ISSN 1870-4069

Obstacle Detection and Trajectory Estimation in Vehicular Displacements based on Computational Vision

Lauro Reyes-Cocoletzi, Ivan Olmos-Pineda, J. Arturo Olvera-López

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación, Puebla, Mexico
lauro.reyesc@alumno.buap.mx, {iolmos,aolvera}@cs.buap.mx

Abstract. Obstacle detection and trajectory estimation in vehicular environments is an open problem in autonomous vehicles development. The automotive industry has made significant progress in research and development of tools; however, there are still challenges to overcome and opportunity areas to be exploited in order to achieve full autonomy in vehicles. This paper presents an analysis of different methods proposed for obstacle detection and trajectory estimation, leading into a proposal of solution for solving the problem of trajectory estimation based on computer vision techniques. This proposal covers the context of traffic environments in Latin America, where basic signage (such as the dividing lines of the road) is absent or not easily visible, among other typical characteristics of countries (like Mexico and others) where the infrastructure for maintaining safe road conditions (speedbumps, potholes) is limited.

Keywords: Obstacle Detection, Trajectory Estimation, Autonomous Vehicles, Computer Vision.

1 Introduction

Over the years, the automotive industry has made efforts to provide vehicles with autonomous displacement (autonomous taxis, autonomous private cars). Recently, cars have been equipped with Driver Assistance Systems (ADAS), but their applications are limited for specific conditions and controlled environments; therefore, research in this area is incomplete. A vision-based DAS consists of modules for obstacle detection and recognition, obstacle tracking and displacement prediction [13].

It is important to clarify that the meaning of an obstacle for the purposes of this research is: an above-ground object that presents a collision risk when the vehicle is traveling along a road. Note that the basic functions for autonomous movement are obstacle detection and tracking in traffic environments.

The information provided by the automotive industry is limited [2] due to the commercial implications. However, it should be noted that research work has been carried out not only in private business, but also in universities and academic research centers.

Some limitations of research into autonomous displacement are: low performance in terms of detection rate [3], a limited number of objects perceived as obstacles (pedestrians and vehicles only) [6], limited information acquired from the environment (sensors) [13], the high cost of sensors, in some cases (laser radar), and high processing runtimes [1], among other issues.

Video cameras commonly provide information about the environment, these devices are low cost, do not emit signals into the environment (electromagnetic or sonic radiation), and these do not suffer from frequency interference. Video cameras are used in this research due to the aforementioned advantages over other sensors (sonic or laser radar systems), additionally, using a stereoscopic computational system provides spatial information from the environment (depth and position of objects).

This paper is organized as follows: Section 2 describes some methods performed in academic research centers, Section 3 shows the results obtained in related research works. Section 4 discusses an initial stage of the proposed solution. Section 5 presents some initial experiments with respect to the proposal, Section 6 briefly analyzes the main problem of tracking objects in realistics scenarios. Section 7 offers a brief conclusion about the state of the research and the work to be developed.

2 Detection and Estimation of Obstacle Trajectory

In general, methods that address the problem of obstacle trajectory detection and estimation are diverse and depend on the type of sensor used and information obtained from the environment. The most common sensors are ultrasonic, radars, laser radars and video cameras (stereoscopic or monocular).

Due to the amount of information obtained from a digital image (color, texture, resolution, spatial position, etc.), computational vision techniques are similar to environmental human perception [13].

For these reasons, it was decided to work with the information captured on video in order overcome the detection rates of recent works compared against the use of other types of sensors (laser radars).

There are several strategies for detecting and monitoring obstacles via computational vision; some of these methods are mentioned briefly in following paragraphs.

Algorithms based on convolutional neural networks [8] for recognition of cars and pedestrians classifies these into three collision categories (high risk, low risk, medium risk) by means of a deep learning system that considers multiple sources of local pattern information and depth in the scene.

Techniques based on stereo vision produce a dense disparity map, which is used as the input for adaptive disparity algorithms U-V [11] when mapping the regions of interest (ROI) and the consequent in-scene spacial obstacle localization.

Evolutionary algorithms (EA) aim to detect multiple obstacles in a video through the use of a genetic regression algorithm (GA) and prevent false detection through the perception of movement [15]. This basis method is based on the operation of the fitness function with the heuristic parameters, which means choosing the optimal fitness parameters using EA and the least squares method.

Table 1 presents some reference works in this area: the methods implemented to solve the problem are reported, as well as the test sets, the hardware, the software used, and the results obtained, such as detection rate (DR), mean average precision (mAP), frames per second (fps), or multi-objective tracking accuracy (MOTA).

Table 1. Quantitative results and tools used in some related works about obstacle detection.

Reference	**Method**	**Technique (detection / tracking)**	**Quantitative results**	**Test set**	**Software hardware**
[17]	Stereo map	-Disparity maps -Stereo comparison -Local binary patterns (LBP) -Gradient histogram	-DR = 82.5 % -mAP = 79.5 -fps = 14	-KITTI Benchmark [5] -HCI EISAT	-CPU xeon 2.67 GHz -Tesla K40 GPU
[20]		- Conditional random field -Trajectory model	-MOTA = 58.9 % -Recall = 69.2 % -Accuracy = 88 %	-Training: 490 frames 1,578 pedestrians labeled resolution 640 x 480 pixels	-nVidia GeForce 8800
[23]	CNN	-LBP -K-medias -Classifiers -Local trust patterns	(Optimal lighting conditions) -DR = 77 % -Fault alarm rate (FA) FA = 1.25 % (variable lighting conditions) -DR = 50.8 %, FA = 1 % -mPA = 63.4 % , fps = 45	-KITTI Benchmark -Daimler dataset: 15,560 pedestrian samples -21,790 images with 5692 pedestrians labeled	-CPU Intel-xeon -Tesla K40 GPU
[6]		-Fast RCNN -Polynomial prediction	-Regions of trust with 95 % -Risk of estimated collision higher than 50 % - Estimated Travel to 8.6 m at a speed 50 km / hr -50 fps	-150 pedestrian paths -They are divided into three different classes: Begin: consists of 50 scenes of approx. 1 sec Stopped: consists of 50 scenes, 1 sec Walking: consists of 50 movements	2 Monochromatic cameras resolution 1920x1080 pixels
[10]	Deep learning	-Progressive convolutional layers -Decoding stage	-DR =81.5 to 90.6 % depending on the class -mAP = 78.6 % -Runtime 110 ms average	-PASCAL VOC 2007 -40 k iterations -COCO benchmark	-GPU TitanX -Caffe
[18]		-Deconvolutional detectors	-MOTA = 11.8 % -fps = 87.9	-DETRAC 60 sequences of training. -40 test sequences.	-AlexNet -Caffe -Titan Black GPU -iiRAV
[15]	Evolutionary algorithms	-Genetic algorithm -Function fitness -Hypothesis of generation (HOG)	-Value fitness = 0.1216 to 0.4315 -Processing time = 48 ms -DR = 96.08 %	-3025 images captured in a controlled set and another real environment vel. from 20 to 70 km/hr	-FPGA xilinx

Table 1 also shows important research parameters for each technique; it can be seen that the detection rate is close to 100% in some cases, however, the total number of obstacles detected must be taken into consideration, as well as the class or type (pedestrian, vehicle, animals, etc.), which in many cases only detect limited classes. To track the obstacles, even the MOTA parameter is below 100 % as the error with respect to the next position or movement in the video increases due to the complexity and quantity of data to be processed, the algorithm processing speed, the object's movement speed, and the number of objects present on the scene among others.

In order to test the approaches proposed in the literature, some well-known datasets are used, such as the KITTI Vision Benchmark suite from the Karlsrushe Institute of Technology, datasets are captured in urban areas on highways, and up to 15 cars and 30 pedestrians are visible per image [5].

The PASCAL Visual Object Classes Challenge also has data blocks available to researchers [4], while the MOT challenge benchmark provides ready to use information [12]. COCO is a large-scale object detection, segmentation, and labelling dataset [9], the UA-DETRAC Benchmark Suite is a multi-object tracking benchmark whith a dataset consisting of 10 hours of surveillance cameras video at 24 different locations in Beijing, China [7].

Depending on the method to be used, there are some limitations; a summary is provided in the following section.

3 Analysis of Related Works

The method proposed in the literature has some positive aspects, but also some limiting factors. Information about the deficiencies found in related works is covered below, including detection, tracking, and estimation issues.

With respect to the mapping ROI models: the problem of trajectory prediction has areas of opportunity such as detection failures, objects with similar appearances, occlusions, and variations in illumination and points of view [11, 20]. Only two kinds of obstacles (vehicles and pedestrians) are taken into consideration [3, 11, 17, 20, 19]. High processing runtime rate and low obstacle detection rate [3, 14]. Monocular mapping (ROI), in some cases acquires data through static surveillance cameras [11].

For the Probability models case: trajectory estimation methods with a high uncertainty have problems with complexity (with many samples) or low accuracy (with an insufficient number of samples) [1, 19, 22]. Kalman filter algorithms and particle filters used for the purpose of tracking obstacles suffer from drift problems caused by changes in appearance [19].

Deep learning models have the following characteristics: they only consider characteristics such as color or depth, which implies a limitation to obtaining a higher level of abstraction of the representative characteristics of the obstacles [23, 18, 21]. Tracking methods based on online-classifiers learning suffer the problem of error accumulation during the self-learning process [21, 16]. Some algorithms based on the convolutional neural networks approach only capture translational invariances and not the rotational invariance or out-of plane rotation, which makes them susceptible to error when classifying and identifying obstacles [6, 23, 16].

Evolutionary algorithm models tend to fail in object classification, as they confuse similar forms, which implies an erroneous obstacle detection, for example, detecting a car in the scene when it is not in fact in the image [15].

The general stages that an algorithm must follow in order to solve the problems of obstacle detection, classification, and tracking as well as of trajectory prediction, obtained from related works, are presented in Table 2 below.

Table 2. Problem stages addressed by different authors.

			Paper reference															
#	Stage		[3]	[5]	[7]	[10]	[11]	[12]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
I	Selection ROI			✓	✓	✓		✓	✓		✓	✓		✓	✓			✓
II	Spatial position estimation		✓	✓	✓		✓		✓		✓			✓	✓	✓	✓	
III	Object detection			✓	✓	✓	✓	✓	✓				✓		✓	✓	✓	
IV	Feature extraction	Training								✓			✓	✓			✓	
		Neighborhood Mapping			✓		✓	✓			✓	✓			✓			
V	Classification	Specific class											✓		✓		✓	
		Obstacle No obstacle	✓	✓	✓			✓	✓	✓	✓	✓		✓				
VI	Rank (danger level)				✓		✓	✓	✓		✓			✓			✓	
VII	Tracking	Multiple obstacle				✓									✓			
		Unique obstacle	✓				✓				✓		✓	✓				
VIII	Trajectory estimation				✓								✓				✓	

From these stages and the associated review of the works, we can obtain an approximation of the strengths of the work done until now and the work to be solved. In the table 2 we can see the distribution of related works and points (stages) that have less development progress. Based on this information, it can be affirmed that the pending tasks are tracking and trajectory estimation.

Obstacle classification has been developed and has reported rates greater than 90%; however, it still requires a class for obstacles other than pedestrians and vehicles, this is due to the fact that peculiar obstacles can be found on the roads. Figure 1 shows obstacles that are common to find on Latin American roadways, but difficult to identify with current obstacle recognition/detection models.

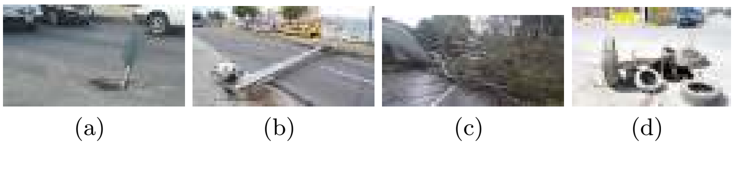
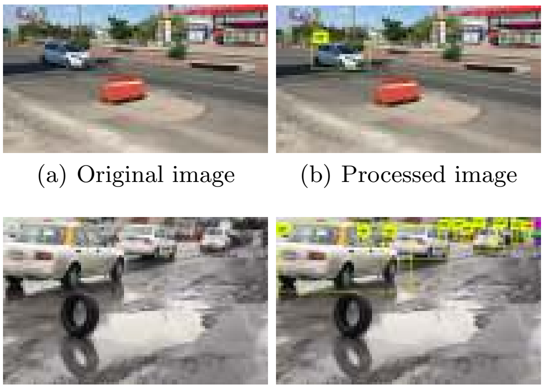


Fig. 1. Obstacles present in road, a) road sign in sewer, b) light pole in the middle of the street, c) fallen tree blocking the street, d) and tires in sewer.

In order to observe the performance of obstacle detection in Latin American environments, a test was performed with a deep learning algorithm (designed by Rendom and Divvala [16]).

Figure 2 shows that the algorithm cannot detect the tire or the road barrier as obstacles blocking the road. To date, the different models and computational techniques have failed to detect these types of obstacles in the images.



(a) Original image (b) Processed image

(c) Original image (d) Processed image

Fig. 2. Original and processed images for obstacle detection.

This is due to the fact that in some cases the algorithm focuses on detecting the objects (automobile) in the scene through previously known and defined characteristics. However, objects dis-similar to the known detection characteristics are not detected or identified.

In the following paragraphs the contributions that can be made to the research problem are discussed; this is a challenge due to the amount of information to be processed and the complexity of the tasks to be performed:

Multiple-obstacle tracking. The inference method approach for trajectory and the on-scene prediction of multiple obstacles based on a Gaussian probability distribution. This is possible due to roads being visualized as a structured environment, where the driver's infinite number of possible movements can be approximated by a limited number of maneuvers.

Block trajectory estimation. Trajectory detection and estimation in unstructured vehicular environments, that is, cities with problems such as potholes, a lack of signs, speedbumps without height allowed, abrupt lane changes by drivers, jaywalking pedestrians, cyclists or motorcyclists cutting lanes, or animals crossing the road, among others.

Feature extraction. Occlusion when tracking real scenarios in traffic environments. Training of the learning model should include some sets with partially hidden objects that consider the disparity map to complete the missing information.

Classification. Characterize cars, pedestrians, cyclists, animals or (foreign) objects that obstruct the road into-pattern classes. On Latin American roadways it is common to find animals crossing or traveling next to vehicles, cars cutting lanes, cyclists ridding very close to cars, or even tires or trash cans to signal a damaged manhole.

4 Obstacle Detection and Trajectory Estimation Proposal

The initial proposed system is structured into stages; the flowchart of the stages is provided below (Figure 3) and a description of each is provided in the following paragraphs.

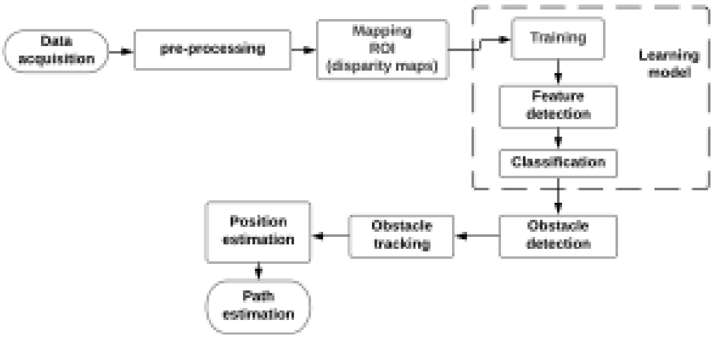


Fig. 3. Flow diagram method for obstacle detection and trajectory estimation.

The initial step is environment perception by means of a pair of cameras (stereoscopic vision) together with the preprocessing.

Among the typical preprocessing stages used to improve image quality are the application of noise filters, contrast enhancement, histogram equalization, and image scaling.

The fundamental step of the detection stage is to perform disparity mapping to simplify the information to be processed from the RGB channel intensity levels in the image. Disparity mapping [17] is used to obtain basic characteristics for the representation of objects in a scene in complex vehicular environments, it also has the advantage of providing spatial depth information in the scene.

The ROI obtained in the disparity map is filtered by means of edge detection (Shi-Thomasi algorithm), horizontal or vertical pattern detection (Haar algorithm) and even LBP [13] to obtain better defined regions and disparity maps.

The subsequent stages include processing based on learning models, by means of which it is possible to classify obstacles perceived in the scene, and identify obstacle classes (it is proposed to first identify cars, pedestrians, animals, and consider additional kinds of obstacles in a future work).

The learning system includes a training block, by means of a learning database. In training, abstract features must be learned in the manner least sensitive to rotational invariance, out-of-plane rotation, and reduced sensitivity to noise, that is, a deep learning system [8].

The training block aims to provide a sample set of instances identified to learn and classify the characteristics in specific categories such as: cars, pedestrians or dogs. Prior to this, an exploration process is carried out in order to determine if the analyzed object is an obstacle.

The proposal model implies representation of the elements with n-levels of abstraction to determine the n-representative characteristics of these elements. These parameters are used to define the number of instances (database). For example, in the case of cars body color, shape, or texture or some other characteristics might be considered. In the case of pedestrians, there is skin color, posture, body shape (2 upper extremities, 2 lower extremities), etc. as well as other yet to be identified obstacles.

The obstacles must be followed from the n-frame to the n+1 frame and on the consecutive frames of the video. The optical flow algorithm [3] is proposed in order to solve the tracking block, using the U-V disparity information. It is worth mentioning that Kalman filter and particle filter algorithms have been successfully used to track vehicles, although they present performance drawbacks (as noted in section 3).

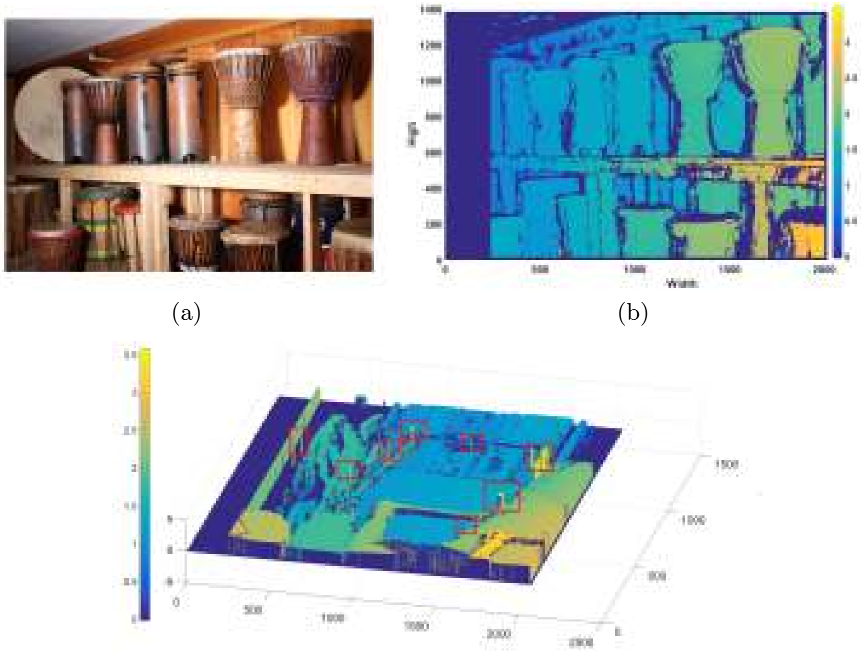
Finally, current position estimation (in the image) and trajectory estimation are connected; these can be denoted as the algorithms whose task is to track multiple objects. The novel proposal to solving the trajectory estimation problem is through tracking by detection in order to calculate the estimation of the next position of the obstacles in vehicular traffic. This was first addressed as a problem of estimation probability; the object trajectories can be considered as a group of points moving simultaneously (in space and time). Based on the previous position and route information followed by the tracking stage, a trajectory hypothesis was established. The trajectory hypothesis was parameterized with a Bayesian network and a proposed mathematical model based on statistical probability distribution (a trajectory estimation model is in development).

5 Experimental Results

Experiments are performed to obtain initial results with respect to the proposal and determine its feasibility.

The stages carried out in the experiments include the object detection stage (disparity maps experiment) and the object tracking stage (optical flow experiment).

The disparity method is a first approximation of the objects in the scene; however, it is necessary to first detail the obstacle detection results. Figure 4a shows the original image, while Figure 4b shows the disparity map and the values calculated with the stereo correspondence. Blue tones indicate objects at the back of the scene and yellow tones those closer to the camera. The matrix of values obtained are from 0 to 3.5 units. Value 3.5 represents the volume ratio of the objects with respect to the background (zero). Figure 4c shows a view of the disparity map that serves to observe the regions in 3d; this volume is constructed using the data obtained, which allows segmenting the background and objects. It is possible to observe scene-depth levels of the objects present with respect to the background. In addition, some undesirably high values can be observed, which must be eliminated (maximum peaks).



(a)

(b)

(c)

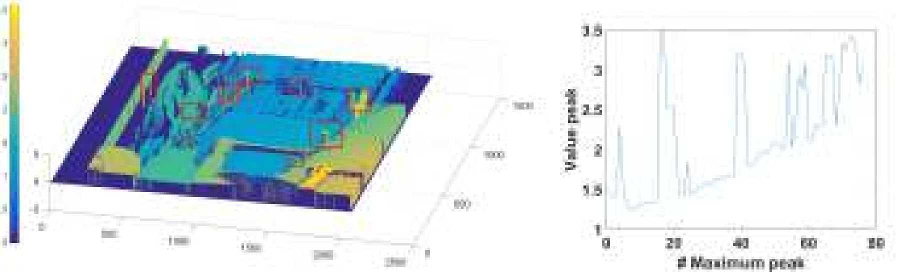
Fig. 4. a) Original image, b) dense disparity map, and c) side view disparity map.

Maximum peaks values are presented in figure 5, based on the number of peaks to the number of pixels, error estimation can be calculated and the disparity maps obtained. With the number of maximum peaks found against the total number of values obtained, a percentage of error in obtaining the disparity maps can be found:

$$Error\_rate(\%) = \frac{\# \quad maximun \quad peaks}{\# \quad values \quad obtained} \times 100. \tag{1}$$

Table 3 shows some data regarding this error, these errors (maximum peaks) can be eliminated by means a cutting filter (from atypical data). Otherwise, the selectable regions of interest must be composed of a block of pixels sufficiently large to discard this type of outlier.

At this stage, the actual distance of the objects in relation to the data obtained from the disparity map still needs to be parameterized.



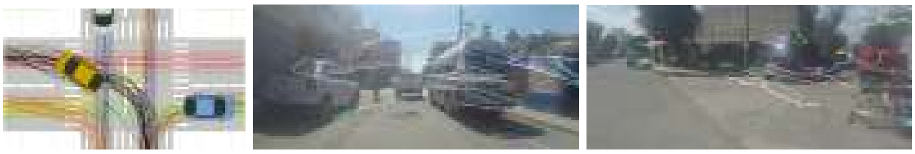
(a) (b)

Fig. 5. a) Disparity map (peaks in red rectangle) and b) maximum unwanted peaks in the calculations.

Table 3. Analysis of the data obtained disparity maps.

# Disparity map	1	2	3	4	5	6	7	8	9	10	11
# Values obtained	971	1013	1589	1650	1037	860	600	1546	1358	1900	901
# Maximum peaks	61	76	210	150	89	45	89	75	156	341	100
Error_rate (%)	6.28	7.50	13.22	9.09	8.58	5.23	14.83	4.85	11.49	17.97	11.10
Average error (%)										10.01	
Standard deviation error (%)										4.15	

The next experiment consists of tracking the groups of pixels that move together (representing objects) and plotting points from the previous positions for a standard representation of the dense motion (figure 6a). Figure 6b-c shows a traffic scene captured in the real environment of a typical street in Puebla City, Mexico, where public transport, cyclists and vehicles can be observed. Every pixel is labelled by a motion vector, indicating the change in image data from time t to time $t+1$, in such a way that shows the trajectory taken by the detected pixels of interest.



(a) (b) (c)

Fig. 6. a) Theoretical trajectory [1], b), and c) tracing approximate trajectory points using an optical flow algorithm in a real vehicular environment.

It is necessary to define obstacles regions to be followed, since the pixel displacement information (analyzed in this experiment) provides only a preliminary approximation of local regions in the image where the obstacles must be detected. This involves reference corner detection (Shi–Tomasi algorithm) and tracking of same (Lucas–Kanade optical flow algorithm) [12].

The characteristics to take into consideration in the process are the following. Shi–Tomasi corner detection parameters:

- Maximum number of corners.
- Block size: size of the neighborhood considered for corner detection.
- Minimal distance: distance between detected corners.

Lucas–Kanade optical flow parameters:

- Max level: 0-based maximum pyramid level number.
- Window size: size of the search window at each pyramid level.
- Count: maximum number of iterations or elements to compute.

The tracking rate is below 50% given that regions of interest for object detection are not yet limited.

Table 4. Object tracking parameters.

Shi-Thomasi corner detector	# Max corners	Block size	Minimum distance (# pixels)
	50 to 100	7 to 18	7 to 50
Lucas-Kanade optical flow	# Max level	Window size	Count
	3	15x15	12

The work to be done is to implement a form descriptor that adapts to any type of irregular object present on the road that represents an obstacle to displacement, so that whatever the obstacle while having volume will be detected. The features to extract (object descriptors) must be rotation invariant i.e., to the non-linear deformation transformation (perspective). In addition, they must be discriminant in order to distinguish objects of different classes and robustness levels so that they are insensitive to variations in noise and illumination.

Figure 7 shows the common descriptors used for the basic representation of objects.

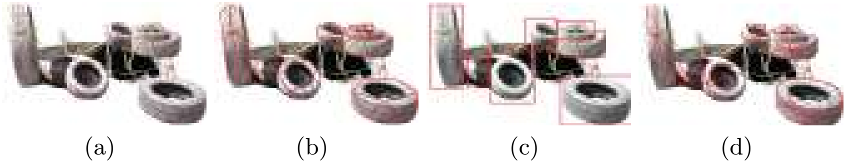

(a) (b) (c) (d)

Fig. 7. Representation of shape descriptors for the detection of objects a) cloud points, b) polygonal approximations, c) bounding box, and d) skeleton.

The typical databases used in the related works are being considered for verification and validation of initial results [5, 4, 12, 9, 7]; however, related works consider controlled environments, so the contribution focuses on uncontrolled environments, such as those found in Latin America.

Finally, these experiments serve as a basis for determining the requirements in terms of the amount of data to be processed, the scenes to be taken into consideration, the infrastructure of the video capture equipment, the traffic schedules, routes, and safety conditions when gathering the data. Similarly, it should be noted that the lighting conditions or the number of vehicles or objects on the route of travel are not controlled.

6 Discussion

In this section, the main problems that arise when tracking objects in realistic scenarios are briefly analyzed. These include the location of objects as they undergo occlusion and the constant tracking of unique object trajectories as they are viewed through multiple cameras.

Occlusion. The partial occlusion of an object by a structure in the image is difficult to detect, since it is not always possible to differentiate between the object changing shape and the becoming occluded. A common approach to handling complete occlusion during tracking is to model the object's motion by either linear or nonlinear models and, in the case of occlusion, to continue predicting the object's location until it reappears [11, 20]. The chance of occlusion can be reduced by means of an appropriate selection of camera positions. For instance, if the cameras are mounted on vehicles, that is, when a birds-eye view of the scene is available, occlusions between objects on the ground occur less frequently.

Stereo camera tracking. The need for multiple cameras for tracking arises for two reasons. The first is the use of depth information for tracking and occlusion resolution [3]. The second an increase to the area under view, since it is not possible for a single camera to observe large areas because of a finite sensor field of view [11]. Performance depends greatly on how closely the objects follow the established paths and the expected time intervals across cameras [17]. For scenarios in which spatio-temporal constraints cannot be used, for example, objects moving arbitrarily in the non-overlapping region, only the tracking by recognition approach can be employed, which uses the appearance and the shape of the object to recognize it when it reappears in the camera's view.

Significant progress has been made in object tracking over the last few years, and several robust algorithms have been developed which can track objects in real time in simple scenarios. However, it is clear that the assumptions used to make the tracking problem manageable, for example, smoothness of motion, minimal occlusion, illumination constancy, high contrast with respect to background, etc., are violated in many realistic scenarios and therefore limit a trackers usefulness in vehicle navigation applications. Thus, tracking and associated feature selection, object representation, dynamic shape, and motion estimation problems are very active research areas and new solutions are continuously being proposed.

7 Conclusions

This paper briefly describes the work related to obstacle detection in real environments, presents a general description, analyzes the existing techniques and how they have proposed to solve the problem.

The proposed model for estimating trajectories through probability methods is focused on determining the future displacement of the detected obstacles within a confidence range. The proposed solution establishes a theoretical overview of the solution; the work to be done is extensive and the solution model is in development. Future work involves implementing the model stages and obtaining performance parameters to compare with the existing research.

From this work, it can be concluded that the problem of obstacle detection is an area of significant interest and that there is still room for improvement and opportunities for innovation. There is much research to be done and it is worth continuing with the development of techniques and tools to solve this problem. A large amount of data must usually be processed, so the computing aspects (memory, processing speed) must be robust to meet the needs of operating close to real time.

Acknowledgments. The first author thanks the support provided by the CONACYT scholarship number 708553.

References

1. Barth, A., Siegemund, J., Meißner, A., Franke, U., Förstner, W.: Probabilistic multi-class scene flow segmentation for traffic scenes. In: Joint Pattern Recognition Symposium. pp. 503–512. Springer (2010)
2. Dikmen, M., Burns, C.M.: Autonomous driving in the real world: Experiences with tesla autopilot and summon. In: Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications. pp. 225–228. ACM (2016)
3. Dur, E.: Optical flow-based obstacle detection and avoidance behaviors for mobile robots used in unmaned planetary exploration. In: 4th international conference on recent advances in space technologies. pp. 638–647. IEEE (2009)
4. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision 88(2), 303–338 (2010)
5. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research 32(11), 1231–1237 (2013)
6. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE international conference on computer vision. pp. 1134–1142 (2015)

7. Lee, B., Erdenee, E., Jin, S., Nam, M.Y., Jung, Y.G., Rhee, P.K.: Multi-class multi-object tracking using changing point detection. In: European Conference on Computer Vision. pp. 68–83. Springer (2016)
8. Li, P., Mi, Y., He, C., Li, Y.: Detection and discrimination of obstacles to vehicle environment under convolutional neural networks. In: Chinese Control And Decision Conference (CCDC). pp. 337–341. IEEE (2018)
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
10. Lu, K., An, X., Li, J., He, H.: Efficient deep network for vision-based object detection in robotic applications. Neurocomputing 245, 31–45 (2017)
11. Mane, S.B., Vhanale, S.: Real time obstacle detection for mobile robot navigation using stereo vision. In: International Conference on Computing, Analytics and Security Trends (CAST). pp. 637–642. IEEE (2016)
12. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
13. Mukhtar, A., Xia, L., Tang, T.B.: Vehicle detection techniques for collision avoidance systems: A review. IEEE Transactions on Intelligent Transportation Systems 16(5), 2318–2338 (2015)
14. Nguyen, V.D., Nguyen, D.D., Lee, S.J., Jeon, J.W.: Local density encoding for robust stereo matching. IEEE Transactions on Circuits and Systems for Video Technology 24(12), 2049–2062 (2014)
15. Nguyen, V.D., Nguyen, T.T., Nguyen, D.D., Lee, S.J., Jeon, J.W.: A fast evolutionary algorithm for real-time vehicle detection. IEEE Transactions on Vehicular Technology 62(6), 2453–2468 (2013)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
17. Roggeman, H., Marzat, J., Derome, M., Sanfourche, M., Eudes, A., Le Besnerais, G.: Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2090–2099 (2017)
18. Sangineto, E., Nabi, M., Culibrk, D., Sebe, N.: Self paced deep learning for weakly supervised object detection. IEEE Transactions on pattern analysis and machine intelligence 41(3), 712–725 (2018)
19. Schulz, J., Hubmann, C., Löchner, J., Burschka, D.: Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction. In: 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 1467–1474. IEEE (2018)
20. Song, W., Yang, Y., Fu, M., Qiu, F., Wang, M.: Real-time obstacles detection and status classification for collision warning in a vehicle active safety system. IEEE Transactions on intelligent transportation systems 19(3), 758–773 (2017)
21. Wang, J.G., Zhou, L., Pan, Y., Lee, S., Song, Z., Han, B.S., Saputra, V.B.: Appearance-based brake-lights recognition using deep learning and vehicle detection. In: Intelligent Vehicles Symposium (IV). pp. 815–820. IEEE (2016)
22. Yoon, J.H., Yang, M.H., Lim, J., Yoon, K.J.: Bayesian multi-object tracking using motion context from multiple objects. In: Winter Conference on Applications of Computer Vision. pp. 33–40. IEEE (2015)
23. Yu, H., Hong, R., Huang, X., Wang, Z.: Obstacle detection with deep convolutional neural network. In: Sixth International Symposium on Computational Intelligence and Design. vol. 1, pp. 265–268. IEEE (2013)