SN Computer Science (2021) 2:375
https://doi.org/10.1007/s42979-021-00759-6

ORIGINAL RESEARCH

Collision Avoidance Using Deep Learning-Based Monocular Vision

Róbert-Adrian Rill[1,2] · Kinga Bettina Faragó[1]

Received: 4 December 2020 / Accepted: 22 June 2021 / Published online: 10 July 2021

Abstract
Autonomous driving technologies, including monocular vision-based approaches, are in the forefront of industrial and research communities, since they are expected to have a significant impact on economy and society. However, they have limitations in terms of crash avoidance because of the rarity of labeled data for collisions in everyday traffic, as well as due to the complexity of driving situations. In this work, we propose a simple method based solely on monocular vision to overcome the data scarcity problem and to promote forward collision avoidance systems. We exploit state-of-the-art deep learning-based optical flow and monocular depth estimation methods, as well as object detection to estimate the speed of the ego-vehicle and to identify the lead vehicle, respectively. The proposed method utilizes car stop situations as collision surrogates to obtain data for time to collision estimation. We evaluate this approach on our own driving videos, collected using a spherical camera and smart glasses. Our results indicate that similar accuracy can be achieved on both video sources: the external road view from the car's, and the ego-centric view from the driver's perspective. Additionally, we set forth the possibility of using spherical cameras as opposed to traditional cameras for vision-based automotive sensing.

Keywords Monocular vision · Time to collision · Deep learning · Spherical camera · Ego-centric video

Introduction

Autonomous driving, self-driving car, driverless car, highly automated vehicle, autonomous vehicle, intelligent vehicle, advanced driver assistance system, intelligent transportation system—these are popular topics in the forefront of the industrial and research communities. Automated driving raises novel challenges, since modern cars consist of a number of complex systems with increasing processing and communication requirements [5, 30]. The technological, scientific and engineering advances make machine learning solutions attractive in the automotive industry [37]. As elaborated in the 2013 "blue paper" of the Research Division of Morgan Stanley, highly automated vehicle technologies are expected to have a significant economical and societal impact [32]. However, they will have limitations in terms of crash avoidance [30]. The main reasons are the rarity of collisions in everyday traffic [3, 24, 35] and the complexity of the driving situations to be considered when designing collision avoidance systems. In addition, traffic accidents represent a major source of injuries and fatalities [10], with primary causes being human errors and inadequate driver states [18]. Therefore, contributions towards reliable, robust and real-time driver assistance systems—including collision avoidance systems—are highly desired [24, 35]. These safety systems should perceive the surrounding environment and warn the driver about dangerous situations or take proactive steps to prevent accidents before they would occur.

One important step is the optimal selection of vehicle sensors in terms of cost, range parameters and reliability [18]. The most common automotive sensing technologies are Radio Detection and Ranging (RADAR) and Light Detection and Ranging (LiDAR) [1, 16, 18], which use radio frequency or laser signals, respectively. RADAR sensors are more robust to weather conditions, but have lower accuracy being exposed to interference issues of the dynamic and

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013 and EFOP-3.6.3-VEKOP-16-2017-00002)

✉ Róbert-Adrian Rill
rillroberto88@yahoo.com

Kinga Bettina Faragó
faragokinga@inf.elte.hu

[1] Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

[2] Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania

noisy environment. LiDAR based systems can identify high-resolution details of the 3D environment, but are costly to produce and maintain as they require large processing power.

In recent years vision-based techniques for road safety improvement, and more generally for autonomous driving, have gained increased attention [18]. The main advantages of optical sensors are low cost, robustness to dynamic and crowded traffic scenarios and they are free from interference problems. Off-the-shelf cameras provide high-quality data, and increased computing power is accessible due to graphical processing units and multicore processors. Moreover, complementing the traditional sensors with vision-based alternatives, solving the same task in different ways and combining the results improves the overall reliability of a system.

The main contribution of the present work consists in the proposition of a simple approach for time-to-collision (TTC) estimation based solely on monocular vision algorithms, to promote forward collision avoidance systems and to overcome the data deficiency problem. We leverage the results of deep learning-based computer vision algorithms to construct the prediction features, and evaluate the TTC estimation method on our own real driving videos collected using an off-the-shelf spherical camera and smart glasses worn by the driver. We show that similar performance can be achieved on both the external road view and the driver's ego-centric view videos. As an additional contribution, we set forth the potential usage of spherical cameras as opposed to classical cameras for vision-based automotive sensing, by discussing briefly their benefits.

The paper is organized as follows. "Related Work" provides a background for the present work by reviewing previous studies that use monocular vision for autonomous driving and for collision avoidance. "Materials and Methods" presents briefly the state-of-the-art deep learning-based algorithms that we employ in this work, outlines the method used for ego-speed estimation ("Speed Estimation"), describes the data collection procedure ("Driving Data Collection") and presents the details of the proposed TTC estimation approach ("TTC Estimation"). In "Results" we report the results of our quantitative analysis, and in "Discussion" examine the implications and limitations of our approach, together with discussing the benefits of spherical cameras for autonomous driving. Finally, "Conclusion" concludes the paper.

Related Work

Various works are concerned with monocular vision-based approaches to solve the autonomous driving problem. The direct perception paradigm [8] consists in mapping the input image to a few key affordance indicators (e.g., angle of the car relative to the road, distance from lane markings and surrounding cars) that are used by a controller to make driving decisions. Direct perception was preceded by: (i) mediated perception involving a number of sub-components creating a consistent representation of the car's surroundings to enable an AI engine to make decisions; and (ii) behavior reflex, also called end-to-end approaches, that map the input image directly to driving actions. For more details see [8] and the references therein. The direct perception approach was improved in [4] using a larger convolutional neural network architecture and less affordance indicators (5 instead of 14). A different approach was investigated by Kendall et al. [15], who applied for the first time deep reinforcement learning to learn to drive a real car with 30 min of exploration using only on-board computation. An end-to-end method was investigated by Bojarski et al. [6] who used driving videos of less than 100 hours to train a car to operate in diverse environments and conditions. Chen et al. [7] took a more systematic end-to-end approach and designed a novel network structure that uses auxiliary tasks (semantic segmentation, transfer learning from image recognition, optical flow, and fusion of temporal information and vehicle kinematics) to boost driving performance.

The works listed above achieve impressive results in autonomous driving based on monocular vision, but they have limitations. Even though driving a car requires only manipulating the direction and the speed [8], most of the works train networks to learn only steering angle, some using the speed as input. Also, evaluations were performed in simulation or restricted environments, whereas real-world driving can involve many complex and unique situations. This is especially important when considering accident prevention, since collisions occur very rarely in everyday traffic [3, 24, 35]. Accordingly, such learning-based approaches like the ones above are not feasible in this case.

Still a number of works use monocular vision-based approaches to improve collision avoidance systems of automated vehicles. Although critical situations can be identified and geographically located to reveal high-risk road sections from smartphone GPS and motion sensor data [3], danger always depends on current driving conditions and circumstances. A safety system should monitor the environment in real time and warn the driver or react before a crash may happen.

Studies investigating collision avoidance using images from a single camera typically estimate time-to-collision (TTC), which in turn addresses the collision rarity problem as well. In a relatively early work, Dagan et al. [10] claimed that even short advance warnings could significantly reduce the number and severity of collisions. They estimated TTC and collision course from the size and position of vehicles on images using a special camera setup in a test area, and

observed errors below 2 s with noise increasing for larger TTC.

TTC can be inferred from obstacle scale change with sparse feature detection and tracking too. For instance in [19] TTC was computed based on the rate change of a geometric invariant called intrinsic scale, and results were compared to those of a laser ranging device. Or in [25] an approach was presented for TTC and collision risk estimation in case of semi-rigid obstacles using videos of pedestrians captured in a controlled environment. However, such methods based on feature tracking and matching are not suitable for efficient hardware implementation [33].

Kilicarslan and Zheng [16] computed TTC from motion divergence to detect approaching targets, avoiding object recognition and depth sensing, and tested their approach in various environments. The motion profiling used by the authors captures object motion, ignores most background objects, is more stable than optical flow, but is error-prone due to fake motion from shadows and reflections, camera shaking on uneven roads and camera tilting in vehicle breaking situations. In a closely related work, Shi et al. [33] proposed a hardware-friendly TTC estimation method from the divergence of dense optical flow fields. They used random forests to compute optical flow from motion energy features. Although their method is well-suited for low-cost real-time embedded systems deployed on smart video sensors, evaluation was limited to synthetic looming image sequences, one real-world sequence with a stationary indoor object and a single driving video clip.

Patra et al. [22] presented a forward collision warning application with smartphones using license plate recognition, but their method requires inter-vehicular communication and is limited to urban speeds and close range distance between vehicles. Wulfe et al. [35] investigated the real-time prediction of collision risk using simulations to help mitigate issues resulting from the rarity of collisions and the need for large amounts of high-dimensional data. The authors formulated risk estimation as policy evaluation within the Markov decision process framework. Phillips et al. [24] predicted collision risk over a 10 s time horizon using a low-cost open-source modular system including object detection and tracking, and state (relative distance and velocity to the ego-vehicle) estimation. The authors also extended their framework to ego speed estimation, however, they make the assumption that dashed lane markings can be easily distinguished and their dimensions are known.

According to the above works, using TTC as a risk metric for collision avoidance systems is widely used. As opposed to previous studies, the uniqueness of our method for TTC estimation consists in the combination of:

– using speed estimated solely from monocular vision, as opposed to taking it as given input,
– making no assumptions about the environment (e.g., known lane marking dimensions),
– relying on object detection results due to its simplicity, and fusing it with monocular depth features, and
– evaluating the method on driving videos collected in a real environment.

Furthermore, we propose to use annotated car stop situations as collision surrogates to train a model offline, which can ultimately be fine-tuned, e.g., by inputs from the driver. We also show that a camera recording the view from the driver's perspective is just as useful for TTC estimation, as a device mounted on the car—after accounting for head motion. Using ordinary least squares linear regression, we achieve a Root Mean Square Error (RMSE) of close to 1 s on our driving videos.

Materials and Methods

In the following, we present briefly the deep learning-based computer vision algorithms that we employ in our work. We used optical flow and monocular depth to estimate the speed of the ego-vehicle, and we used features computed from object detection and monocular depth output for TTC estimation.

Optical Flow Estimation

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. The novel method called PWC-Net[1] [34] was designed according to the simple and well-established principles of using learnable feature pyramids, the warping operation as a layer to estimate large motions and the use of a cost volume layer. PWC-Net makes significant improvements in model size and accuracy over existing convolutional neural network models for optical flow. It is multiple times smaller in size and easier to train than previous models, and achieves state-of-the results on the KITTI 2015 benchmark [17]. In our experiments, we used the pre-trained "default network" provided in the Github repository. For more details about PWC-Net and other optical flow estimation approaches please see [34] and the references therein.

Monocular Depth Estimation

Monocular depth estimation aims to obtain a representation of the spatial structure of a scene by determining the

[1] https://github.com/sniklaus/pytorch-pwc.

distance of objects from a single image. MonoDepth[2] [13] 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 convolutional neural 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. It outperforms supervised methods on the KITTI 2015 benchmark [17]. In our experiments, we used the pre-trained "city2kitti" model provided in the Github repository. For more details about MonoDepth and related single-view depth estimation approaches please see [13] and the references therein.

Object Detection

You Only Look Once (YOLO)[3] [26] frames object detection as a regression problem and uses a single neural network to predict bounding boxes and object class probabilities from images in an end-to-end manner. This unified model has several benefits over traditional methods: it is fast, it reasons globally about the image by encoding contextual information and appearance about object classes, it is highly generalizable to new domains such as artwork. YOLOv2 [27] improves over the initial version by pooling ideas from past work with own novel concepts: batch normalization, higher resolution classification, using anchor boxes to predict bounding boxes, adjusting priors on bounding boxes, using a mechanism for jointly training on classification and detection data. YOLOv3 [28] further improves the previous version by using a larger neural network that presents increased accuracy while still maintaining real-time performance. In our experiments, we used the pre-trained weights for this latter version provided on the project website.

Speed Estimation

To estimate the speed of the ego-vehicle, we rely on our previous work [29], in which a simple and straightforward monocular vision-based approach was proposed. The method relies 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. In the present work, we leverage the PWC-Net and MonoDepth methods described above. For the sake of completeness, we repeat below the steps of the speed estimation pipeline.

1. Retrieve optical flow and monocular depth estimation results for a given image frame from a forward facing camera mounted on the car, recording the road and environment ahead.
2. Compute the scaled speed for the given frame: consider the magnitude of optical flow vectors and disparity values at valid pixels, and compute the quotient between their means. Valid pixels are obtained by imposing minimum threshold limits both for optical flow magnitude and disparity, in order to avoid outlier values.
3. Repeat the previous steps for all the frames from a video and apply temporal smoothing to reduce noise.
4. Repeat the previous step for multiple videos and estimate a scaling factor that minimizes the ratio between the ground truth and predicted speed. The scaling factor is used to convert the estimated speed from the image domain to real-world units.

With careful considerations the speed estimation method described above can achieve a RMSE of less than 1 m/s on the KITTI vision benchmark [11, 12], as evaluated in detail in our previous work [29].

Driving Data Collection

The results in this work are demonstrated on our self-collected driving dataset, which consists of 30 videos recorded in Hungary while traveling repeatedly on the 32 km long route between Budapest and Martonvásár, and back. Multiple videos were recorded on each drive and the length of the videos ranges between 10 and 30 min. Driving situations include both highway and urban traffic. The recordings were captured using two devices: a spherical camera[4] attached to the dashboard of the car recording at 30 frames per second (FPS) and 1920×1080 resolution, and the driver was wearing smart glasses[5] that records an ego-centric video (25 FPS, 1920×1080) and eye movement information (gaze direction, pupil diameter etc. at 50–100 Hz frequency). After synchronizing the two data sources, the total length of the recordings is over 10 h. For sample image frames from our driving videos see Fig. 1.

Recordings of the 360-degree camera have the advantage that different equirectangular projections can be extracted,

[2] https://github.com/mrharicot/monodepth.

[3] https://pjreddie.com/darknet/yolo/.

[4] Ricoh R Development Kit: https://ricohr.ricoh/en/.

[5] Tobii Pro Glasses 2: https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/.

Fig. 1 Sample image frames from our self-collected driving videos. The first two rows present wide equirectangular projections extracted from the spherical videos, with the detected lead vehicle highlighted. The third row contains images from the smart glasses recording the view of the driver, with detected mirror objects highlighted, as well as the lead vehicle relative to the mirrors

the resolution of which can match other driving datasets. For example, we extracted wide frontal projections similar to the recordings from the KITTI benchmark, to be able to apply the speed estimation approach described in the previous section. Moreover, rear projections can be used to monitor the driver and the passengers as well. Further advantages of spherical cameras are discussed in "Spherical Cameras for Automotive Sensing".

TTC Estimation

To estimate TTC we extract frontal projections from all of our spherical videos and employ a multistage process. First, we annotate by hand all the car stop situations, i.e. we mark and save the timepoints and frames when the ego-car stopped completely in traffic, using a video annotation tool [21]. As a result, we have a total of 134 such cases, which include stopping at red traffic lights, intersections and crosswalks to give priority to other vehicles or pedestrians, respectively. Second, we hand-pick the stop situations where there is a lead vehicle in front of the ego-car and for each case we consider the time interval before the annotated frame. Third, we estimate the ego-speed in the resulting video segments using the approach described in "Speed Estimation". Note that the videos were downsampled to 10 FPS to match the frequency of the recordings from the KITTI benchmark, on which the speed estimation was calibrated. The estimated speed is upsampled to the original 30 FPS by linearly interpolating the missing values. We drop situations where the estimated speed does not seem realistic after visual comparison with the videos. Fourth, we consider 10 s long time intervals to estimate TTC in case of a potential frontal collision with the lead car. TTC is defined for each moment (video frame) as the time remaining until the car stops if the current speed was kept constant. Calculation of TTC is possible since we hand-annotated the frames where the car stopped and have an estimation of the speed for every frame. Also, collision with the vehicle in front is assumed to occur at the hand-annotated frames. We further restrict the data for TTC prediction using the following thresholds.

- Minimum starting speed: we consider only the stopping situations where the speed at 10 s before the annotated stop timepoint is above 3 m/s.
- Minimum TTC and number of samples: when estimating TTC, we restrict the minimum to 1.0 s, and from the remaining data we randomly sample 150 datapoints for each situation.

The first threshold restricts the considered car stop situations and is used to avoid cases when the ego-car decelerated very slowly over 10 s. This results in 29 car stop situations used for evaluation. The second two thresholds limit the data used for TTC prediction from each situation considered: TTC has to be predicted in advance so that the driver assistance system has a possibility to react before collision; also TTC values are approximately the same for consecutive frames in our data. We note that other threshold values bring similar results to the ones presented in "Results", and does not influence the contributions and conclusions of the present work.

To evaluate TTC estimation we use linear regression and employ a *leave-one-situation-out* approach, where the model is trained on the data from 28 cases and tested on the 150 datapoints of the remaining situation. This process is repeated for all 29 possible combinations and the cross-validated RMSE error metrics are reported. We use the following set of independent variables and their subsets for prediction:

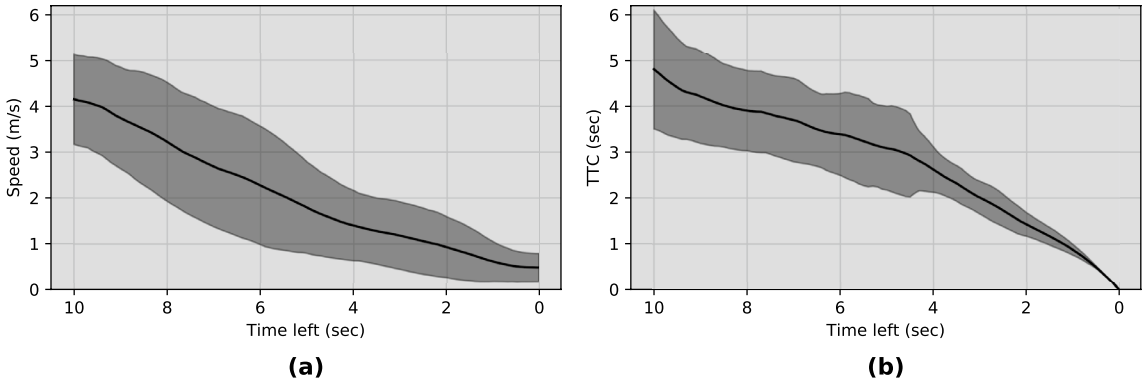

Fig. 2 Illustration of (**a**) estimated speed and (**b**) TTC in selected car stopping situations. Mean curves and range bands in terms of standard deviation are shown. The x-axis refers to the time remaining until assumed collision with lead vehicle

- speed: the estimated speed using the method presented in "Speed Estimation" and calibrated on the KITTI benchmark;
- BB-w: width of the bounding box of the lead vehicle from the equirectangular projections extracted from the spherical videos—for sample images and detected lead vehicles see Fig. 1 (the coordinates of missing bounding boxes are interpolated linearly using the available detections of the pretrained YOLOv3 model);
- MD: the median of the MonoDepth disparity values inside the bounding box of the lead vehicle;
- BB-w driver: width of the bounding box of the lead vehicle from the ego-centric videos—example images and lead vehicles are shown on Fig. 1.

To account for head motions in the videos of the driver's perspective, the lead vehicle is selected relative to the bounding boxes of the left-side wing and rear-view mirrors. To detect the mirrors we train a separate Tiny YOLOv3 [28] model on 200 images from our videos, hand-annotated with bounding boxes for the wing and rear-view mirror objects. The data was split into 175 train and 25 validation samples, and the model was trained for 160,000 iterations with a batch size of 24 and 8 subdivisions. For example images with detected mirrors see Fig. 1. When tested on 100 hand-annotated images from different videos than the training and validation images were selected from, the mirror detection model is highly accurate giving a perfect mean average precision of 1.0 (with no false positive nor false negative detections), and an average intersection over union of 0.91. This is not surprising since we detect the mirrors in a restricted environment. Although the model overfits our dataset, it is suitable for helping to identify the lead vehicle in the ego-centric videos.

Results

Figure 2 shows the average of the speed and TTC curves and their range in terms of standard deviation for the 29 car stop situations selected for quantitative evaluation. We can observe a linear decreasing pattern of the speed values over time, with the covered range also shrinking as the car gets closer to the moment of the assumed collision. A small variation is still present at 0 s, due to using estimated speed values that may contain errors. TTC curves also show

Table 1 Cross-validated RMSE values for time-to-collision estimation using linear regression with different feature subsets as predictors

Speed	BB-w	MD	BB-w driver	RMSE
✓				1.09
	✓			1.05
		✓		1.24
			✓	1.05
✓	✓			1.03
✓		✓		1.10
✓			✓	1.03
	✓	✓		1.06
	✓		✓	1.06
		✓	✓	1.05
✓	✓	✓		1.05
✓	✓		✓	1.02
✓		✓	✓	1.03
	✓	✓	✓	1.06
✓	✓	✓	✓	1.05

Speed is estimated using the method described in text; *BB-w*: width of the bounding box of the lead vehicle, *MD*: median of MonoDepth disparity values from the bounding box, *BB-w driver*: width of the bounding box of the lead vehicle from the videos of the driver's perspective

Table 2 Overall correlation coefficients between features and time-to-collision (TTC)

	Speed	BB-w	MD	BB-w driver
Speed				
BB-w	– 0.72			
MD	– 0.50	0.34		
BB-w driver	– 0.71	0.95	0.28	
TTC	0.61	– 0.69	– 0.33	– 0.69

Speed is estimated using the method described in text; *BB-w*: width of the bounding box of the lead vehicle, *MD*: median of MonoDepth disparity values from the bounding box, *BB-w driver*: width of the bounding box of the lead vehicle from the videos of the driver's perspective. All correlations are statistically significant at the 0.001 level

a decreasing pattern, with variance decreasing fast below 4 s and the trajectories converging to 0.

Table 1 displays the cross-validated RMSE between the actual and predicted TTC values obtained with linear regression using the different subsets of the predictor features. RMSE values are close to 1 s. Ground truth speed is expected to highly correlate with TTC in normal car stopping situations, like the ones considered in this study. Nonetheless, in other circumstances speed could be replaced by the width of the bounding boxes, as demonstrated by our results: bounding box width alone achieved slightly better performance (1.05 s) than speed (1.09 s). The MonoDepth feature is extremely noisy due to the low-quality images, the shaking of the camera during driving and the distortions due to projections from the spherical videos. It showed a worse RMSE (1.24 s) than the other features. Using multiple features as predictors may improve accuracy only slightly (the minimum RMSE was 1.02 s). Notably, when the BB-w feature is replaced by BB-w driver in the appropriate subsets, prediction performance changes only marginally.

To inspect the relationships between the different features and TTC, Table 2 shows the correlation coefficients computed over all the data resulting from the 29 car stop situations selected for evaluation. The highest correlation (0.95) is shown between the bounding box width from the two information sources, namely the view from the car's and the driver's perspective. Despite being estimated, speed is positively and moderately correlated with TTC (0.61). Bounding box width negatively correlates with TTC (– 0.69), and also there is a weak negative relationship between MonoDepth features and TTC (– 0.33). A strong correlation is observed between speed and bounding box width, for both types of the latter feature (– 0.72 and – 0.71).

Discussion

Our results presented in Tables 1 and 2 indicate that speed and bounding box width are important predictors of TTC. Also, monocular depth information was able to achieve reasonable performance. Moreover, even bounding box width from the view of the driver's perspective was found to be a useful predictor of TTC, as it showed very close results to those from the car's perspective, after accounting for head motions.

We are aware that our restricted dataset contains situations where the speed was decreased slowly and linearly (cf. Figure 2). Still, our results and observations have implications in generalizing the approach to more atypical cases beyond car stop situations. Especially consider the fact that in our analysis we have used estimated speed, because unfortunately we did not record ground truth values during our driving data collection. Nonetheless, using speed and bounding box size as predictors may yield fast and accurate estimates of TTC in more complex driving circumstances. It is also worth mentioning that using monocular depth is a promising direction too, as continuous efforts are being made to improve this fundamental computer vision problem, including its joint training with optical flow in an unsupervised manner (see, e.g., [38]), or combining it with semantic segmentation [14]. Implementing robust depth estimation in mono vision systems has the potential to drastically reduce human error related traffic accidents [31].

Another possibility for generalizing our TTC estimation approach to more complex situations is to use high precision RADAR sensors to measure ground truth distance from the lead car in real traffic and use the data to train a model offline. In general, risk prediction trained offline has the advantage that it overcomes the need for simulation data and leads to potential performance improvements [24, 35]. The pretrained model can then be fine-tuned for specific needs and driving styles.

Some might argue that bounding boxes alone are inaccurate for TTC estimation, as the size changes randomly between consecutive frames [16]. This was not a problem in our experiments. Moreover, results were not impaired considerably neither by the fact that our car stop situations included cases where the lead car was moving forward as well, or cases where another car cut into the frontal lane while the ego-vehicle was slowing down.

On the other hand, an important aspect that affected the precision of our measurements was the poor quality video data acquired. The resolution of our spherical recordings was only 1920×1080, while the recommended resolution is 3840×2160 or even 7168×3584 [2]. However, more powerful spherical cameras are available and we are optimistic that they will further improve in the near future.

One commonly known disadvantage of optical sensors is their sensitivity to changing lighting conditions. Night vision capabilities, infrared mode could represent one solution. Or to tackle the direct sunlight problem computationally efficient high dynamic range imaging algorithms can be applied [23]. Also on-board cameras suffer from vibrations due to vehicle movement; accordingly an additional step of stabilizing videos may be required. The fusion of multiple sensors can yield more reliable and secure systems by validating each others results and providing more precise measurements [10, 18].

To summarize, the presented results are a small-scale, proof of concept evaluation of simple ideas, in which we leveraged novel deep learning methods. In terms of real-time performance, the bottleneck is represented only by the deep learning algorithms. This is an active research area and specialized hardware is being developed in the industry to address the issue [20]. Our evaluations regarding the commonly investigated TTC metric can be generalized to more atypical cases, as only the scale of one parameter (speed) would change. Therefore we expect that the presented method can be scaled up to real emergency braking situations. Furthermore, combining traditional automotive sensors with the spherical camera and potentially smart glasses could support our findings and facilitate improved and more reliable driver assistance systems. This opens up new possibilities, and fosters innovation, but first it should be investigated by follow-up studies as a next step in our research.

Spherical Cameras for Automotive Sensing

The demand for omnidirectional vision in autonomous cars makes the adoption of spherical cameras a promising approach. Using low cost optical sensors in general results in affordable, easy to install collision avoidance systems [10]. While one regular camera can monitor a single view at a time from the full 360-degree neighboring environment [18, 24], a spherical camera can record the front, the side and the rear surroundings of the vehicle at the same time if mounted on top, and can monitor the road ahead and driver behavior as well if mounted inside. This latter will be desired at Level 4 and 5 of automation too as defined by the Society of Automotive Engineers, since human behavior as passengers still remains relevant [30].

Although 360-degree videos pose challenges due to their higher bit rates and larger resolutions, they present less variability in terms of bit rates and camera motion than regular videos [2]. This can be beneficial for the networked system analyzing the video stream. Another main challenge of 360-degree image content is the severe geometric distortion in equirectangular projections. However, research efforts are unfolding to address this issue. For example Yang et al. [36] proposed a multi-projection variant of YOLO and achieved promising results. Cohen et al. [9] extended the commonly used convolutional neural networks to spherical images and demonstrated their computational efficiency, numerical accuracy and effectiveness in 3D model recognition. This direction may eliminate altogether the need for sphere-to-plane projections of 360-degree images.

Conclusion

In this work we proposed a simple method for time-to-collision estimation to overcome the lack of labeled data problem caused by the rarity of collisions in everyday traffic. We rely solely on monocular vision and exploit state-of-the-art deep learning-based optical flow and depth estimation methods to estimate the speed of the ego-vehicle using a straightforward and intuitive approach, and object detection as well to identify the lead vehicle. We made use of the approximated speed, width of the bounding boxes—both from videos from the car's and the driver's perspective—and monocular depth features to estimate time-to-collision in car stop situations where the moment of the potential collision is assumed to be known. The proposed approach was evaluated on our self-collected driving videos recorded with an off-the-shelf spherical camera and smart glasses. We achieved a cross-validated RMSE of close to 1 s on both the road view and the ego-centric videos. Our results and conclusions may have implications in designing and improving collision avoidance systems for self-driving cars.

As an additional contribution, we also discussed the benefits of using spherical cameras as a favorable replacement of multiple traditional cameras for vision-based automotive sensing. We are confident that given the unfolding recent research efforts regarding 360-degree images, the rapid improvement of deep learning models, and the continuous technological and engineering advancements, monocular vision using spherical cameras will become widely applied in the autonomous driving domain.

Acknowledgements The authors acknowledge András Lőrincz, their Ph.D. supervisor, for his efforts in the acquisition of funding and his help in starting this research project. The authors would also like to thank Szilvia Szeier and Kevin A. Hartyányi for their work and useful ideas during the progression of the project.

Author Contributions Conceptualization: R-AR, KBF; Methodology: R-AR; Software: R-AR; Validation: R-AR; Formal analysis: R-AR; Investigation: R-AR, KBF; Resources: R-AR, KBF; Data curation: R-AR, KBF; Writing–original draft preparation: R-AR; Writing–review and editing: R-AR, KBF; Visualization: R-AR; Supervision: R-AR, KBF; Project administration: R-AR, KBF.

Funding Open access funding provided by Eötvös Loránd University.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Abuella H, Miramirkhani F, Ekin S, Uysal M, Ahmed S. Vildar - visible light sensing based speed estimation using vehicle's headlamps. IEEE Trans Veh Technol. 2019. https://doi.org/10.1109/TVT.2019.2941705.
2. Afzal S, Chen J, Ramakrishnan KK. Characterization of 360-degree videos. In: Proceedings of the workshop on virtual reality and augmented reality network, ACM, New York, NY, USA; 2017. pp. 1–6, https://doi.org/10.1145/3097895.3097896.
3. Aichinger C, Nitsche P, Stütz R, Harnisch M. Using low-cost smartphone sensor data for locating crash risk spots in a road network. Transp Res Proc. 2016;14:2015–24. https://doi.org/10.1016/j.trpro.2016.05.169.
4. Al-Qizwini M, Barjasteh I, Al-Qassab H, Radha H. Deep learning algorithm for autonomous driving using googlenet. In: IEEE Intell Veh Symp (IV). 2017. pp. 89–96. https://doi.org/10.1109/IVS.2017.7995703.
5. Bello LL, Mariani R, Mubeen S, Saponara S. Recent advances and trends in on-board embedded and networked automotive systems. IEEE Trans Ind Informatics. 2019;15(2):1038–51. https://doi.org/10.1109/TII.2018.2879544.
6. Bojarski M, Testa DD, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J, Zhang X, Zhao J, Zieba K. End to end learning for self-driving cars. CoRR. 2016. arxiv:1604.07316.
7. Chen C, Seff A, Kornhauser A, Xiao J. Deep driving: learning affordance for direct perception in autonomous driving. In: IEEE Int Conf Comput Vis (ICCV). 2015. pp. 2722–30. https://doi.org/10.1109/ICCV.2015.312.
8. Chen Y, Palanisamy P, Mudalige P, Muelling K, Dolan JM. Learning on-road visual control for self-driving vehicles with auxiliary tasks. In: IEEE Winter Conf Appl Comput Vis (WACV). 2019. pp. 331–8. https://doi.org/10.1109/WACV.2019.00041.
9. Cohen TS, Geiger M, Köhler J, Welling M. Spherical CNNs. CoRR. 2018. arxiv:1801.10130.
10. Dagan E, Mano O, Stein GP, Shashua A. Forward collision warning with a single camera. IEEE Intell Veh Symp. 2004. pp. 37–42. https://doi.org/10.1109/IVS.2004.1336352.
11. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conf Comput Vis Pattern Recogn (CVPR). 2012. pp. 3354–61. https://doi.org/10.1109/CVPR.2012.6248074.
12. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: The KITTI dataset. Int J RobotRes (IJRR). 2013; 32(11):1231–37. https://doi.org/10.1177/0278364913491297.
13. Godard C, Aodha OM, Brostow GJ. Unsupervised monocular depth estimation with left-right consistency. In: IEEE Conf Comput Vis Pattern Recogn (CVPR). 2017. pp. 6602–11. https://doi.org/10.1109/CVPR.2017.699.
14. Jiang H, Larsson G, Maire M, Shakhnarovich G, Learned-Miller E. Self-supervised relative depth learning for urban scene understanding. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision - ECCV 2018. Cham: Springer International Publishing; 2018. pp. 20–37. https://doi.org/10.1007/978-3-030-01252-6_2.
15. Kendall A, Hawke J, Janz D, Mazur P, Reda D, Allen JM, Lam VD, Bewley A, Shah A. Learning to drive in a day. In: Proc Int Conf Robot Autom (ICRA). 2019. pp. 8248–54. https://doi.org/10.1109/ICRA.2019.8793742.
16. Kilicarslan M, Zheng JY. Predict vehicle collision by TTC from motion using a single video camera. IEEE Transactions on Intelligent Transportation Systems. 2019;20(2):522–33. https://doi.org/10.1109/TITS.2018.2819827.
17. Menze M, Geiger A. Object scene flow for autonomous vehicles. In: Conf Comput Vis Pattern Recogn (CVPR). 2015. pp. 3061–70. https://doi.org/10.1109/CVPR.2015.7298925.
18. Mukhtar A, Xia L, Tang TB. Vehicle detection techniques for collision avoidance systems: A review. IEEE Transactions on Intelligent Transportation Systems. 2015;16(5):2318–38. https://doi.org/10.1109/TITS.2015.2409109.
19. Nègre A, Braillon C, Crowley JL, Laugier C. Real-Time Time-to-Collision from Variation of Intrinsic Scale, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. pp. 75–84. https://doi.org/10.1007/978-3-540-77457-0_8.
20. NVIDIA Corporation. NVIDIA DRIVE INFRASTRUCTURE: End-to-end solutions for training, development, and validation of autonomous vehicles. 2021. http://www.nvidia.com/en-us/self-driving-cars/infrastructure/. Accessed 17 May 2021.
21. Palotai Z, Láng M, Sárkány A, Tősér Z, Sonntag D, Toyama T, Lőrincz A. Labelmovie: Semi-supervised machine annotation tool with quality assurance and crowd-sourcing options for videos. In: 12th International workshop on content-based multimedia indexing (CBMI), 2014. pp. 1–4. https://doi.org/10.1109/CBMI.2014.6849850.
22. Patra S, Veelaert P, Calafate CT, Cano JC, Zamora W, Manzoni P, González F. A forward collision warning system for smartphones using image processing and v2v communication. Sensors. 2018;18(8). https://doi.org/10.3390/s18082672.
23. Paul N, Chung C. Application of hdr algorithms to solve direct sunlight problems when autonomous vehicles using machine vision systems are driving into sun. Comput Indus. 2018;98:192–6. https://doi.org/10.1016/j.compind.2018.03.011.
24. Phillips DJ, Aragon JC, Roychowdhury A, Madigan R, Chintakindi S, Kochenderfer MJ. Real-time prediction of automotive collision risk from monocular video. CoRR. 2019. arxiv:1902.01293.
25. Pundlik S, Peli E, Luo G. Time to collision and collision risk estimation from local scale and motion. In: Bebis G, Boyle R, Parvin B, Koracin D, Wang S, Kyungnam K, Benes B, Moreland K, Borst C, DiVerdi S, Yi-Jen C, Ming J, editors. Advances in visual computing. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. pp. 728–37.
26. Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. In: IEEE conference on computer vision and pattern recognition (CVPR). 2017. pp. 6517–25. https://doi.org/10.1109/CVPR.2017.690.

27. Redmon J, Farhadi A. YOLOv3: An incremental improvement. CoRR. 2018. arxiv:1804.02767.
28. Redmon J, Divvala SK, Girshick RB, Farhadi A. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. pp. 779–88. https://doi.org/10.1109/CVPR.2016.91.
29. Rill RA. Intuitive estimation of speed using motion and monocular depth information. Studia Universitatis Babeş-Bolyai Informatica. 2020;65(1):33–45. https://doi.org/10.24193/subbi.2020.1.03.
30. Ryerson MS, Miller JE, Winston FK. Edge conditions and crash-avoidance roles: the future of traffic safety in the world of autonomous vehicles. Inj Prev. 2019;25(2):76–9. https://doi.org/10.1136/injuryprev-2017-042567.
31. Schennings J. Deep convolutional neural networks for real-time single frame monocular depth estimation. UPTEC F. 2017;17060. Thesis at Uppsala University, Division of Systems and Contro.l.
32. Shanker R, Jonas A, Devitt S, Huberty K, Flannery S, Greene W, Swinburne B, Locraft G, Wood A, Weiss K, Moore J, Schenker A, Jain P, Ying Y, Kakiuchi S, Hoshino R, Humphrey A. Autonomous cars: Self-driving the new auto industry paradigm. Morgan Stanley Research Division: Morgan Stanley Blue Paper; 2013.
33. Shi C, Dong Z, Pundlik S, Luo G. A hardware-friendly optical flow-based time-to-collision estimation algorithm. Sensors. 2019;19(4). https://doi.org/10.3390/s19040807.
34. Sun D, Yang X, Liu MY, Kautz J. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. pp. 8934–43. https://doi.org/10.1109/CVPR.2018.00931.
35. Wulfe B, Chintakindi S, Choi SCT, Hartong-Redden R, Kodali A, Kochenderfer MJ. Real-time prediction of intermediate-horizon automotive collision risk. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems. 2018. pp. 1087–96. http://dl.acm.org/citation.cfm?id=3237383.3237858.
36. Yang W, Qian Y, Kämäräinen JK, Cricri F, Fan L. Object detection in equirectangular panorama. In: 2018 24th International Conference on Pattern Recognition (ICPR). 2018. pp. 2190–95. https://doi.org/10.1109/ICPR.2018.8546070.
37. Yao B, Feng T. Machine learning in automotive industry. Adv Mech Eng. 2018. https://doi.org/10.1177/1687814018805787.
38. Zou Y, Luo Z, Huang JB. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision - ECCV 2018. Cham: Springer International Publishing; 2018. pp. 38–55.