End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners

Simon Hecker[1], Dengxin Dai[1], and Luc Van Gool[1,2]

[1] ETH Zurich, Zurich, Switzerland
{heckers,dai,vangool}@vision.ee.ethz.ch
[2] KU Leuven, Leuven, Belgium

Abstract. For human drivers, having rear and side-view mirrors is vital for safe driving. They deliver a more complete view of what is happening around the car. Human drivers also heavily exploit their mental map for navigation. Nonetheless, several methods have been published that learn driving models with only a front-facing camera and without a route planner. This lack of information renders the self-driving task quite intractable. We investigate the problem in a more realistic setting, which consists of a surround-view camera system with eight cameras, a route planner, and a CAN bus reader. In particular, we develop a sensor setup that provides data for a 360-degree view of the area surrounding the vehicle, the driving route to the destination, and low-level driving maneuvers (e.g. steering angle and speed) by human drivers. With such a sensor setup we collect a new driving dataset, covering diverse driving scenarios and varying weather/illumination conditions. Finally, we learn a novel driving model by integrating information from the surround-view cameras and the route planner. Two route planners are exploited: 1) by representing the planned routes on OpenStreetMap as a stack of GPS coordinates, and 2) by rendering the planned routes on TomTom Go Mobile and recording the progression into a video. Our experiments show that: 1) 360-degree surround-view cameras help avoid failures made with a single front-view camera, in particular for city driving and intersection scenarios; and 2) route planners help the driving task significantly, especially for steering angle prediction. Code, data and more visual results will be made available at http://www.vision.ee.ethz.ch/~heckers/Drive360.

Keywords: Autonomous driving · end-to-end learning of driving · route planning for driving · surround-view cameras · driving dataset

1 Introduction

Autonomous driving has seen dramatic advances in recent years, for instance for road scene parsing [1,2,3,4], lane following [5,6,7], path planning [8,9,10,11], and end-to-end driving models [12,13,14,15]. By now, autonomous vehicles have driven many thousands of miles and companies aspire to sell such vehicles in a few years. Yet, significant technical obstacles, such as the necessary robustness of driving models to adverse weather/illumination conditions [2,3,4] or the capability to anticipate potential risks in advance [16,17], must be overcome before assisted driving can be turned into full-fletched automated driving. At the same time, research on the next steps towards
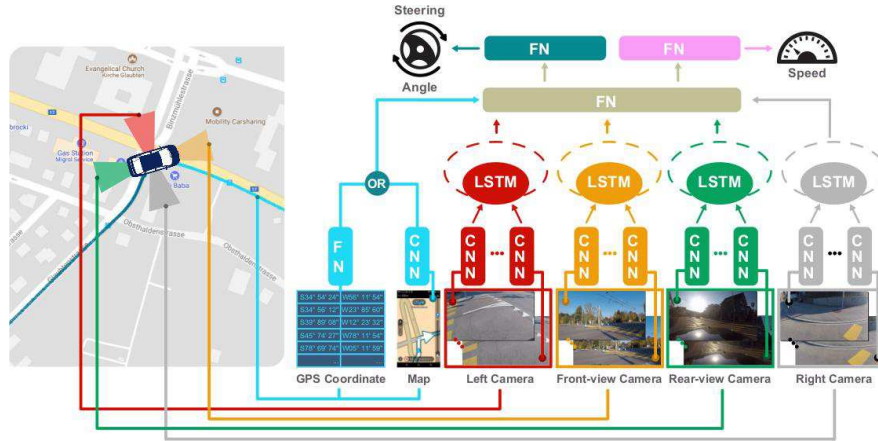
Fig. 1: An illustration of our driving system. Cameras provide a 360-degree view of the area surrounding the vehicle. The driving maps or GPS coordinates generated by the route planner and the videos from our cameras are synchronized. They are used as inputs to train the driving model. The driving model consists of CNN networks for feature encoding, LSTM networks to integrate the outputs of the CNNs over time; and fully-connected networks (FN) to integrate information from multiple sensors to predict the driving maneuvers.

'complete' driving systems is becoming less and less accessible to the academic community. We argue that this is mainly due to the lack of large, shared driving datasets delivering more *complete* sensor inputs.

Surround-view cameras and route planners. Driving is inarguably a highly visual and intellectual task. Information from all around the vehicle needs to be gathered and integrated to make safe decisions. As a virtual extension to the limited field of view of our eyes, side-view mirrors and a rear-view mirror are used since 1906 [18] and in the meantime have become obligatory. Human drivers also use their internal maps [19,20] or a digital map to select a route to their destination. Similarly, for automated vehicles, a decision-making system must select a route through the road network from its current position to the requested destination [21,22,23].

As said, a single front-view camera is inadequate to learn a safe driving model. It has already been observed in [24] that upon reaching a fork - and without a clearcut idea of where to head for - the model may output multiple widely discrepant travel directions, one for each choice. This would result in unsafe driving decisions, like oscillations in the selected travel direction. Nevertheless, current research often focuses on this setting because it still allows to look into plenty of challenges [6,25,12]. This is partly due to the simplicity of training models with a single camera, both in terms of available datasets and the complexity an effective model needs to have. Our work includes a surround-view camera system, a route planner, and a data reader for the vehicle's CAN bus. The setting provides a 360-degree view of the area surrounding the vehicle, a planned driving route, and the 'ground-truth' maneuvers by human drivers. Hence, we obtain

a learning task similar to that of a human apprentice, where a (cognitive/digital) map gives an overall sense of direction, and the actual steering and speed controls need to be set based on the observation of the local road situation.

Driving Models. In order to keep the task tractable, we chose to learn the driving model in an end-to-end manner, i.e. to map inputs from our surround-view cameras and the route planner directly to low-level maneuvers of the car. The incorporation of detection and tracking modules for traffic agents (e.g. cars and pedestrians) and traffic control devices (e.g. traffic lights and signs) is future work. We designed a specialized deep network architecture which integrates all information from our surround-view cameras and the route planner, and then maps these sensor inputs directly to low-level car maneuvers. See Figure 1 and the supplemental material for the network's architecture. The route planner is exploited in two ways: 1) by representing planned routes as a stack of GPS coordinates, and 2) by rendering the planned routes on a map and recording the progression as a video.

Our main contributions are twofold: 1) a new driving dataset of 60 hours, featuring videos from eight surround-view cameras, two forms of data representation for a route planner, low-level driving maneuvers, and GPS-IMU data of the vehicle's odometry; 2) a learning algorithm to integrate information from the surround-view cameras and planned routes to predict future driving maneuvers. Our experiments show that: a) 360-degree views help avoid failures made with a single front-view camera; and b) a route planner also improves the driving significantly.

2   Related Work

Our work is relevant for 1) driving models, 2) assistive features for vehicles with surround view cameras, 3) navigation and maps, and 4) driving scene understanding.

2.1   Driving Models for Automated Cars

Significant progress has been made in autonomous driving, especially due to the deployment of deep neural networks. Driving models can be clustered into two groups [7]: mediated perception approaches and end-to-end mapping approaches, with some exceptions like [7]. Mediated perception approaches require the recognition of all driving-relevant objects, such as lanes, traffic signs, traffic lights, cars, pedestrians, etc. [26,1,27]. Excellent work [28] has been done to integrate such results. This kind of systems developed by the automotive industry represent the current state-of-the-art for autonomous driving. Most use diverse sensors, such as cameras, laser scanners, radar, and GPS and high-definition maps [29]. End-to-end mapping methods construct a direct mapping from the sensory input to the maneuvers. The idea can be traced back to the 1980s, when a neural network was used to learn a direct mapping from images to steering angles [24]. Other end-to-end examples are [5,25,12,14,15]. In [12], the authors trained a neural network to map camera inputs directly to the vehicle's ego-motion. Methods have also been developed to explain how the end-to-end networks work for the driving task [30] and to predict when they fail [17]. Most end-to-end work has been demonstrated with a front-facing camera only. To the best of our knowledge, we present the

first end-to-end method that exploits more realistic input. Please note that our data can also be used for mediated perception approaches. Recently, reinforcement learning for driving has received increasing attention [31,32,33]. The trend is especially fueled by the release of excellent driving simulators [34,35].

2.2   Assistive Features of Vehicle with Surround View Cameras

Over the last decades, more and more assistive technologies have been deployed to vehicles, that increase driving safety. Technologies such as lane keeping, blind spot checking, forward collision avoidance, adaptive cruise control, driver behavior prediction etc., alert drivers about potential dangers [36,37,38,39]. Research in this vein recently has shifted focus to surround-view cameras, as a panoramic view around the vehicle is needed for many such applications. Notable examples include object detection, object tracking, lane detection, maneuver estimation, and parking guidance. For instance, a bird's eye view has been used to monitor the surrounding of the vehicle in [40]. Trajectories and maneuvers of surrounding vehicles are estimated with surround view camera arrays [41,42]. Datasets, methods and evaluation metrics of object detection and tracking with multiple overlapping cameras are studied in [43,44]. Lane detection with surround-view cameras is investigated in [45] and the parking problem in [46]. Advanced driver assistance systems often use a 3-D surround view, which informs drivers about the environment and eliminates blind spots [47]. Our work adds autonomous driving to this list. Our dataset can also be used for all aforementioned problems; and it provides a platform to study the usefulness of route planners.

2.3   Navigation and Maps

In-car navigation systems have been widely used to show the vehicle's current location on a map and to inform drivers on how to get from the current position to the destination. Increasing the accuracy and robustness of systems for positioning, navigation and digital maps has been another research focus for many years. Several methods for high-definition mapping have been proposed [48,49], some specifically for autonomous driving [50,51]. Route planning has been extensively studied as well [52,53,54,55,56], mainly to compute the fastest, most fuel-efficient, or a customized trajectory to the destination through a road network. Yet, thus far their usage is mostly restricted to help human drivers. Their accessibility as an aid to learn autonomous driving models has been limited. This work reports on two ways of using two kinds of maps: a s-o-t-a commercial map TomTom Maps [3] and the excellent collaborative project OpenStreetMaps [57].

While considerable progress has been made both in computer vision and in route planning, their integration for learning driving models has not received due attention in the academic community. A trending topic is to combine digital maps and street-view images for accurate vehicle localization [58,59,60,61].

---

[3] `https://www.tomtom.com/en_us/drive/maps-services/maps/`

2.4 Driving Scene Understanding

Road scene understanding is a crucial enabler for assisted or autonomous driving. Typical examples include the detection of roads [62], traffic lights [63], cars and pedestrians [64,65,1,2], and the tracking of such objects [66,67,68]. We refer the reader to these comprehensive surveys [69,70]. Integrating recognition results like these of the aforementioned algorithms may well be necessary but is beyond the scope of this paper.

3 The Driving Dataset

We first present our sensor setup, then describe our data collection, and finally compare our dataset to other driving datasets.

3.1 Sensors

Three kinds of sensors are used for data collection in this work: cameras, a route planner (with a map), and a USB reader for data from the vehicle's CAN bus.

Cameras. We use eight cameras and mount them on the roof of the car using a specially designed rig with 3D printed camera mounts. The cameras are mounted under the following angles: 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315° relative to the vehicle's heading direction. We installed GoPro Hero 5 Black cameras, due to their ease of use, their good image quality when moving, and their weather-resistance. All videos are recorded at 60 frames per second (fps) in 1080p. As a matter of fact, a full 360-degree view can be covered by four cameras already. Please see Figure 2 for our camera configuration.

Route Planners. Route planners have been a research focus over many years [53,54]. While considerable progress has been made both in computer vision and in route planning, their integration for learning to drive has not received due attention in the academic community. Routing has become ubiquitous with commercial maps such as Google Maps, HERE Maps, and TomTom Maps, and on-board navigation devices are virtually in every new car. Albeit available in a technical sense, their routing algorithms and the underlying road networks are not yet accessible to the public. In this work, we exploited two route planners: one based on TomTom Map and the other on OpenStreetMap.

TomTom Map represents one of the s-o-t-a commercial maps for driving applications. Similar to all other commercial counterparts, it does not provide open APIs to access their 'raw' data. We thus exploit the visual information provided by their Tom-Tom GO Mobile App [71], and recorded their rendered map views using the native screen recording software supplied by the smart phone, an iPhone 7. Since map rendering comes with rather slow updates, we capture the screen at 30 fps. The video resolution was set to 1280×720 pixels.

Apart from the commercial maps, OpenStreetMaps (OSM) [57] has gained a great attention for supporting routing services. The OSM geodata includes detailed spacial and semantic information about roads, such as name of roads, type of roads (e.g. highway or footpath), speed limits, addresses of buildings, etc. The effectiveness of OSM

(a) our camera rig   (b) rig on the vehicle
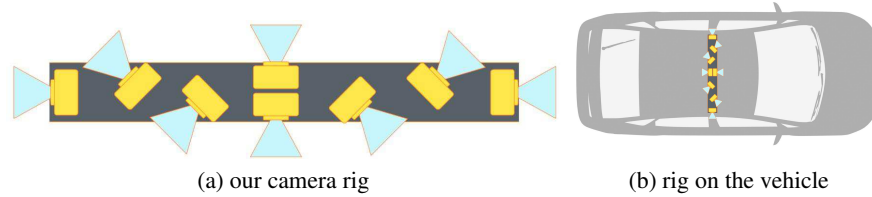
Fig. 2: The configuration of our cameras. The rig is 1.6 meters wide so that the side-view cameras can have a good view of road surface without the obstruction by the roof of the vehicle. The cameras are evenly distributed laterally and angularly.

for Robot Navigation has been demonstrated by Hentschel and Wagner [72]. We thus, in this work, use the real-time routing method developed by Luxen and Vetter for OSM data [73] as our second route planner. The past driving trajectories (a stack of GPS coordinates) are provided to the routing algorithm to localize the vehicle to the road network, and the GPS tags of the planned road for the next 300 meters ahead are taken as the representation of the planned route for the 'current' position. Because the GPS tags of the road networks of OSM are not distributed evenly according to distance, we fitted a cubic smoothing spline to the obtained GPS tags and then sampled 300 data points from the fitted spline with a stride of 1 meter. Thus, for the OSM route planner, we have a 300×2 matrix (300 GPS coordinates) as the representation of the planned route for every 'current' position.

Human Driving Maneuvers. We record low level driving maneuvers, i.c. the steering wheel angle and vehicle speed, registered on the CAN bus of the car at 50Hz. The CAN protocol is a simple ID and data payload broadcasting protocol that is used for low level information broadcasting in a vehicle. As such, we read out the specific CAN IDs and their corresponding payload for steering wheel angle and vehicle speed via a CAN-to-USB device and record them on a computer connected to the bus.

Vehicle's Odometry. We use the GoPro cameras' built-in GPS and IMU module to record GPS data at 18Hz and IMU measurements at 200Hz while driving. This data is then extracted and parsed from the meta-track of the GoPro created video.

3.2   Data Collection

Synchronization. The correct synchronization amongst all data streams is of utmost importance. For this we devised an automatic procedure that allows for synchronization to GPS for fast dataset generation. During all recording, the internal clocks of all sensors are synchronized to the GPS clock. The resulting synchronization error for the video frames is up to 8.3 milliseconds (ms), i.e. half the frame rate. If the vehicle is at a speed of 100 km/h, the error due to the synchronization for vehicle's longitudinal position is about 23 cm. We acknowledge that a camera which can be triggered by accurate trigger signals are preferable with respect to synchronization error. Our cameras, however, provide good photometric image quality and high frame rates, at the price of moderate synchronization error. The synchronization error of the maps to our video

frame is up to 0.5 s. This is acceptable, as the planned route (regardless of its representation) is only needed to provide a global view for navigation. The synchronization error of the CAN bus signal to our video frames is up to 10 ms. This is also tolerable as human drivers issue driving actions at a relative low rate. For instance, the mean reaction times for unexpected and expected human drivers are 1.3 and 0.7 s [74].

Drive360 **dataset**. With the sensors described, we collect a new dataset *Drive360*. *Drive360* is recorded by driving in (around) multiple cities in Switzerland. We focus on delivering realistic dataset for training driving models. Inspired by how a driving instructor teaches a human apprentice to drive, we chose the routes and the driving time with the aim to maximize the opportunity of exposing to all typical driving scenarios. This reduces the chance of generating a biased dataset with many 'repetitive' scenarios, and thus allowing for an accurate judgment of the performance of the driving models. *Drive360* contains 60 hours of driving data.

The drivers always obeyed Swiss driving rules, such as respecting the driving speed carefully, driving on the right lane when not overtaking a vehicle, leaving the required amount of distance to the vehicle in front etc. We have a second person accompanying the drivers to help (remind) the driver to always follow the route planned by our route planner. We have used a manual setup procedure to make sure that the two route planners generate the 'same' planned route, up to the difference between their own representations of the road networks. After choosing the starting point and the destination, we first generate a driving route with the OSM route planner. For TomTom route planner, we obtain the same driving route by using the same starting point and destination, and by adding a consecutive sequence of waypoints (intermediate places) on the route. We manually verified every part of the route before each driving to make sure that the two planned routes are in deed the same. After this synchronization, TomTom Go Mobile is used to guide our human drivers due to its high-quality visual information. The data for our OSM route planner is obtained by using the routing algorithm proposed in [73]. In particular, for each 'current' location, the 'past' driving trajectory is provided to localize the vehicle on the originally planned route in OSM. Then the GPS tags of the route for the next 300 meters ahead are retrieved.

3.3 Comparison to other datasets

In comparison to other datasets, see Table 1, ours has some unique characteristics.
Planned routes. Since our dataset is aimed at understanding and improving the fallacies of current end-to-end driving models, we supply map data for navigation and offer the only real-world dataset to do so. It is noteworthy that planned routes cannot be obtained by post-processing the GPS coordinates recorded by the vehicle, because planned routes and actual driving trajectories intrinsically differ. The differences between the two are resulted by the actual driving (e.g. changing lanes in road construction zones and overtaking a stopped bus), and are indeed the objectives meant to be learned by the driving models.
Surround views and low-level driving maneuvers. Equally important, our dataset is the only dataset working with real data and offering surround-view videos with low-level driving maneuvers (e.g. steering angle and speed control). This is particularly valuable for end-to-end driving as it allows the model to learn correct steering

Datasets	driving time (h)	# cams	fps	maneuvers, e.g. steering	route planner	GPS IMU	control of cam pose	data type	lidar
Drive360	60	8, ▨ ◈	60	✓	✓	✓	✓	real	✗
KITTI [26]	1	2, ☗	10	✗	✗	✓	✓	real	✓
Cityscapes [1]	< 100	2, ☗	16	✗	✗	✓	✓	real	✗
Comma.ai	7.3	1, ☗	20	✓	✗	✓	N.A.	real	✗
Oxford [75]	214	4, ▨	16	✗	✗	✓	✓	real	✓
BDDV [12]	10k	1, ☗	30	✗	✗	✓	✗	real	✗
Udacity [76]	1.1	3, ▨	30	✓	✗	✓	N.A.	real	✗
GTA	N.A.	1 ☗	✓	✓	✗	N.A.	rendered	synthetic	✗

Table 1: Comparison of our dataset to others compiled for driving tasks (cam=camera).

for lane changes, requiring 'mirrors' when carried out by human drivers, or correct driving actions for making turns at intersections. Compared with BDDV [12] and Oxford dataset [75], we offer low level driving maneuvers of the vehicle via the CAN bus, whereas they only supply the cars ego motion via GPS devices. This allows us to predict input control of the vehicle which is one step closer to a fully autonomous end-to-end trained driving model. Udacity [76] also offers low-level driving maneuvers via the CAN bus. It, however, lacks of route planners and contains only a few hours of driving data.

Dataset focus. As shown in Table 1, there are multiple datasets compiled for tasks relevant to autonomous driving. These datasets, however, all have their own focuses. KITTI, Cityscapes and GTA focus more on semantic and geometric understanding of the driving scenes. Oxford dataset focus on capturing the temporal (seasonal) changes of driving scenes, and thus limited the driving to a 'single' driving route. BDDV [12] is a very large dataset, collected from many cities in a crowd-sourced manner. It, however, only features a front-facing dashboard camera.

4   Approach

The goal of our driving model is to map directly from the planned route, the historical vehicle states and the current road situations, to the desired driving actions.

4.1   Our Driving Model

Let us denote by I the surround-view video, P the planned route, L the vehicle's location, and S and V the vehicle's steering angle and speed. We assume that the driving model works with discrete time and makes driving decisions every $1/f$ seconds. The inputs are all synchronized and sampled at sampling rate f. Unless stated otherwise, our inputs and outputs all are represented in this discretized form.

We use subscript t to indicate the time stamp. For instance, the current video frame is I_t, the current vehicle's speed is V_t, the kth previous video frame is I_{t-k}, and the kth previous steering angle is S_{t-k}, etc. Then, the k recent samples can be denoted by $\mathbf{V}_{[t-k+1,t]} \equiv \langle V_{t-k+1},...,V_t \rangle$, $\mathbf{S}_{[t-k+1,t]} \equiv \langle S_{t-k+1},...,S_t \rangle$ and $\mathbf{V}_{[t-k+1,t]} \equiv \langle V_{t-k+1},...,V_t \rangle$, respectively. Our goal is to train a deep network that predicts desired driving actions from the vehicle's historical states, historical and current visual observations, and the planned route. The learning task can be defined as:

$$F : (\mathcal{S}_{[t-k+1,t]}, \mathcal{V}_{[t-k+1,t]}, \mathcal{L}_{[t-k+1,t]}, \mathcal{I}_{[t-k+1,t]}, P_t) \rightarrow \mathcal{S}_{t+1} \times \mathcal{V}_{t+1} \qquad (1)$$

where $\mathcal{S}_{t+1}$ represents the steering angle space and $\mathcal{V}_{t+1}$ the speed space for future time $t+1$. $\mathcal{S}$ and $\mathcal{V}$ can be defined at several levels of granularity. We consider the continuous values directly recorded from the car's CAN bus, where $\mathcal{V} = \{V | 0 \leq V \leq 180$ for speed and $\mathcal{S} = \{S | -720 \leq S \leq 720\}$ for steering angle. Here, kilometer per hour (km/h) is the unit of V, and degree (°) the unit of S. Since there is not much to learn from the historical values of P, only P_t is used for the learning. P_t is either a video frame from our TomTom route planner or a 300×2 matrix from our OSM route planner.

Given N training samples collected during real drives, learning to predict the driving actions for the future time $t + 1$ is based on minimizing the following cost:

$$L(\theta) = \sum_{n=1}^{N} \Big(l(S_{t+1}^n, F_s(\mathbf{S}_{[t-k+1,t]}^n, \mathbf{V}_{[t-k+1,t]}^n, \mathbf{L}_{[t-k+1,t]}^n, \mathbf{I}_{[t-k+1,t]}^n, P_t)) \\ + \lambda l(V_{t+1}^n, F_v(\mathbf{S}_{[t-k+1,t]}^n, \mathbf{V}_{[t-k+1,t]}^n, \mathbf{L}_{[t-k+1,t]}^n, \mathbf{I}_{[t-k+1,t]}^n, P_t)) \Big), \qquad (2)$$

where λ is a parameter balancing the two losses, one for steering angle and the other for speed. We use $\lambda = 1$ in this work. F is the learned function for the driving model. For the continuous regression task, $l(.)$ is the $L2$ loss function. Finding a better way to balance the two loss functions constitutes our future work. Our model learns from multiple previous frames in order to better understand traffic dynamics.

4.2 Implementation

Our driving system is trained with four cameras (front, left, right, and rear view), which provide a full panoramic view already. We recorded the data with all eight cameras in order to keep future flexibility.

This work develops a customized network architecture for our learning problem defined in Section 4.1, which consists of deep hierarchical sub-networks. It comes with multiple CNNs as feature encoders, four LSTMs as temporal encoders for information from the four surround-view cameras, a fully-connected network (FN) to fuse information from all cameras and the map, and finally two FNs to output future speed and steering angle of the car. The illustrative architecture is show in Figure 1.

During training, videos are all resized to 256×256 and we augment our data by using 227×227 crops, without mirroring. For the CNN feature encoder, we take ResNet34 [77] model pre-trained on the ImageNet [78] dataset. Our network architecture is inspired by the Long-term Recurrent Convolutional Network developed in [79]. A more detailed description about the network architecture is provided in the supplementary material.

Table 2: MSE of speed prediction and steering angle prediction when a single front-facing camera is used (previous driving states are given).

	CAN-only	[25]	[12]	Ours
Steering	0.869	1.312	0.161	**0.134**
Speed	0.0147	0.6533	0.0066	**0.0030**

5   Experiments

We train our models on 80% of our dataset, corresponding to 48 hours of driving time and around 1.7 million unique synchronized sequence samples. Our driving routes are normally 2 hours long. We have selected 24 out of the 30 driving routes for training, and the other 6 for testing. This way, the network would not overfit to any type of specific road or weather. Synchronized video frames are extracted at a rate of 10 fps, as 60 fps will generate a very large dataset. A synchronized sample contains four frames at a resolution of 256×256 for the corresponding front, left, right and rear facing cameras, a rendered image at 256×256 pixels for TomTom route planner or a 300×2 matrix for OSM route planner, CAN bus data and the GPS data of the the 'past'.

We train our models using the Adam Optimizer with an initial learning rate of 10^{-4} and a batch size of 16 for 5 epochs, resulting in a training time of around 3 days. For the four surround-view cameras, we have used four frames to train the network: 0.9s in the past, 0.6s in the past, 0.3s in the past, and the current frame. This leads to a sampling rate of $f = 3.33$. A higher value can be used at the price of computational cost. This leads to $4 \times 4 = 16$ CNNs for capturing street-view visual scene.

We structure our evaluation into two parts: evaluating our method against existing methods, and evaluating the benefits of using a route planner and/or a surround-view camera system.

5.1   Comparison to other single-camera methods

We compare our method to the method of [12] and [25]. Since BDDV dataset does not provide data for driving actions (e.g. steering angle) [12], we train their networks on our dataset and compare with our method directly. For a fair comparison, we follow their settings, by only using a single front-facing camera and predicting the driving actions for the future time at $0.3s$.

We use the mean squared error (MSE) for evaluation. The results for speed prediction and steering angle prediction are shown in Table 2. We include a baseline reference of only training on CAN bus information (no image information given). The table shows that our method outperforms [25] significantly and is slightly better than [12]. [25] does not use a pre-trained CNN; this probably explains why their performance is a lot worse. The comparison to these two methods is to verify that our frontal-view driving model represents the state of the art so that the extension is made to a sensible basis to include multiple-view cameras and to include route planners.

We note that the baseline reference performs quite well, suggesting that due to the inertia of driving maneuvers, the network can already predict speed and steering angle of $0.3s$ further into the future quite well, solely based on the supplied ground truth maneuver of the past. For instance, if one steers the wheels to the right at time t, then at

Cameras	Route planner	Full dataset		Subset: GT $\leq$ 30 km/h	
		Steering	Speed	Steering	Speed
	None	0.967	0.197	4.053	0.167
Front-view	TomTom	0.808	**0.176**	3.357	0.268
	OSM	0.981	0.212	4.087	0.165
	None	0.927	0.257	3.870	**0.114**
Surround-view	TomTom	**0.799**	0.200	**3.214**	0.142
	OSM	0.940	0.228	3.917	0.125

Table 3: MSE (smaller=better) of speed and steering angle prediction by our method, when different settings are used. Predictions on full evaluation set and the subset with human driving maneuver $\leq$ 30 km/h.

$t + 0.3s$ the wheels are very likely to be at a similar angle to the right. In a true autonomous vehicle the past driving states might not be always correct. Therefore, we argue that the policy employed by some existing methods by relying on the past 'ground-truth' states of the vehicle should be used with caution. For the real autonomous cars, the errors will be exaggerated via a feedback loop. Based on this finding, we remove $\mathcal{S}_{[t-k+1,t]}$ and $\mathcal{V}_{[t-k+1,t]}$, i.e. without using the previous human driving maneuvers, and learn the desired speed and steering angle only based on the planned route, and the visual observations of the local road situation. This new setting 'forces' the network to learn knowledge from route planners and road situations.

5.2 Benefits of Route Planners

We evaluate the benefit of a route planner by designing two networks using either our visual TomTom, or our numerical OSM guidance systems, and compare these against our network that does not incorporate a route planner. The results of each networks speed and steering angle prediction are summarized in Table 3. The evaluation shows that our visual TomTom route planner significantly improves prediction performance, while the OSM approach does not yield a clear improvement. Since, the prediction of speed is easier than the prediction of steering angle, using a route planner will have a more noticeable benefit on the prediction of steering angles.

Why the visual TomTom planner is better? It is easy to think that GPS coordinates contain more accurate information than the rendered videos do, and thus provide a better representation for planned routes. This is, however, not case if the GPS coordinates are used directly without further, careful, processing. The visualization of a planned route on navigation devices such as TomTom Mobile Go makes use of accurate vehicle localization based on vehicle's moving trajectories to provide accurate procedural knowledge of the routes along the driving direction. The localization based on vehicle's moving trajectories is tackled under the name *map-matching*, and this, in itself, is a long-standing research problem [80,81,82]. For our TomTom route planner, this is done with TomTom's excellent underlying *map-matching* method, which is unknown to the public though. This rendering process converts the 'raw' GPS coordinates into a more structural representation. Our implemented OSM route planner, however, encodes more

of a global spatial information at a map level, making the integration of navigation information and street-view videos more challenging. Readers are referred to Figure 3 for exemplar representations of the two route planners.

In addition to *map-matching*, we provide further possible explanations: **1)** raw GPS coordinates are accurate for locations, but fall short of other high-level and contextual information (road layouts, road attributes, etc.) which is 'visible' in the visual route planner. For example, raw GPS coordinates do not distinguish 'highway exit' from 'slight right bend' and do not reveal other alternative roads in an intersection, while the visual route planner does. It seems that those semantic features optimized in navigation devices to assist human driving are useful for machine driving as well. Feature designing/extraction for the navigation task of autonomous driving is an interesting future topic. **2)** The quality of underlying road networks are different from TomTom to OSM. OSM is crowdsourced, so the quality/accuracy of its road networks is not always guaranteed. It is hard to make a direct comparison though, as TomTom's road networks are inaccessible to the public.

5.3   Benefits of Surround-View Cameras

Surround-view cameras offer a modest improvement for predicting steering angle on the full evaluation set. They, however, appear to reduce the overall performance for speed prediction. Further investigation has shown that surround-view cameras are especially useful for situations where the ego-car is required to give the right of way to other (potential) road users by controlling driving speed. Notable examples include 1) busy city streets and residential areas where the human drives at low velocity; and 2) intersections, especially those without traffic lights and stop signs. For instance, the speed at an intersection is determined by whether the ego-car has a clear path for the planned route. Surround-view cameras can see if other cars are coming from any side, whereas a front camera only is blind to many directions. In order to examine this, we have explicitly selected two specific types of scenes across our evaluation dataset for a more fine-grained evaluation of front-view vs. surround-view: 1) low-speed (city) driving according to the speed of human driving; and 2) intersection scenarios by human annotation. The evaluation results are shown in Table 3 and Table 4, respectively. The better-performing TomTom route planner models are used for the experiments in Table 4. Surround-view cameras significantly improve the performance of speed control in these two very important driving situations. For 'high-speed' driving on highway or countryside road, surround-view cameras do not show clear advantages, in line with human driving – human drivers also consult non-frontal views less frequently for high-speed driving.

As a human driver, we consult our navigation system mostly when it comes to multiple choices of road, namely at road intersections. To evaluate whether route planning improves performance specifically in these scenarios, we select a subset of our test set for examples with a low speed by human, and report the results in this subset also in Table 3. Results in Table 3 supports our claim that route planning is beneficial to a driving model, and improves the driving performance especially for situations where a turning maneuver is performed. In future work, we plan to select other interesting situations for more detailed evaluation.

Cameras	≤ 10 km/h	≤ 20 km/h	≤ 30 km/h	≤ 40 km/h	≤ 50 km/h
Front-view	0.118	0.150	0.158	0.157	0.148
Surround-view	**0.080**	**0.127**	**0.145**	**0.146**	**0.143**

Table 4: MSE (smaller=better) of speed prediction by our Front-view+TomTom and Surround-view+TomTom driving models. Evaluated on manually annotated intersection scenarios over a 2-hour subset of our evaluation dataset. Surround-view significantly outperforms front-view in intersection situations.

Qualitative Evaluation While standard evaluation techniques for neural networks such as mean squared error, do offer global insight into the performance of models, they are less intuitive in evaluating where, at a local scale, using surround view cameras or route planning improves prediction accuracy. To this end, we use our visualization tool to inspect and evaluate the model performances for different 'situations'.

Figure 3 shows examples of three model comparisons (TomTom, Surround, Surround+TomTom) row-wise, wherein the model with additional information is directly compared to our front-camera-only model, shown by the speed and steering wheel angle gauges. The steering wheel angle gauge is a direct map of the steering wheel angle to degrees, whereas the speed gauge is from 0km/h to 130km/h. Additional information a model might receive is 'image framed' by the respective color. Gauges should be used for relative model comparison, with the front-camera-only model prediction in orange, model with additional information in red and human maneuver in blue. Thus, for our purposes, we define a well performing model when the magnitude of a model gauge is identical (or similar) to the human gauge. Column-wise we show examples where: (a) both models perform well, (b) model with additional information outperforms , (c) both models fail.

Our qualitative results, in Figure 3 (1,b) and (3,b), support our hypothesis that a route planner is indeed useful at intersections where there is an ambiguity with regards to the correct direction of travel. Both models with route planning information are able to predict the correct direction at the intersection, whereas the model without this information predicts the opposite. While this 'wrong' prediction may be a valid driving maneuver in terms of safety, it nonetheless is not correct in terms of arriving at the correct destination. Our map model on the other hand is able to overcome this. Figure 3 (2,b) shows that surround-view cameras are beneficial at predicting the correct speed. The frontal view supplied could suggest that one is on a country road where the speed limit is significantly higher than in the city, as such, our front-camera-only model predicts a speed much greater than the human maneuver. However, our surround-view system can pick up on the pedestrians on the right of the car, thus adjusts the speed accordingly. The surround-view model thus has a more precise understanding of its surroundings.

Visualization tool. To obtain further insights into where current driving models perform well or fail, we have developed a visual evaluation tool that lets users select scenes in the evaluation set by clicking on a map, and then rendering the corresponding 4 camera views, the ground truth and predicted vehicle maneuver (steering angle and speed) along with the map at that point in time. These evaluation tools along with the dataset will be released to the public. In particular, visual evaluation is extremely helpful to understand
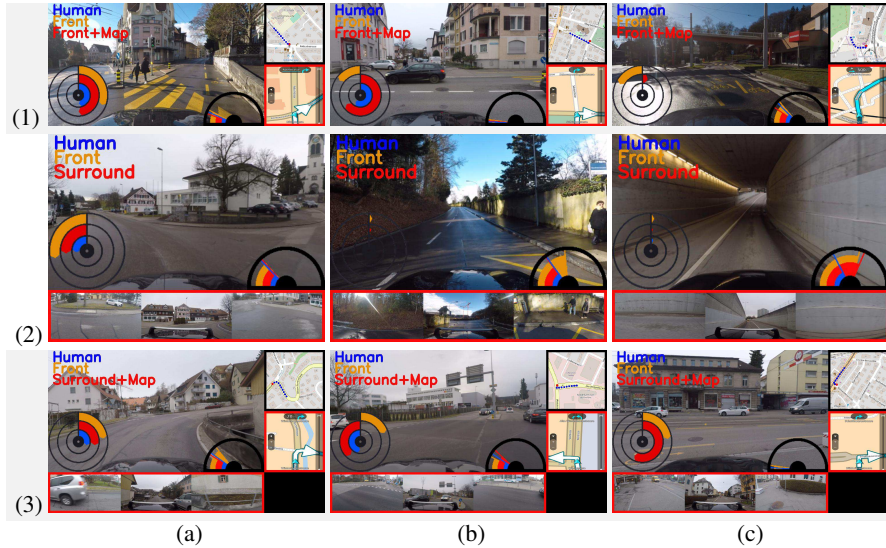
Fig. 3: Qualitative results for future driving action prediction, to compare three cases to the front camera-only-model: (1) learning with TomTom route planner, (2) learning with surround-view cameras (3) learning with TomTom route planner and surround-view cameras. TomTom route planer and surround-view images shown in red box, while OSM route planner in black box. Better seen on screen.

where and why a driving model predicted a certain maneuver, as sometimes, while not coinciding with the human action, the network may still predict a safe driving maneuver.

6   Conclusion

In this work, we have extended learning end-to-end driving models to a more realistic setting from only using a single front-view camera. We have presented a novel task of learning end-to-end driving models with surround-view cameras and rendered maps, enabling the car to 'look' to side, rearward, and to 'check' the driving direction. We have presented two main contributions: 1) a new driving dataset, featuring 60 hours of driving videos with eight surround-view cameras, low-level driving maneuvers recorded via car's CAN bus, two representations of planned routes by two route planners, and GPS-IMU data for the vehicle's odometry; 2) a novel deep network to map directly from the sensor inputs to future driving maneuvers. Our data features high temporary resolution and 360 degree view coverage, frame-wise synchronization, and diverse road conditions, making it ideal for learning end-to-end driving models. Our experiments have shown that an end-to-end learning method can effectively use surround-view cameras and route planners. The rendered videos outperforms a stack of raw GPS coordinates for representing planned routes.

References

1. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
2. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision (2018)
3. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: A diverse driving video database with scalable annotation tooling. CoRR (2018)
4. Dai, D., Van Gool, L.: Progressive model adaptation and knowledge transfer from daytime to nighttime for semantic road scene understanding. In: IEEE International Conference on Intelligent Transportation Systems. (2018)
5. LeCun, Y., Muller, U., Ben, J., Cosatto, E., Flepp, B.: Off-road obstacle avoidance through end-to-end learning. In: NIPS. (2005)
6. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F., Coates, A., Ng, A.Y.: An empirical evaluation of deep learning on highway driving. CoRR **abs/1504.01716** (2015)
7. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2722–2730
8. Caltagirone, L., Bellone, M., Svensson, L., Wahde, M.: Simultaneous perception and path generation using fully convolutional neural networks. arXiv preprint arXiv:1703.08987 (2017)
9. Chen, S., Zhang, S., Shang, J., Chen, B., Zheng, N.: Brain inspired cognitive model with attention for self-driving cars. arXiv preprint arXiv:1702.05596 (2017)
10. Paxton, C., Raman, V., Hager, G.D., Kobilarov, M.: Combining neural networks and tree search for task and motion planning in challenging environments. In: IROS. (2017)
11. Pendleton, S.D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y.H., Rus, D., Ang, M.H.: Perception, Planning, Control, and Coordination for Autonomous Vehicles. Machines **5**(1) (2017)
12. Huazhe Xu, Yang Gao, F.Y., Darrell, T.: End-to-end learning of driving models from large-scale video datasets. In: Computer Vision and Pattern Recognition (CVPR). (2017)
13. Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. (2018)
14. Chen, Y., Wang, J., Li, J., Lu, C., Luo, Z., Xue, H., Wang, C.: Lidar-video driving dataset: Learning driving policies effectively. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
15. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
16. McAllister, R., Gal, Y., Kendall, A., Van Der Wilk, M., Shah, A., Cipolla, R., Weller, A.: Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In: International Joint Conference on Artificial Intelligence. (2017)
17. Hecker, S., Dai, D., Van Gool, L.: Failure prediction for autonomous driving models. In: IEEE Intelligent Vehicles Symposium (IV). (2018)
18. The Automobile (weekly), Thursday, December 27 (1906)
19. Tolman, E.C.: Cognitive maps in rats and men. Psychological Review **55** (1948) 189–208
20. Maguire, E.A., Burgess, N., Donnett, J.G., Frackowiak, R.S.J., Frith, C.D., O'Keefe, J.: Knowing where and getting there: A human navigation network. Science **280**(5365) (1998) 921–924

21. Urmson, C., Anhalt, J., Bae, H., Bagnell, J.A.D., Baker, C.R., Bittner, R.E., Brown, T., Clark, M.N., Darms, M., Demitrish, D., Dolan, J.M., Duggins, D., Ferguson, D., Galatali, T., Geyer, C.M., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Likhachev, M., Litkouhi, B., Kelly, A., McNaughton, M., Miller, N., Nickolaou, J., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Sadekar, V., Salesky, B., Seo, Y.W., Singh, S., Snider, J.M., Struble, J.C., Stentz, A.T., Taylor, M., Whittaker, W.R.L., Wolkowicki, Z., Zhang, W., Ziglar, J.: Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I **25**(8) (June 2008) 425–466

22. Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V., Sokolsky, M., Stanek, G., Stavens, D., Teichman, A., Werling, M., Thrun, S.: Towards fully autonomous driving: Systems and algorithms. In: IEEE Intelligent Vehicles Symposium (IV). (2011)

23. Luettel, T., Himmelsbach, M., Wuensche, H.J.: Autonomous ground vehicles—concepts and a path to the future. Proceedings of the IEEE **100** (2012) 1831–1839

24. Pomerleau, D.A.: Nips. (1989)

25. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)

26. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11) (2013) 1231–1237

27. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: CVPR. (2017)

28. Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R.: 3d traffic scene understanding from movable platforms. IEEE transactions on pattern analysis and machine intelligence **36**(5) (2014) 1012–1025

29. Anderson, James M., N.K.K.D.S.P.S.C.S., Oluwatola, T.A.: Autonomous Vehicle Technology: A Guide for Policymakers. Santa Monica, CA: RAND Corporation (2016)

30. Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L.D., Muller, U.: Explaining how a deep neural network trained with end-to-end learning steers a car. CoRR (2017)

31. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540) (2015) 529–533

32. Shalev-Shwartz, S., Shammah, S., Shashua, A.: Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295 (2016)

33. : Deep reinforcement learning framework for autonomous driving. Electronic Imaging (2017)

34. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics conference. (2017)

35. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. (2017) 1–16

36. Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., Borrelli, F.: Automated driving: The role of forecasts and uncertainty—a control perspective. European Journal of Control **24** (2015) 14–32

37. Shia, V.A., Gao, Y., Vasudevan, R., Campbell, K.D., Lin, T., Borrelli, F., Bajcsy, R.: Semiautonomous vehicular control using driver modeling. IEEE Transactions on Intelligent Transportation Systems **15**(6) (2014) 2696–2709

38. Kasper, D., Weidl, G., Dang, T., Breuel, G., Tamke, A., Wedel, A., Rosenstiel, W.: Object-oriented bayesian networks for detection of lane change maneuvers. IEEE Intelligent Transportation Systems Magazine **4**(3) (2012) 19–31

39. Jain, A., Koppula, H.S., Raghavan, B., Soh, S., Saxena, A.: Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3182–3190

40. Liu, Y.C., Lin, K.Y., Chen, Y.S.: Bird's-eye view vision system for vehicle surrounding monitoring. In: International Conference on Robot Vision. (2008)

41. Dueholm, J.V., Kristoffersen, M.S., Satzoda, R.K., Moeslund, T.B., Trivedi, M.M.: Trajectories and maneuvers of surrounding vehicles with panoramic camera arrays. IEEE Transactions on Intelligent Vehicles **1**(2) (2016) 203–214

42. Khosroshahi, A., Ohn-Bar, E., Trivedi, M.M.: Surround vehicles trajectory analysis with recurrent neural networks. In: International Conference on Intelligent Transportation Systems (ITSC). (2016) 2267–2272

43. Bertozzi, M., Castangia, L., Cattani, S., Prioletti, A., Versari, P.: 360° detection and tracking algorithm of both pedestrian and vehicle using fisheye images. In: IEEE Intelligent Vehicles Symposium (IV). (2015) 132–137

44. Dueholm, J.V., Kristoffersen, M.S., Satzoda, R.K., Ohn-Bar, E., Moeslund, T.B., Trivedi, M.M.: Multi-perspective vehicle detection and tracking: Challenges, dataset, and metrics. In: International Conference on Intelligent Transportation Systems (ITSC). (2016)

45. Kum, C.H., Cho, D.C., Ra, M.S., Kim, W.Y.: Lane detection system with around view monitoring for intelligent vehicle. In: International SoC Design Conference (ISOCC). (2013) 215–218

46. Yu, M., Ma, G.: A visual parking guidance for surround view monitoring system. In: IEEE Intelligent Vehicles Symposium (IV). (2015)

47. Gao, Y., Lin, C., Zhao, Y., Wang, X., Wei, S., Huang, Q.: 3-d surround view for advanced driver assistance systems. IEEE Transactions on Intelligent Transportation Systems **19**(1) (2018) 320–328

48. Chen, A., Ramanandan, A., Farrell, J.A.: High-precision lane-level road map building for vehicle navigation. In: IEEE/ION Position, Location and Navigation Symposium. (2010)

49. Schindler, A., Maier, G., Janda, F.: Generation of high precision digital maps using circular arc splines. In: IEEE Intelligent Vehicles Symposium. (2012)

50. Rizaldi, A., Althoff, M.: Formalising traffic rules for accountability of autonomous vehicles. In: International Conference on Intelligent Transportation Systems. (2015)

51. Bender, P., Ziegler, J., Stiller, C.: Lanelets: Efficient map representation for autonomous driving. In: IEEE Intelligent Vehicles Symposium. (2014)

52. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2011) 316–324

53. Bast, H., Delling, D., Goldberg, A.V., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F.: Route planning in transportation networks. In: Algorithm Engineering - Selected Results and Surveys. (2016) 19–80

54. Zheng, Y.T., Yan, S., Zha, Z.J., Li, Y., Zhou, X., Chua, T.S., Jain, R.: Gpsview: A scenic driving route planner. ACM Trans. Multimedia Comput. Commun. Appl. **9**(1) (2013) 3:1–3:18

55. Chen, C., Zhang, D., Guo, B., Ma, X., Pan, G., Wu, Z.: Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Transactions on Intelligent Transportation Systems **16**(3) (2015) 1259–1273

56. Yang, B., Guo, C., Ma, Y., Jensen, C.S.: Toward personalized, context-aware routing. The VLDB Journal **24**(2) (2015) 297–318

57. Haklay, M., Weber, P.: Openstreetmap: User-generated street maps. IEEE Pervasive Computing **7**(4) (2008) 12–18
58. Mattern, N., Schubert, R., Wanielik, G.: High-accurate vehicle localization using digital maps and coherency images. In: IEEE Intelligent Vehicles Symposium. (2010)
59. Tao, Z., Bonnifait, P., Frémont, V., Ibañez-Guzman, J.: Mapping and localization using gps, lane markings and proprioceptive sensors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. (2013)
60. Nedevschi, S., Popescu, V., Danescu, R., Marita, T., Oniga, F.: Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map. IEEE Transactions on Intelligent Transportation Systems **14**(2) (2013) 673–687
61. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Mapnet: Geometry-aware learning of maps for camera localization. CoRR **abs/1712.03342** (2017)
62. Bar Hillel, A., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: A survey. Mach. Vision Appl. **25**(3) (April 2014) 727–745
63. Jensen, M.B., Philipsen, M.P., Møgelmose, A., Moeslund, T.B., Trivedi, M.M.: Vision for looking at traffic lights: Issues, survey, and perspectives. IEEE Transactions on Intelligent Transportation Systems **17**(7) (July 2016) 1800–1815
64. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). (2015)
65. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
66. Sivaraman, S., Trivedi, M.M.: Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Transactions on Intelligent Transportation Systems **14**(4) (2013) 1773–1795
67. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. In: European Conference on Computer Vision (ECCV). (2016)
68. Manen, S., Gygli, M., Dai, D., Van Gool, L.: Pathtrack: Fast trajectory annotation with path supervision. In: International Conference on Computer Vision (ICCV). (2017)
69. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. arXiv preprint arXiv:1704.05519 (2017)
70. Ohn-Bar, E., Trivedi, M.M.: Looking at humans in the age of self-driving and highly automated vehicles. IEEE Transactions on Intelligent Vehicles **1**(1) (2016) 90–104
71. TomTom GO Mobile App: `https://itunes.apple.com/us/app/tomtom-go-mobile/id884963367?mt=8`. (Accessed from 2017-10 to 2018-03)
72. Hentschel, M., Wagner, B.: Autonomous robot navigation based on openstreetmap geodata. In: IEEE Conference on Intelligent Transportation Systems. (2010)
73. Luxen, D., Vetter, C.: Real-time routing with openstreetmap data. In: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. (2011)
74. Ma, X., Andréasson, I.: Estimation of driver reaction time from car-following data: Application in evaluation of general motor-type model. Transportation Research Record: Journal of the Transportation Research Board **1965** (2006) 130–141
75. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research **36**(1) (2017) 3–15
76. : Udacity: Public driving dataset. https://www. udacity.com/self-driving-car (2017)
77. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
78. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)

79. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR. (2015)
80. Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y.: Map-matching for low-sampling-rate gps trajectories. In: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. (2009) 352–361
81. Yuan, J., Zheng, Y., Zhang, C., Xie, X., Sun, G.Z.: An interactive-voting based map matching algorithm. In: International Conference on Mobile Data Management. (2010)
82. Chen, B.Y., Yuan, H., Li, Q., Lam, W.H.K., Shaw, S.L., Yan, K.: Map-matching algorithm for large-scale low-frequency floating car data. Int. J. Geogr. Inf. Sci. **28**(1) (2014)