoothed lists are taken for multiple video recordings and a scaling factor is approximated that minimizes the ratio between the ground truth and predicted speed. This scaling factor is used to convert the estimated speed from the image domain to real-world units.

To summarize, the speed estimation pipeline steps are as follows:

1. Run optical flow and depth estimation method on image frame.

2. Compute the scaled speed for the given frame: consider the OF and DISP values at valid pixels from a predefined image crop, and compute the quotient between their means.

3. Repeat the previous steps for all the frames from a video and apply temporal smoothing.

4. Repeat the previous step for multiple videos and estimate scaling factor.

Several modifications can be made to the base speed estimation algorithm defined above; these will be experimented in Section 4.

(E_1) As speed is expected to highly correlate with optical flow, depth information could be disregarded, considering only OF.

(E_2) Instead of OF, one could consider using the magnitude of horizontal optical flow only (which is expected to highly correlate with the moving speed especially towards the edges of the image).

(E_3) Temporal smoothing can also be applied at the pixel-level separately for OF and DISP.

(E_4) As opposed to providing the full frame as input to the neural networks and taking the results on image crops, one could run the methods only on the crops.

(E_5) Different image crops can be considered when taking the OF and DISP results, including the full frame too.

The images in the KITTI dataset are of 1242×375 resolution. Before obtaining the dense optical flow and depth results, the input images need to be resized to the appropriate resolution. We also resize the results of the deep neural networks back to the original resolution as summarized in Table 1. Our choices provided the best test results when evaluating the methods on KITTI. To run the neural networks we used pre-trained weights provided in the corresponding Github repositories.

Method	Input resolution	Input resize	Output resize	Model used*
FlowNet2	divisible by 64	pad with zeros	trim zeros	FlowNet2
PWC-Net	1024×436	bilinear interpolation	anti-aliasing	default network
MonoDepth	512×256	anti-aliasing	anti-aliasing	city2kitti
MegaDepth	512×384	anti-aliasing	anti-aliasing	best generalization

Table 1: **Deep neural network technicalities.** *For details check the references and/or the Github repositories.

To evaluate our speed estimation pipeline we manually selected 15 recordings of rectified images from the left input color camera of the KITTI dataset [5]. The list of drive IDs is summarized in Table 2. These are reresentative videos where the car is moving almost always. We did not include recordings in which the car is stationary from start to finish, as zero speed might highly increase the accuracy of our method.

In experiment E_5 defined previously we evaluate our proposed speed estimation pipeline on three different image crops. They are defined in Table 3 and visualized on two sample images from the KITTI dataset. Reasons for using crops only include the possible unavailability of wide images, or memory and run-time considerations.

Number	KITTI ID
1	2011_09_26_drive_0001
2	2011_09_26_drive_0002
3	2011_09_26_drive_0005
4	2011_09_26_drive_0009
5	2011_09_26_drive_0014
6	2011_09_26_drive_0019
7	2011_09_26_drive_0027
8	2011_09_26_drive_0048
9	2011_09_26_drive_0056
10	2011_09_26_drive_0059
11	2011_09_26_drive_0084
12	2011_09_26_drive_0091
13	2011_09_26_drive_0095
14	2011_09_26_drive_0096
15	2011_09_26_drive_0104

Table 2: **List of KITTI drive IDs used in our experiments.**

Crop name	bounding box			
	x	y	w	h
cropB	720	180	200	120
cropG	700	100	400	240
cropR	640	20	580	340

Table 3: **Definition of image crops used in our experiments.** (x, y) defines the upper left corner and w, h the width and height of the bounding boxes.

4 Experiments and results

Figure 2 shows visualizations of the deep neural network methods used in our experiments on one sample frame from the KITTI dataset. Both optical flow estimation methods produce smooth flow fields with sharp motion boundaries. PWC-Net seems to be more robust against shadow effects. The depth prediction methods also show good visual quality, with MonoDepth able to capture object boundaries and thin structures more reliably.

We also evaluate quantitatively the methods. The optical flow estimation algorithms are evaluated on the 200 training images from the KITTI 2015 benchmark [14]. The results displayed in Table 4 show that PWC-Net outperforms FlowNet2. The

Figure 1: **Image crops used in our speed estimation experiments.** Bounding boxes are overlaid on two sample frames from the KITTI dataset (left: frame 14 of *drive_0027*, right: frame 26 of *drive_0095*). The definition of the bounding boxes is shown in Table 3.

depth prediction algorithms are evaluated on the given 1000 manually selected images from the full validation split of the derived depth prediction and completion KITTI dataset [19]. Table 5 shows the results for several error metrics. We can see that MonoDepth achieves better accuracy values compared to MegaDepth. In all four cases the results are in correspondence with those reported in the references presenting the methods.

Method	AEPE	Fl-all
FlowNet2	11.686	32.183%
PWC-Net	2.705	9.187%

Table 4: **Evaluation of optical flow estimation methods on the KITTI 2015 benchmark [14].** AEPE: average endpoint error; Fl-all: Ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$.

Method	RMSE	RMSE(log)	Abs Rel	Sq Rel	log10	Scale-inv.
MonoDepth	4.532	0.150	0.090	0.749	0.040	0.142
MegaDepth	6.719	0.336	0.322	1.994	0.124	0.289

Table 5: **Evaluation of depth estimation methods on the manual selection of the validation split of the derived depth prediction and completion KITTI 2017 dataset [19].**

Table 6 shows the speed estimation results for experiments E_1, E_2, E_3, E_4. We can formulate the following conclusions: in all cases considering depth information as well, as opposed to optical flow alone improves performance (E_1), replacing OF with horizontal optical flow results in slightly higher RMSE values (E_2), applying temporal smoothing at the pixel level may improve performance but only marginally
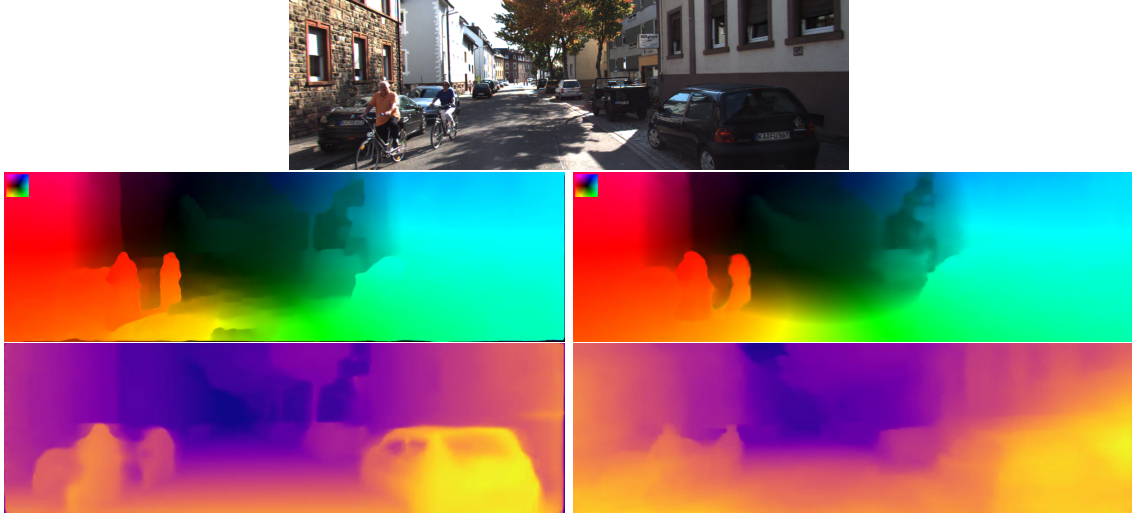
Figure 2: **Sample visualisation of network results.** From left to right and top to bottom: frame 71 of *drive_0095*, FlowNet2, PWC-Net, MonoDepth, MegaDepth. The colored square represents the color coding of optical flow.

(E_3), applying the deep neural network methods on a smaller image region (crop) – as opposed to first applying on the original wide frame and then extracting the results from the corresponding crop – influences performance considerably in a negative way (E_4).

Inspecting Table 6 we can see that the best results are obtained when PWC-Net is combined with MonoDepth. Figure 3 shows speed estimation results in this case using the base pipeline for two KITTI recordings. Our simple method is able to capture speed changes with low error in straight travel scenarios (Figure 3a), but having difficulty in cases when the car is taking a turn (around frame 200 on Figure 3b speed decreases as the car is turning left, yet optical flow increases on the right side – and in *cropG* too – of the wide KITTI images).

The results of experiment E_5 are illustrated in Table 7. We evaluate the base speed estimation algorithm on the three image crops from Table 3 and using the full frame as well. We can see that speed estimation accuracy improves in general as the size of the image used is increasing. Furthermore, note that similarly to the previous experiments, the best results are obtained by the PWC-Net – MonoDepth combination. When the full frame is used errors moderately decrease at car turning events (around frame 200 on Figure 3b), but overestimations are still present in dynamic scenes due to the motion of other cars for instance (after frame 400 on Figure 3b).

Method	Base pipeline	Horizontal optical flow (E_2)	Pixel-level smoothing (E_3)	Methods only on crop (E_4)
FlowNet2 (E_1)	2.921	3.005	2.919	3.093
PWC-Net (E_1)	2.472	2.621	2.472	3.170
FlowNet2 & MonoDepth	2.305	2.448	2.399	2.618
PWC-Net & MegaDepth	1.915	2.059	1.908	3.180
FlowNet2 & MegaDepth	2.485	2.526	2.475	2.901
PWC-Net & MonoDepth	**1.467**	1.707	1.865	2.967

Table 6: **Results of speed estimation experiments.** RMSE values are shown using $cropG$ defined in Table 3. $E_i, i \in \{1, 2, 3, 4\}$ refers to the experiments defined in the previous section.

Method	$cropB$	$cropG$	$cropR$	Full frame	TC
FlowNet2 & MonoDepth	2.170	2.305	2.558	2.370	2.138
PWC-Net & MegaDepth	2.363	1.915	2.015	1.786	1.445
FlowNet2 & MegaDepth	2.671	2.485	2.583	2.544	2.125
PWC-Net & MonoDepth	1.735	1.467	1.505	**1.178**	**0.977**

Table 7: **Results using different image crops.** RMSE values are shown for experiment E_5. Image crops are defined in Table 3. TC refers to commpensating for car turning events (for details see text).

In order to decrease speed overestimations at car turning events we experimented with a modification of the base algoritmic pipeline. In such cases the average of horizontal optical flow of the left part of the image has the same sign as the average from the right side. Whenever this condition is true, instead of the mean of the optical flow magnitude for the full wide frame, the absolute value of the difference between the means of horizontal optical flow of the left and right sides is computed, and divided by the mean disparity corresponding to the whole frame. Applying this compensation method for turning events (TC) decreases the RMSE to under 1 m/s, as shown in Table 7. This performance improvement is illustrated by Figure 3b as well.
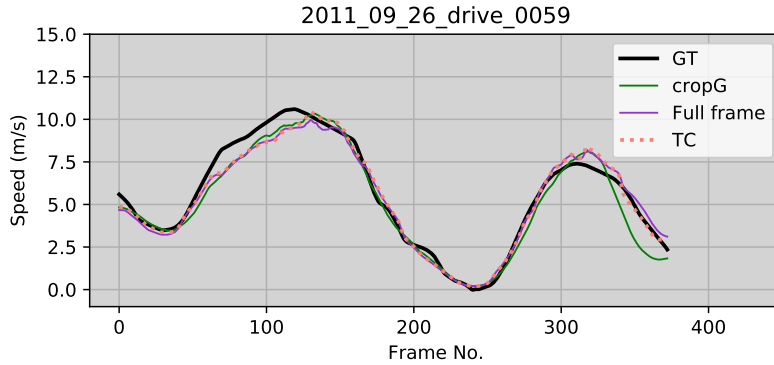
(a)
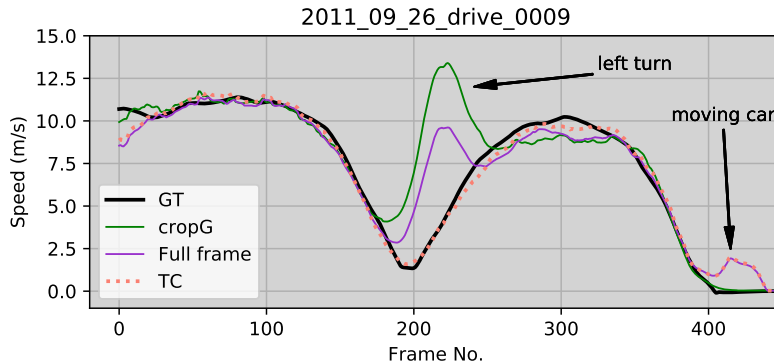
(b)

Figure 3: **Speed estimation results on two sample KITTI recordings.** The base pipeline was used with the PWC-Net and MonoDepth methods; *cropG* is defined in Table 3 and *Full frame* refers to the full wide image. TC refers to compensating for car turning events. For more details see text.

5   Discussion and conclusion

In this work we investigated a simple algorithm for ego-speed estimation using state-of-the-art deep neural network based optical flow estimation and monocular depth prediction on images from a camera mounted on the moving car. We relied on the intuition that optical flow magnitude is highly correlated with the moving speed of the observer and that the closer objects are to the observer the faster they appear to be moving. We defined a base algorithmic pipeline for speed estimation, evaluated several modifications to it and achieved a RMSE of less than 1 m/s on a subset of

12

the widely used KITTI dataset [5, 4].

We investigated two optical flow estimation (FlowNet2 [8] and PWC-Net [17]) and two depth estimation (MonoDepth [6] and MegaDepth [12]) algorithms. We evaluated these on ground truth data from the KITTI dataset and saw that PWC-Net and MonoDepth achieved better performance in several error metrics. The reason for this is probably some combination of the following: the MonoDepth model used in our work was fine-tuned on data from KITTI, MegaDepth does not predict metric depth but ordinal depth, it seems that during training of the PWC-Net model KITTI data was used as well. Nonetheless, the conclusion is that better performance optical flow and depth estimation methods result in better speed estimates. Besides, continuous efforts are made to improve these two fundamental computer vision algorithms, including their joint training in an unsupervised manner (see, e.g., [23]). Accordingly, fine-tuning to arbitrary images becomes easily accessible without the need for difficult to obtain ground truth labels.

We also evaluated several modifications to our proposed base algorithm and formulated meaningful conclusions. Experiment E_1 showed that optical flow alone reached reasonable performance, but using depth information too improved the results. Experiment E_2 illustrated that using horizontal optical flow gives lower performance than using the magnitude of the optical flow vectors. In experiment E_3 we saw that applying temporal smoothing at the pixel level (as opposed to applying it after image-level aggregation) can only marginally improve performance in some cases. Experiment E_4 illustrated that applying the deep neural network methods only on smaller crops of the original wide images significantly influences results in a negative manner. Experiment E_5 compared the base speed estimation pipeline using different image crops and proved that using larger regions is better, with the full frame providing the best overall result.

There are two limitations to our method that need to be mentioned. On one hand, speed is erroneously estimated when the car is turning. However, in such cases estimation errors can be corrected by taking into account that horizontal optical flow from the left and right side of the wide images have the same direction (see the last column of Table 7 and Figure 3b). On the other hand, the proposed method is most reliable when the background is static. For example in heavy traffic scenarios when the surrounding cars are moving as well, the correlation of optical flow with ego-speed might be small and speed can be over- (see Figure 3b) or underestimated. In such scenarios combining monocular depth estimation with semantic segmentation [9] represents one promising direction; and the estimation of relative speed can help, which is another problem where recent advances are being made (see, e.g., [16] or [10]). One might also argue that the deep neural network methods providing the

best speed estimates in our case were fine-tuned on ground truth data from KITTI. But, as explained above, efforts are being made to train such methods on unlabelled data [23].

To improve our simple method future works can treat the task as a regression problem and adopt for example a lightweight multilayer perceptron using as input the aggregated optical flow and depth results from different smaller regions of the original larger image(s). Another possibility is the exploitation of the more sophisticated convolutional neural network, which, however would possibly require a larger amount of training data. We close this work by noting that due to the recent and ongoing advancements in deep learning, monocular vision-based approaches are a promising direction for ego-speed estimation, and autonomous driving in general.

Acknowledgements

References

[1] H. Abuella, F. Miramirkhani, S. Ekin, M. Uysal, and S. Ahmed. Vildar - visible light sensing based speed estimation using vehicle's headlamps. *arXiv e-prints*, page arXiv:1807.05412, Jul 2018.

[2] K. Banerjee, T. Van Dinh, and L. Levkova. Velocity estimation from monocular video for automotive applications using convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 373–378, June 2017.

[3] S. Doğan, M. S. Temiz, and S. Külür. Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. In *Sensors*, 2010.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[6] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017.

[7] I. Han. Car speed estimation based on cross-ratio using video data of car-mounted camera (black box). *Forensic Science International*, 269:89–96, 2016.

[8] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

[9] H. Jiang, G. Larsson, M. Maire, G. Shakhnarovich, and E. Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 20–37, Cham, 2018. Springer International Publishing.

[10] M. Kampelmühler, M. Müller, and C. Feichtenhofer. Camera-based vehicle velocity estimation from monocular video. In *Computer Vision Winter Workshop (CVWW)*, February 2018.

[11] A. Kumar, P. Khorramshahi, W.-A. Lin, P. Dhar, J.-C. Chen, and R. Chellappa. A semi-automatic 2d solution for vehicle speed estimation from monocular videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[12] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[13] D. C. Luvizon, B. T. Nassu, and R. Minetto. A video-based system for vehicle speed measurement in urban roadways. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1393–1404, June 2017.

15

[14] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] X. Qimin, L. Xu, W. Mingming, L. Bin, and S. Xianghui. A methodology of vehicle speed estimation based on optical flow. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 33–37, Oct 2014.

[16] S. Salahat, A. Al-Janahi, L. Weruaga, and A. Bentiba. Speed estimation from smart phone in-motion camera for the next generation of self-driven intelligent vehicles. In *IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2017.

[17] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[18] M. S. Temiz, S. Kulur, and S. Doğan. Real time speed estimation from monocular video. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, XXXIX-B3:427–432, 2012.

[19] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[20] Q. Xu, X. Li, and C.-Y. Chan. A cost-effective vehicle localization solution using an interacting multiple model-unscented kalman filters (imm-ukf) algorithm and grey neural network. In *Sensors*, 2017.

[21] Y. G. Anil Rao, N. Sujith Kumar, H. S. Amaresh, and H. V. Chirag. Real-time speed estimation of vehicles from uncalibrated view-independent traffic cameras. In *TENCON 2015 - IEEE Region 10 Conference*, pages 1–6, Nov 2015.

[22] B. Yao and T. Feng. Machine learning in automotive industry. *Advances in Mechanical Engineering*, 2018.

[23] Y. Zou, Z. Luo, and J.-B. Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 38–55, Cham, 2018. Springer International Publishing.