

# A visual blindspot monitoring system for safe lane changes

Jamal Saboune<sup>1</sup>, Mehdi Arezoomand<sup>1</sup>, Luc Martel<sup>2</sup>, and Robert Laganier<sup>1</sup>

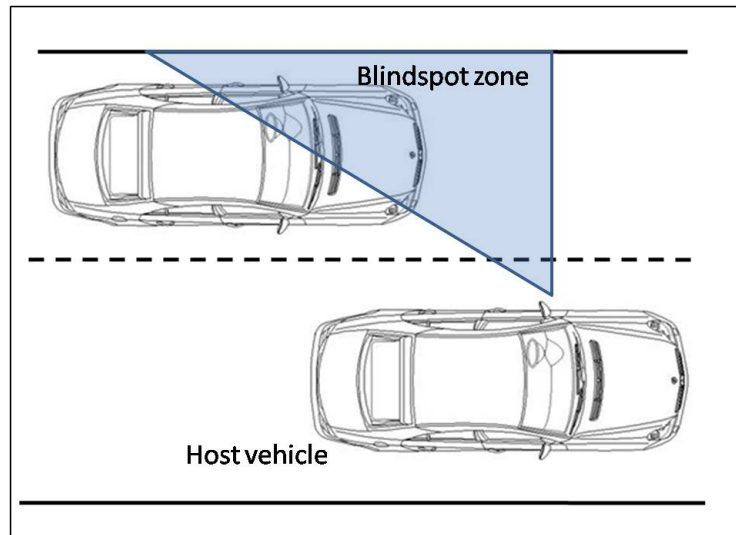
<sup>1</sup> VIVA Lab, School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

<sup>2</sup> Cognivue Corporation, Gatineau, Quebec J8X 4B5 Canada  
jsaboune, marezoom, laganier@site.uottawa.ca, lmartel@cognivue.com

**Abstract.** The goal of this work is to propose a solution to improve a driver's safety while changing lanes on the highway. In fact, if the driver is not aware of the presence of a vehicle in his blindspot a crash can occur. In this article we propose a method to monitor the blindspot zone using video feeds and warn the driver of any dangerous situation. In order to fit in a real time embedded car safety system, we avoid using any complex techniques such as classification and learning. The blindspot monitoring algorithm we expose here is based on a features tracking approach by optical flow calculation. The features to track are chosen essentially given their motion patterns that must match those of a moving vehicle and are filtered in order to overcome the presence of noise. We can then take a decision on a car presence in the blindspot given the tracked features density. To illustrate our approach we present some results using video feeds captured on the highway.

## 1 Introduction

Car accidents on the highways are a big factor of mortality and can cause severe injuries. Actually, drivers nowadays are getting more concerned about safety features in their cars and thus are willing to pay the cost of acquiring safer vehicles. On the other hand, the public services are interested in reducing the mortality rate on the roads considered nowadays as an indicator of the quality of life. Motivated by economic factors, a new research domain has thus emerged in the recent years; It is known as pre-crash sensing. The research in this domain, conducted by car manufacturers as well as by public research institutions, aims to make the vehicles safer and as a result reduce the number of crashes and their severity. The main threat for a driver on the highway comes from the surrounding cars especially when he is not aware of their close presence. In fact one of the the main features of an onboard car safety system is to detect the presence of a close car in the driver's blindspot (Figure 1) and warn the latter about it. This information can help the driver in a lane change situation and affect his decision to perform this task. In this paper we present a simple and fast approach for blindspot monitoring using computer vision. This feature is essential and can help preventing many risky driving situations.



**Fig. 1.** The blindspot zone description: We define the blindspot of a driver as the zone he can not see through his side and rear view mirrors.

The blindspot monitoring problem is a problem of a car detection in a given zone surrounding the host car. This car detection task which is the initial step to accomplish in any collision avoidance system, has been widely addressed. The first generation of collision avoidance systems is based on using a radar technology. These systems adopt diverse technologies such as infrared, ultrasonic waves [23], sonars [15] or laser scanners [27] in order to detect the presence of any object in the range of the sensors embedded in the car's body. Some of the radars used can also detect the shape of the object. However the radars used have a small range and as a result are not able to detect some approaching vehicles. On the other hand, their field of view is reduced and present blindspots. Thus, in order to cover the wide area surrounding the car many sensors are needed which increases the cost of the system. With the recent advances in nanotechnology and in imagery sensing devices the new generation of embedded security system is relying on the use of small cameras installed in different locations of the vehicle; the cameras can have a large field of view which enables the system to detect vehicles moving in a large area and overcome the radars disadvantages. The cameras are also low cost and can be combined with radars to improve the car detection [14–16, 19, 27].

In order to detect a vehicle in the camera feeds, three approaches were adopted in previous works. The first one known as 'knowledge based' relies on recognizing the vehicle in a certain single image given some distinctive features. In fact, vehicles have some distinctive visual features (color, shape etc.) and thus vehicle detection in an image can be reduced to a classical pattern recognition problem. This problem can be solved using a features vector classification

technique given a database of learned features vectors representing vehicles and roads. The features used for classification can be of different types. Tsai et al. [25] use color and edge information and a bayesian classifier to resolve this problem. Haar like [10, 20, 27] and Histogram of Oriented Gradient (HOG) [2, 17, 20, 1] features were also widely used. These distinctive features can then be classified using Support Vector Machine (SVM) [2, 17, 1] or Adaboost classifiers [10, 20, 27]. However, vehicles are of different shapes and colors and can be viewed from different angles and under different illumination conditions in videos. Thus, the database should be very wide in order to be inclusive and to have a good recognition. The classification step is also time consuming. Given that, we can say that these algorithms are complex and not well adapted to an on-board system. In order to avoid the learning and classification steps, other methods using a car distinctive features were proposed. Detecting the shadow underneath a car is a sign of a car's presence [8, 15, 28]. Unfortunately it is not always possible to detect this shadow especially for cars moving far from the camera or in a cloudy or dark environment. This feature can thus be combined with other features such as vertical edges and symmetry rate [15], left and right car borders [8] or lights detection [28] for night situations. Despite those improvements, the vehicles recognition in this type of approaches is not accurate and is highly perturbed by shadows of the background objects (guard rails, trees etc.). Collado et al. [9] constructed geometric models of the car with energy functions including shape and symmetry. This approach succeeded in detecting far preceding cars but presented some weakness in detecting lateral and close cars. Wu et al. [29] succeeded in detecting cars in the blindspot by comparing the grayscale histogram of the road surface to that of a patch covering the neighbouring lane. This idea is effective in detecting any object whose colour is different from that of the road but does not imply that the object is a car.

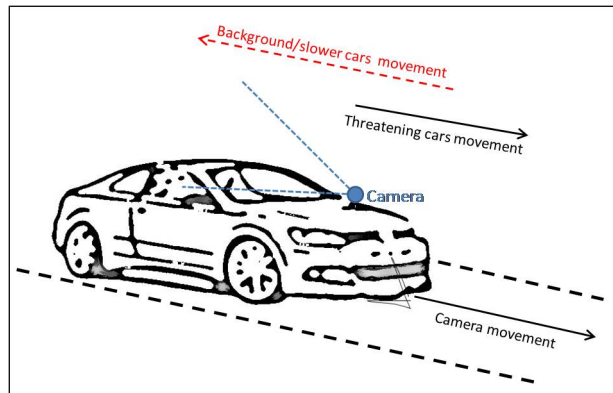
The second approach known as 'motion based' uses an object's motion information estimated through successive images to detect the vehicle's presence. This idea is motivated by the fact that a vehicle moves relatively to the background with a standard motion pattern. In order to estimate the motion of an object we need to find the correspondances between the features describing it in the successive images. To accomplish this, color, edge and contour information [5, 21, 24] can be efficient as well as SURF [6] or SIFT [13] features. Spatiotemporal wavelet transforms [26], image entropy [7] and optical flow algorithms [3, 11] were also employed for motion estimation. These techniques proved to be less complex and more efficient than the 'knowledge based' ones although they present some errors in specific cases. The motion and knowledge based approaches can also be combined. The third technique is a stereo vision method [4, 12] that calculates the disparity map and accomplishes a 3D reconstruction but is very complex and highly inaccurate.

In order to simplify the problem and to develop a fast algorithm that can be easily implemented we decided to adopt a 'motion based' strategy for car detection and tracking in the blindspot. Our method uses an optical flow technique to estimate the motion of some features that would represent a car. These features

are well chosen in a way to avoid false detections and thus reduce the algorithm complexity. The developed method will be exposed in section 2. Results and discussion will later be exposed in section 3.

## 2 Blindspot car detection and tracking

The easiest and most efficient solution to detect a car moving in front of the camera can be accomplished by detecting the zone in the image representing it and then track it using a template matching technique. To detect a vehicle presence in the blindspot, we cannot use the same logic. In fact, in order to accomplish this task efficiently we need to install the camera in a way to be able to see a car when it is approaching from behind and then passing by the driver. We cannot therefore install the camera on the backside of the car but it should be installed in front. In this scene's configuration, a car shape and contour seen from the front, change continuously when its approaching or passing by the driver. As a result, a template matching technique would act poorly. In order to overcome this problem we decided to adopt a 'motion based' approach using the optical flow calculation. Our idea is motivated by the fact that when the camera is moving forward, the background objects and the slower cars the driver is passing by, move backward relatively to the camera. On the other hand, the approaching cars which present a threat move with a motion pattern close to that of the camera or faster and thus move relatively forward. (Figure 2).



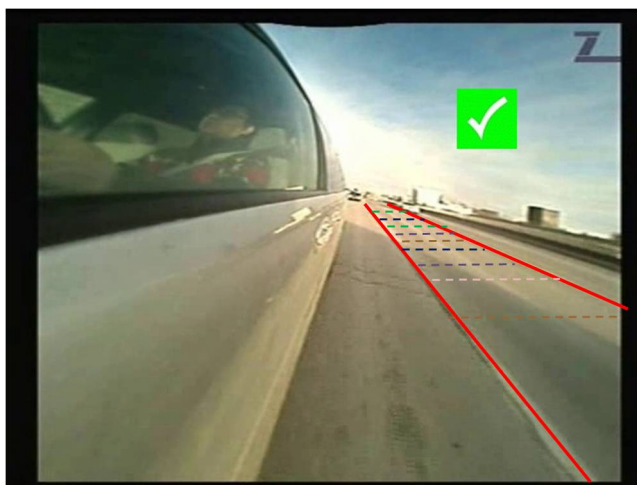
**Fig. 2.** Configuration of the camera for the blindspot detection problem: The background objects and slower cars move relatively backward and the threatening cars move forward.

To estimate the motion of the moving objects in the successive images we use an optical flow calculation approach on a group of features describing these objects. We opted to use the Shi & Tomasi Good features to track [22] that proved to be efficient for object tracking and to use the pyramidal Lucas - Kanade

tracker [18] for optical flow calculation. We then apply a number of filters in order to make sure we are tracking only features representing a vehicle and not noise. The different steps of our algorithm are the following:

- We first specify a zone in the grayscale image covering the lane next to the camera; This will be the zone we want to track a vehicle in (the blindspot zone).
- Every 8 frames we extract the features to track in this zone, as we observed that new objects do not appear completely in less than that temporal distance. Thus we are able to reduce the complexity of the method. The extracted features are added to the set of features to track  $S$  (resulting from the previous frames).
- By calculating the optical flow of all features in  $S$  we then estimate the motion vector of each. We consider that the motion vectors verifying some conditions represent a potential car and we label them as 'valid features'. Else, they are rejected from  $S$  and we stop tracking them. These conditions are based on the motion patterns described earlier; In fact if a motion vector forms an angle with the horizontal greater than 30 deg. and smaller than 60 deg. and has a value bigger than 1 pixel it is considered as describing a 'valid feature'. This choice is justified by the fact that objects moving with such motion pattern would represent objects (cars) moving in a similar way to the host car and thus are considered as dangerous. Else if a car in the blindspot zone moves with a motion vector which angle to the horizontal is between -30 and 30 deg or 60 and 120 deg its driver is most probably trying to change lanes to the left (getting far from the host car) or to the right (going behind the host car in the same lane). In all the other configurations of this angle the object is either a part of the background or moving slower than the host car and as a result its non-threatening.
- We observe the motion vectors of a 'valid feature' calculated by optical flow over three frames. If these vectors respect the motion conditions we established earlier, for the three frames, we label the corresponding feature as a 'potential feature' and we keep it in our set  $S$ . If not, it would be rejected as well and its tracking is stopped. We thus make sure that we eliminate objects having an inconsistent movement.
- In order to avoid tracking features representing some noise we impose an additional condition on the tracked 'potential features'. If a car is present in a certain zone of the blindspot area, its corresponding features should be as well. As a result, we would have a minimum number of 'potential features' in this zone. On the other hand, if a feature is isolated in the zone it is most likely that it represents noise. To illustrate this idea we divide the blindspot zone in five zones of different sizes and we impose a threshold for each of them (Figure 3). If the number of 'potential features' present in one of the

zones is less than the threshold fixed for the zone, these features are rejected. The zone containing the biggest number of features is considered as the one containing the vehicle.



**Fig. 3.** Image decomposition into zones: The region of the image delimited by the two red lines is considered as the blindspot zone. The dashed colored lines delimit the zones considered as potentially containing a vehicle. Their sizes are estimated given the shape of the vehicle in the next lane and its distance from the camera.

- Despite these strict conditions some noise features were able to survive and we had to add a last condition. It is based on the comparison of the pixels intensity distribution in each zone. When a zone represents the road, the standard deviation of its pixels intensities is small. In opposition, when a car covers the zone, this standard deviation is big. By imposing a threshold on the standard deviation of intensities we were able to eliminate the false positive detections. The features surviving all the conditions are finally labeled as 'vehicle features' and are kept in the set  $S$  of features to track.

This scheme proved to be efficient for detecting and tracking all cars moving faster than the camera. In some particular cases where a car tried to pass the camera car but stayed in its blindspot, our system failed. This is due to the fact that in this case the threatening car was moving relatively backward or at the same speed. To solve this issue we decided to keep tracking all the features labeled as 'vehicle features' even if their motion does not respect the conditions established before. In fact we only stop tracking those features in the case they disappear from the image or move with a strict horizontal movement that implies the car is changing lanes.

### 3 Application and results

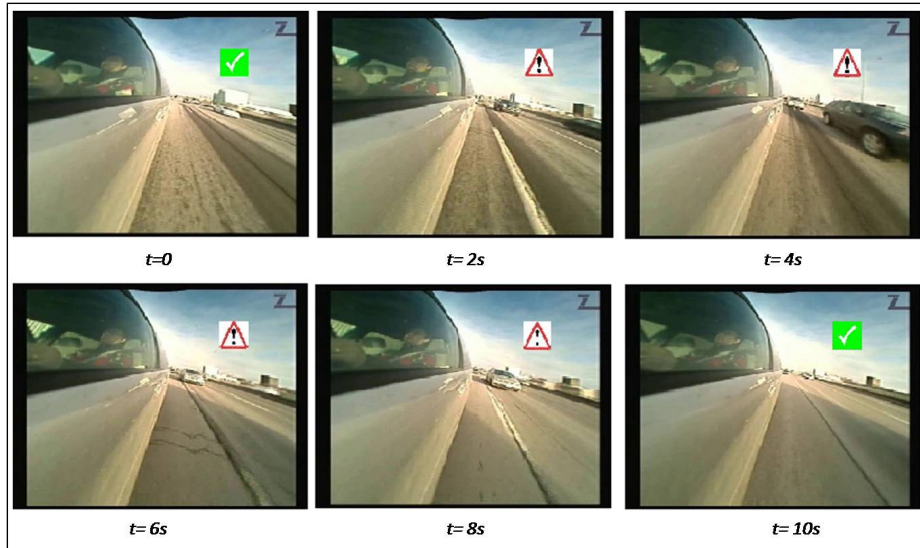
This algorithm was applied to video feeds captured using a commercial CMOS camera fixed on the side mirror of a car moving on a highway. The algorithm was implemented in C++ using the OpenCV library and an Intel Core2 quad processor. The features extraction task, done every 8 frames, took 100 ms to be accomplished. The optical flow calculation and all the filtering steps took 20 ms of calculation time/frame. For 10 000 frames captured, 41 situations of a car presence in the blindspot were encountered. For these 41 situations we were able to detect and track cars in 38 situations without any false positive result. The first situation we missed is caused by the fact that the threatening car passed under a bridge so it was covered by shadow and its tracking was lost momentarily but was detected afterwards as a new threat. The other two missed cars are cars the driver tried to pass by without success. For this particular situation, the cars had a permanent relative backwards movement and we have to find a new approach for this particular case. Overall the results are very satisfactory (Figure 4) and proved that we are able to detect a risky situation fast and with a simple algorithm. Our approach can also be qualified as generic as our system was able to detect different car types (Figure 5) in opposition to a classification approach where samples of each type have to be included in the learning database.

### 4 Conclusion

In this paper we presented a new system for blindspot monitoring, intended to be implemented in a car's safety system. Our challenge was therefore to use a simple and fast algorithm but efficient at the same time. We managed to avoid a complex learning - classification technique and came up with a generic solution to detect any type of cars without the need of any database. A car presence in the blindspot was detected based on its features motion patterns. We actually applied a features tracking using optical flow calculation and added some filtering steps to make sure we do not have any false positive detection. By applying these filters we also managed to reduce the number of features to track and as a result the calculation time. The first results were encouraging but we still have to find a solution for a rare particular case. Complexity wise, the algorithm we presented here is simple and fast and can be easily tuned and adapted which makes our system a good candidate to be ported on any microchip as a part of a real time automotive safety system.

### References

1. Alvarez, S., Sotelo, M.A., Ocana, M., Llorca, D.F., Parra, I.: Vision-based target detection in road environments. In: WSEAS VIS08 (2008)
2. Balcones, D., Llorca, D., Sotelo, M., Gaviln, M., lvarez, S., Parra, I., Ocaa, M.: Real-time vision-based vehicle detection for rear-end collision mitigation systems 5717, 320-325 (2009)



**Fig. 4.** Blindspot presence detection and tracking result: The images show an example of risky situation detection; At  $t = 0$  when no car is present in the lane next to the driver no alarm is emitted. As soon as a car enters the zone ( $t = 2s$ ) the system was able to detect it. Another car enters the zone ( $t = 4s$ ) and stays in until the first one disappears ( $t = 6s$ ) The alarm in that case was still valid. We only declare the zone safe (no alarm) when the two cars disappear from the lane ( $t = 10s$ )



**Fig. 5.** Blindspot monitoring: The algorithm is efficient in detecting cars of different types and sizes (SUV, truck, standard etc.).



3. Batavia, P., Pomerleau, D., Thorpe, C.: Overtaking vehicle detection using implicit optical flow. In: Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on. pp. 729–734 (Nov 1997)
4. Bertozzi, M., Broggi, A., Fascioli, A., Nichele, S.: Stereo vision-based vehicle detection. In: Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE. pp. 39–44 (2000)
5. Betke, M., Haritaoglu, E., Davis, L.S.: Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications* 12, 69–83 (2000), <http://dx.doi.org/10.1007/s001380050126>, [10.1007/s001380050126](http://dx.doi.org/10.1007/s001380050126)
6. Chang, W.C., Hsu, K.J.: Vision-based side vehicle detection from a moving vehicle. In: System Science and Engineering (ICSSE), 2010 International Conference on. pp. 553–558 (2010)
7. Chen, C., Chen, Y.: Real-time approaching vehicle detection in blind-spot area. In: Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on. pp. 1–6 (2009)
8. Chern, M.Y.: Development of a vehicle vision system for vehicle/lane detection on highway. In: 18th IPPR Conf. on Computer Vision, Graphics and Image Processing. pp. 803–810 (2005)
9. Collado, J., Hilario, C., de la Escalera, A., Armingol, J.: Model based vehicle detection for intelligent vehicles. In: Intelligent Vehicles Symposium, 2004 IEEE. pp. 572–577 (2004)
10. Cui, J., Liu, F., Li, Z., Jia, Z.: Vehicle localisation using a single camera. In: Intelligent Vehicles Symposium (IV), 2010 IEEE. pp. 871–876 (2010)
11. Diaz Alonso, J., Ros Vidal, E., Rotter, A., Muhlenberg, M.: Lane-change decision aid system based on motion-driven vehicle tracking. *Vehicular Technology, IEEE Transactions on* 57(5), 2736–2746 (2008)
12. Franke, U., Joos, A.: Real-time stereo vision for urban traffic scene understanding. In: Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE. pp. 273–278 (2000)
13. Jeong, S., Ban, S.W., Lee, M.: Autonomous detector using saliency map model and modified mean-shift tracking for a blind spot monitor in a car. In: Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on. pp. 253–258 (2008)
14. Kato, T., Ninomiya, Y., Masaki, I.: An obstacle detection method by fusion of radar and motion stereo. *Intelligent Transportation Systems, IEEE Transactions on* 3(3), 182–188 (Sep 2002)
15. Kim, S., Oh, S.Y., Kang, J., Ryu, Y., Kim, K., Park, S.C., Park, K.: Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on. pp. 2173–2178 (2005)
16. Labayrade, R., Royere, C., Gruyer, D., Aubert, D.: Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner. *Autonomous Robots* 19, 117–140 (2005), <http://dx.doi.org/10.1007/s10514-005-0611-7>, [10.1007/s10514-005-0611-7](http://dx.doi.org/10.1007/s10514-005-0611-7)
17. Llorca, D.F., Sanchez, S., Ocaa, M., Sotelo, M.A.: Vision-based traffic data collection sensor for automotive applications. *Sensors* 10(1), 860–875 (2010), <http://www.mdpi.com/1424-8220/10/1/860/>
18. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. pp. 674–679 (1981)

19. Mar, J., Lin, H.T.: The car-following and lane-changing collision prevention system based on the cascaded fuzzy inference system. *Vehicular Technology, IEEE Transactions on* 54(3), 910 – 924 (May 2005)
20. Negri, P., Clady, X., Prevost, L.: Benchmarking haar and histograms of oriented gradients features applied to vehicle detection. In: *ICINCO-RA* (1). pp. 359–364 (2007)
21. She, K., Bebis, G., Gu, H., Miller, R.: Vehicle tracking using on-line fusion of color and shape features. In: *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. pp. 731 – 736 (2004)
22. Shi, J., Tomasi, C.: Good features to track. In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. pp. 593 – 600 (1994)
23. Song, K.T., Chen, C.H., Huang, C.H.C.: Design and experimental study of an ultrasonic sensor system for lateral collision avoidance at low speeds. In: *Intelligent Vehicles Symposium, 2004 IEEE*. pp. 647 – 652 (2004)
24. Techmer, A.: Real time motion analysis for monitoring the rear and lateral road. In: *Intelligent Vehicles Symposium, 2004 IEEE*. pp. 704 – 709 (2004)
25. Tsai, L.W., Hsieh, J.W., Fan, K.C.: Vehicle detection using normalized color and edge map. In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. vol. 2, pp. II – 598–601 (2005)
26. Wang, Y.K., Chen, S.H.: A robust vehicle detection approach. In: *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. pp. 117 – 122 (2005)
27. Wender, S., Dietmayer, K.: 3d vehicle detection using a laser scanner and a video camera. *Intelligent Transport Systems, IET* 2(2), 105 –112 (2008)
28. Wu, B.F., Chen, C.J., Li, Y.F., Yang, C.Y., Chien, H.C., Chang, C.W.: An embedded all-time blind spot warning system. In: Zeng, Z., Wang, J. (eds.) *Advances in Neural Network Research and Applications, Lecture Notes in Electrical Engineering*, vol. 67, pp. 679–685. Springer Berlin Heidelberg (2010)
29. Wu, B.F., Chen, W.H., Chang, C.W., Chen, C.J., Chung, M.W.: A new vehicle detection with distance estimation for lane change warning systems. In: *Intelligent Vehicles Symposium, 2007 IEEE*. pp. 698 –703 (2007)