Vision-based Overhead Front Point Recognition of Vehicles for Traffic Safety Analysis

Byeongjoon Noh
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
powernoh@kaist.ac.kr

Wonjun No
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
jn0704@kaist.ac.kr

David Lee
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
david733@kaist.ac.kr

UbiComp/ISWC'18 Adjunct, October 8–12, 2018, Singapore, Singapore
© 2018 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-5966-5/18/10…$15.00
https://doi.org/10.1145/3267305.3274165

Abstract

Pedestrian-vehicle accidents are the cause of many human injuries and deaths. To address this challenge, vision-based traffic systems have focused on detecting traffic-related objects' behaviors, such as vehicle position and velocity relative to pedestrians. In this paper, we propose a new and simple model for effectively recognizing overhead front point of vehicles, while only using a single stationary camera capturing from an oblique angle. The proposed system uses faster R-CNN model for detecting object bounding box and mask, projects the mask's extreme points down to find the car's ground front point, and transforms these coordinates from oblique to overhead frame of reference. Our experimental result shows that this method is effective for recognizing overhead front point of car (accuracy: 92.4%) within a certain tolerance.

Author Keywords

object detection; traffic safety system; overhead front point recognition; crosswalk accident;

ACM Classification Keywords

I.4.8. Overhead front point recognition: Scene Analysis; I.4.9. Vision-based traffic safety system: Applications; I.4.m. Vehicle's behavior detection: Miscellaneous

Introduction

Over 1 million people die each year from road traffic accidents, and up to 50 million are injured [1]. Especially, most accidents where pedestrians are injured occur at crosswalks, since pedestrians are most exposed to fast vehicle movement in these spaces [2]. As one of the methods for alleviating deaths of vulnerable road users (VRUs), vision-based traffic safety/surveillance systems have been extensively applied, focusing on traffic-related objects' (e.g. pedestrians and vehicles) behavior analysis and potential collision risk (e.g. near-miss collision) analysis [1]. Consequently, these vision-based systems make it possible to understand driver behavior in specified places, identify patterns, prioritize the most dangerous places, and test improvements to their design.

In vision-based traffic safety systems, one important process is to extract the vehicles' and pedestrians' behavioral factors from video, such as vehicles' velocity, direction, number of pedestrians, and their positions. These factors are applied to detect dangerous events and quantify risk using time-to-collision (TTC), post-encroachment time (PET) and other measures [3]. Therefore, it is essential to extract these features accurately from video.

However, many studies have extracted these features by manual inspection from huge volumes of traffic video [4]. In addition, most vision sensors, such as closed circuit television (CCTV) cameras, take oblique views of the vehicles, making it difficult to extract the precise features such as velocity and position from an overhead view. Often, extracting these features requires multiple sensors and additional complex processing, at additional time and cost.

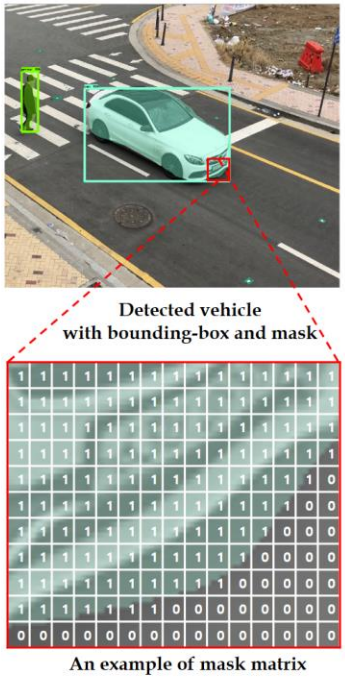
In order to address these challenges, we propose a simpler model for effectively recognizing overhead front points of vehicles by using single sensor, RGB camera. The proposed system consists of three core modules; 1) object detection module; 2) vehicle's front point recognition module; and 3) perspective transformation module.

First, in the object detection module, faster R-CNN (region-convolutional neural network) model is used to detect the object in video frames. As output of object detection, we can obtain the bounding boxes and masks of the detected objects. In the second module, we extract the ground-front point of the vehicles using the masks generated in the previous step. Finally, all coordinates are transformed into the overhead frame of reference from the oblique view.

Related works

Much research has been conducted on extracting the more precise object behavioral features, such as velocity, in fields of traffic safety and computer vision. Early, they focused on discriminating objects by using various types of statistical learning classifiers such as support vector machines (SVMs) and AdaBoost based on a vector of raw pixel value or features extracted from them, and the output is the decision showing whether the object was detected [1][5]. In addition, for recognizing the objects, fusion of multiple sensors was used. As an example, stereo sensors using visible light and thermal images were used to detect and track objects in order to obtain their 3-D locations [6].

The development of neural network approaches resulted in faster R-CNN models specialized in detecting most types of objects in image. These are now widely



1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 1: Result of object detection and mask matrix

used in various fields as well as traffic safety area [7]. Thus, detecting objects in images is no longer the critical technical barrier; now, extracting an object's behavioral features, processing and analyzing them have become the main issues. Recognizing the overhead front point of the object, especially vehicles, allows us to extract other features, such as velocity, position, and direction, more accurately.

In a prior study for recognizing the overhead front point of the vehicle, *Markéta et al*. proposed a 3-D bounding box constructing method by using the camera calibration [8]. They assumed that vehicle silhouettes can be extracted by background modeling and foreground detection. Then, they used the tangent lines and their relevant intersections to construct the bounding box. However, this method requires high time complexity for extracting vehicle silhouettes by applying background modeling, and calculating tangent lines by vehicle's facing angle. *Zhang et al.* proposed car front detection method by recognizing the license plate on the vehicle and extracting features from this area as front parts of vehicles with cascade ensembles method [9]. This method is good to detect car front. However, it is difficult for this method to extract the accurate front point of the car from an overhead view, since the license plate is some distance above the ground, with its pixel location projecting to the ground somewhere underneath or behind the car's body.

In order to address these challenges, the proposed system uses only one RGB camera sensor, which costs less and is easier to deploy than using multiple sensors. In addition, it makes manual inspection of huge volumes of traffic video unnecessary when extracting the front point location of vehicles in each frame. Consequently, it can quickly derive a vehicle's velocity and acceleration in overhead view using multiple frames. This makes it possible to detect and analyze critical vehicle-pedestrian interactions, such as near-miss collision or potential risk events, based on traffic video.

Materials and Methods

Object Detection based on Faster R-CNN model

In order to detect vehicles, we use faster R-CNN model with mask output [10]. To date, the incredible development of deep CNNs, in field of computer vision, has dominated the various tasks in image processing and recognition [11]. In particular, faster R-CNN is widely used to detect many types of objects in a frame [12]. Here, we used object detection API using the faster R-CNN model with pre-trained weights for Microsoft common objects in context (MS COCO) [13][14].

In our experiment, since the goal of this module was to detect only vehicles and pedestrians to represent their masks as output, additional training was not needed. The accuracy of detecting vehicles and pedestrians is approximately 99.9%. As a result, we obtained the bounding-boxes and masks from input images in form of matrices (see Figure 1). The size of each matrix was equal to that of the bounding-box. If the coordinates of a pixel was on the object, its value was 1, otherwise, 0.

Front Point Recognition

In this section, we describe how to recognize the front point of the vehicle, as the core methodology of the proposed system. We use the extreme points of the object mask along the x and y dimensions.

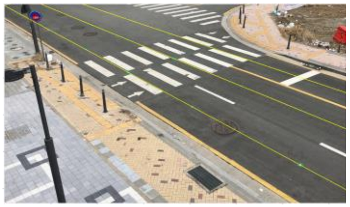


Figure 3: Lane graph (yellow line)

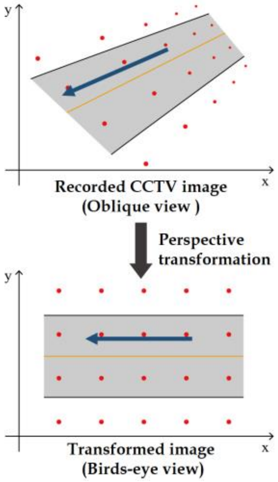


Figure 4: Example of perspective transformation

First, we need to determine whether the car is moving toward the camera or not, by calculating its bounding box movement along the x-axis in consecutive frames. When the car is moving toward camera (left to right), the change in x-value is positive, and the opposite case negative. If the car is moving toward the camera, mask's x-max tip and y-max tip points are calculated, then we can get the front-center of these two points, mask's center point (see Figure 2). In case of reverse direction, mask's x-min tip and y-min tip points are used.

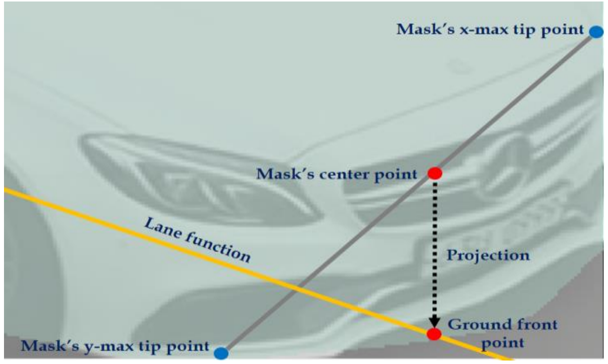


Figure 2: Method of finding ground point with lane function and mask's center point

Generally speaking, since the vehicle moves parallel to the road, we assume that the front point of the vehicle is on the road as seen in Figure 3. Thus, we create a virtual straight line running parallel to the lane and centered between its bounds, called the lane function, and project the mask's center point down to it.

This represents a point on the ground directly underneath the front center of the car, in the oblique view.

Perspective Transformation

By transforming the perspectives, we can obtain the front point in overhead view (see Figure 4). First, we marked the orthogonal grid points at intervals of 5 meters on the road as green dots in Figure 4. Based on four of these points, and their corresponding pixel locations, a transformation matrix is calculated by using *getPerspectiveTransform()* function in OpenCV. We can apply this transformation to any points in the oblique view, converting them to the overhead perspective.

In our experiment, we only transformed the red rectangular zone in Figure 5. As a result, Figure 6 illustrates the transformed image, which includes transformed grid points (white points), crosswalks and sidewalks (red and green regions, respectively).Figure 6 shows the trace of the front-point of a vehicle moving right to left for 25 frames (color gradient representing time).

With the proposed model, we can obtain the more precise overhead front point of vehicles, and calculate various features such as distance, velocity, and directions.

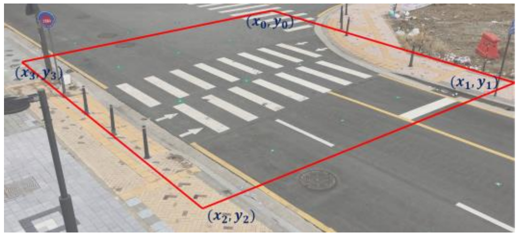


Figure 5: Transformed region (within red rectangle)

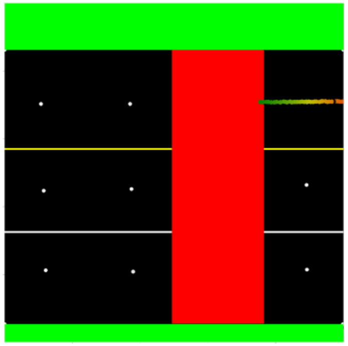


Figure 6: Result of re-constructed frame and trace of vehicle front

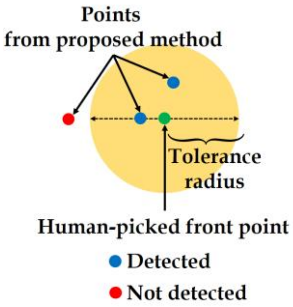


Figure 7: Example of distance tolerance boundary

Experimental Results

In this section, we describe the experimental design for the proposed model recognizing overhead front point of a vehicle. We recorded video for 25minutes with 30 frames/sec at an oblique angle over a crosswalk. The size of area for experiment is about 20m in width and 30m in length, and contains the crosswalk. This video data included 100 frames for each vehicle, consisting of 50 left-to-right frames and 50 right-to-left frames.

To evaluate the feasibility and applicability of the proposed model, we used a concept of "distance tolerance". In fact, it is difficult to define a point that exactly represents the front point of the vehicle by using only one camera sensor. Therefore, we tolerate some errors, and we assumed that if there are estimated points (using the proposed model) within the error boundary, these points are properly detected as follows:

$$\begin{cases} detected; & dist\big(point(\hat{x_i},\hat{y_i}), point(x_i,y_i)\big) \le dist_{error} * k \\ not\ detected; & otherwise \end{cases}$$

where

$point(\hat{x_i},\hat{y_i})$: $estimated\ point\ using\ proposed\ method\ in\ i^{th}\ frame,$

$point(x_i,y_i)$: $real\ point\ selected\ by\ testers\ in\ i^{th} frame$

$dist_{error}$: $distance\ tolerance$

k: $distance\ transformation\ constant$

The value of k is constant which converts real-world distance into image distance. In our experiment, since baseline is $5m$ (green dots in Figure 5), and its distance $260pixel$ in image, the value of k is 0.52 (260/500)$pixel/cm$.

For every distance tolerance, we can derive a velocity tolerance. Since distance has some level of error, velocity also has some level of error. Practically speaking, the velocity error is calculated by measuring the distance between two frames, and dividing by the time between those frames as follows:

$$Velocity_{error} = \frac{2 * dist_{error}}{(1/FPS)}$$

If assuming 30FPS video, and sampling every 10th frame, time between frames is 1/3 seconds. Then, the distance tolerance is 30cm, then the max potential error in that distance the car moved between two frames is $60cm$, or $0.6m$. Thus, velocity tolerance in this scenario is 1.8 m/s, or 6.48 km/h.

In order to conduct comparison, we recruited 5 testers, and asked them to select the pixel location for the actual front point of the vehicle for 100 frames. Each frame includes one vehicle object. Then, we compared the difference between the points derived from the proposed model and the points selected by the testers, and measured accuracy according to various distance tolerances (10cm, 20cm, 35cm, and 50cm) as seen in Figure 7. If the distance between the point picked by the tester and the point derived from the proposed model is within the tolerance radius, we consider that front point correctly detected.

As the result of the comparison, the average accuracy is more than 92.4% when the distance tolerance boundary is $50cm$ (see Table 1). According to the formula, when distance tolerance is $50cm$, velocity tolerance is about $10.8km/h$. In our future research, risk of a vehicle's behavior will be assessed based on its velocity within increments of 10km/h. This makes our detection system mostly sufficient in locating vehicles precisely enough to classify their risk within our model.

Tolerance (cm)	**10**	**20**	**35**	**50**
Proposed model vs. Tester1	37%	67%	92%	98%
Proposed model vs. Tester2	29%	53%	80%	92%
Proposed model vs. Tester3	6%	17%	39%	91%
Proposed model vs. Tester4	15%	29%	64%	89%
Proposed model vs. Tester5	8%	19%	53%	92%
Avg.	19%	37%	65.6%	**92.4%**

Table 1: Result of comparison between proposed model points and 5 testers for tolerance

Conclusions

In this paper, we proposed a simpler model for effectively recognizing the front point of a moving vehicle. The proposed model applied pre-trained faster R-CNN model to detect objects, found ground point by using the lane function and object mask, and transformed the oblique perspective into overhead view. We also confirmed that the proposed model can accurately and stably detect the overhead front point of the vehicle within a certain tolerance by experimental results. As a result, when tolerance is 50cm, average accuracy against manual detection is 92.4%. This selected distance tolerance is also precise enough to distinguish degree of risk according to velocity.

For next steps, we will define rules for potential risk events based on the vehicles' and pedestrians' velocity, direction, and position relative to each other and their context. For example, it is a very dangerous event when a car passes in front of a pedestrian crossing the street, at high speed without stopping. We believe detecting such events will help clarify the causes and contributors to vehicle-pedestrian accidents in crossings throughout the city, and allow us to quickly identify the impacts of design changes on driver behavior in those places.

The final purpose of this ongoing research is to demonstrate an automated methodology diagnosing traffic safety at specified roads such as unsignalized crosswalks and intersections, and to analyze the potential risky event by extracting traffic-related object (e.g. pedestrian and driver) data required to understand their behaviors.

Acknowledgments

We thank Ryan Ramelo for his help in reviewing this manuscript. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1D1A1B07049129), and by Korea Ministry of Land, Infrastructure and Transport (MOLIT) as "U-City Master and Doctor Course Grant Program".

References

1. Gandhi, Tarak, and Mohan Manubhai Trivedi. "Pedestrian protection systems: Issues, survey, and challenges." *IEEE Transactions on intelligent Transportation systems* 8.3 (2007): 413-430.

2. Fu, Ting, Luis Miranda-Moreno, and Nicolas Saunier. "Pedestrian crosswalk safety at nonsignalized crossings during nighttime: use of thermal video data and surrogate safety measures." *Transportation Research Record:*

Journal of the Transportation Research Board 2586 (2016): 90-99.

3. Hussein, Mohamed, et al. "Automated Pedestrian Safety Analysis at a Signalized Intersection in New York City: Automated Data Extraction for Safety Diagnosis and Behavioral Study." *Transportation Research Record: Journal of the Transportation Research Board* 2519 (2015): 17-27.

4. Kadali, B. Raghuram, and P. Vedagiri. "Proactive pedestrian safety evaluation at unprotected mid-block crosswalk locations under mixed traffic conditions." *Safety science* 89 (2016): 94-105.

5. Munder, Stefan, and Dariu M. Gavrila. "An experimental study on pedestrian classification." *IEEE transactions on pattern analysis and machine intelligence* 28.11 (2006): 1863-1868.

6. Krotosky, Stephen, and Mohan Trivedi. "Multimodal stereo image registration for pedestrian detection." *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. IEEE, 2006.

7. Herman, S., and K. Ismail. "Single Camera Object Detection for Self-Driving Vehicle: A Review." *Journal of the Society of Automotive Engineers Malaysia* 1.3 (2017): 198-207.

8. Dubská, Markéta, Adam Herout, and Jakub Sochor. "Automatic Camera Calibration for Traffic Understanding." *BMVC*. Vol. 4. No. 6. 2014.

9. Zhang, Bailing. "Reliable classification of vehicle types based on cascade classifier ensembles." *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013): 322-332.

10. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.

11. Pham, Minh-Tan, and Sébastien Lefèvre. "Buried object detection from B-scan ground penetrating radar data using Faster-RCNN." *arXiv preprint arXiv:1803.08414* (2018).

12. Jiang, Huaizu, and Erik Learned-Miller. "Face detection with the faster R-CNN." *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. IEEE, 2017.

13. https://github.com/tensorflow/models/tree/master/research/object_detection [Accessed: 05-Jul-2018]

14. . http://cocodataset.org/#home [Accessed: 05-Jul-2018]