

On-road Vehicle Detection based on Hierarchical Clustering and Adaptive Vehicle Localization

Moslem Mohammadi Jenghara

Department of Electrical and Computer Engineering, University of Kashan, Kashan, Iran
m.mohammadi.14@gmail.com

Hossein Ebrahimpour Komleh*

Department of Electrical and Computer Engineering, University of Kashan, Kashan, Iran
ebrahimpour@kashanu.ac.ir

Received: 16/May/2015

Revised: 22/Sep/2015

Accepted: 04/Oct/2015

Abstract

Vehicle detection is considered to be a significant task in automatic driving which is regarded as a challenge and thorny issue for researchers in this field. The majority of commercial vehicle detection systems are based on radar. However, methods using radar suffer from problems such as the one encountered in zigzag motions. Image processing techniques can overcome these problems. This paper proposed an approach based on hierarchical clustering in which low-level image features are used to detect on-road vehicles. The approach introduced in this study is based on a new clustering method called teammate selection. In this clustering method, a new merging measure based on cluster center distances and gray scale values was introduced. Each vehicle was assumed to be a cluster. In traditional clustering methods, the threshold distance for each cluster was fixed; however, in the method proposed in this paper, the threshold distance is adaptive which varies according to the position of each cluster. The threshold measure was computed with bivariate normal distribution. Sampling and teammate selection for each cluster were carried out by cluster members based on weighted average. Unlike other methods which used only horizontal or vertical lines, a fully image edge detection algorithm was utilized in this study. Corner is an important video image feature which is commonly used in vehicle detection systems. However, Harris features were used in this paper to detect the corners. Furthermore, LISA data set was used to evaluate the proposed method. Several experiments were conducted to investigate the performance of proposed algorithm. Experimental results indicated good performance compared to other algorithms.

Keywords: Adaptive Feature Grouping; Moving Camera Image Processing; Vehicle Detection; Hierarchical Clustering; Teammate Selection Clustering.

1. Introduction

Driver assistance and traffic monitoring systems are of high significance in intelligent vehicles. Object detection and tracking which were observed by ego-vehicle on/around road such as cars, pedestrians and other obstacles are the main requirement to design and implement these systems. Camera is usually located at the center of the front bumper of ego-vehicle car. The performance of detection system on the road is a critical issue for system administrators with respect to security. Hence, systems must be robust enough for various conditions. Vehicle detection is a challenging domain for the committee of intelligent machines. Road environments vary in terms of traffic situations, number of cars, lighting conditions, weather conditions, construction of roads, tunnels and more. Consequently, a fully adaptive and parametric system is required for the existing variable conditions. Near or mid-range vehicle detection is another effective issue in the structure of environment and camera parameters. Various sensors are available which can be used in driver assistance systems such as lidar, radar, ultrasound and embedded cameras.

There are four situations based on camera and object movement which are listed below:

- I. Stationary camera, constant object
- II. Stationary camera, moving object
- III. Moving camera, stationary object
- IV. Moving camera, moving object

The second situation is more practical and more common [1]. A fixed camera located on a highway for speed control is an example of the second situation. However, driver assistance systems, camera and objects are mobile which is regarded as a highly sophisticated condition. Due to background changes and inapplicability of differential techniques, the detection of a moving object with a moving camera is a challenging task.

Most commercial vehicle detection systems are based on radar. It is a sensor which has lots of limitations such as angular constraint and temporal resolution. In general, radar-based systems can detect vehicles located directly in front of the observing car but they cannot detect marginal cars which are located at different angles. In this case, any change in car line may be dangerous. Also, the use of radar-based systems in high zigzag and steep roads can be problematic. In contrast with radar-based systems, it should be noted that cameras are inexpensive, consume little

* Corresponding Author

power and can be easily managed to capture information from the environment. Visual data processing is complex but it provides valuable information about the environment.

In this paper, the researchers focused on low-level features and used them to specify the location of vehicles. The shape and structure of vehicles were not of any significance. Since the intended vehicle detection system is in a dynamic environment and inside an ego-vehicle, both the observing car and other vehicles are moving. The size and position of the devices can vary over time. Thus, an adaptive method based on coordinates can be useful. This adaptability can be utilized in different components of the system including thresholds setting which is discussed in the proposed method.

2. Related Works

Detection of vehicles using embedded camera image processing can be done through various approaches such as learning-based methods.

Carrafi et al. [4] proposed a learning-based approach using waldboost [2]. They introduced a cascading fine-grain detection system similar to the viola-jones [3] method. Waldboost algorithm is implemented to reject negative choices in the first steps which has a significant impact on system speed [4]. Deformable object model learning [5] was proposed for learning and identifying vehicle features as an object model in which latent support vector machine (LSVM) and Histogram of Oriented Gradients (HOG) were combined. Despite high detection efficiency of the above-mentioned method, its computational complexity allows for only 1 fps processing.

Jazayeri et al. [6] used Hidden Markov Model (HMM) to develop a system which could separate vehicles and background from one another; it was also able to probabilistically model motions in terms of scenes based on frame features. Low-level features such as edges and corners which are resistant to light and shape variations were used in [6].

Samadi et al. [7] proposed a multi-agent system for vehicle detection. Each agent in this multi-agent system carries out one part of the diagnostic process. The hypothesis exchange and conflict resolving are done by cooperation among agents. In this system, the following agents were used: edge detection agent, contour agent, vehicle agent, license-plate rectangle detection agent, license-plate line detection agent, wheel detection agent, plate candidate verification agent and symmetry detection agent.

In recent years, approaches based on active learning led to good results on detecting road vehicles [8]. HOG-SVM and Haar-like features with Adaboost classifier, traditional methods in active learning were investigated and compared with each other in terms of time complexity and other parameters [9]. In general, methods based on active learning produce a simple classifier with an easy supervised learning. Then, querying classifier selects informative patterns for retraining classifier again.

Indeed, the query function which selects informative and complex patterns and classifier input for retraining and updating decision boundaries is the main part of active learning [10,11].

A method based on active learning through split and merge was proposed in [12] to determine various components of vehicles. The learned classifier is SVM. PCA, Gabor, Wavelet and combined Gabor-Wavelet features with NN and SVM classifiers were used by Zhang [13] for diagnosis before the occurrence of accident conditions. An augmented Gabor feature for detecting vehicles was used in [14] where Gabor filter parameters were improved and learned by SVM classifier to cover more sub windows containing pieces of vehicles.

Kim et al. [15] used a combination of sonar and vision sensors where lighting and distance conditions had no impact on system accuracy. Sonar sensors for the distance and cameras up to 10 meters were used. Features such as shadows, road lines, horizontal lines and vehicle symmetry were used in image processing. Template matching [16] and vertical symmetry detection [17] are other methods for vehicle detection. Many researchers work on computer vision-based intelligent transportation systems [18].

Most on-road vehicle detection systems detect vehicles in line with the observer. There are many horizontal lines at the back of the vehicles including shadow lines, window lines and top and bottom lines of the vehicle. This feature is relatively stable against light changing and scaling. Matthews et al [19] used edge and vertical lines detection to indicate the left and right margins of vehicle. Image edge is considered as a favorite feature for researchers in the field of vehicle detection [20, 21]. Corners are maintained in their position on vehicles and background; hence, they provide useful information about the different components of an image. Bertozzi et al. [22] proposed a corner-based method for vehicle detection.

Each vehicle is represented by local variation of gray level with a certain texture [23]. Texture regions can be processed further to make accurate detection. Calculating entropy based on neighboring pixels is the criterion for determining texture. Areas with high entropy are selected for further processing [24]. Kalinke [25] used texture to focus algorithm on the areas having lots of information. Shannon introduced local entropy to measure the information of each image patch [26]. Symmetry is an invariant feature at the rear of vehicles which is stationary and stable under different light conditions and scaling. Many researchers use this feature to detect vehicles [27,28]. A symmetry and edge-based vehicle detection method using edge oriented histogram (EOH) and support vector machine (SVM) was proposed in [29] for approximate vehicle location and improving post-processing.

Optical Flow (OF) is an informative motion-based feature which provides information about the direction and speed of moving objects in video frames [6,30,31]. Pixel-based and feature-based methods are two main approaches for optical flow computation [18]. In

computing optical flow, (u,v) feature points in (I_t) and (I_{t+1}) are mapped so that eq.1 is minimized [32, 33].

$$e(dx, dy) = \sum_{x=u-w}^{u+w} \sum_{y=v-w}^{v+w} (I(x, y) - I(x + dx, y + dy)) \quad (1)$$

Shadow is used as a feature for vehicle bounding. The lower part of vehicles in different lighting conditions has different shadows. This feature can be used to determine the underside of a vehicle [34]. Another feature used for separating vehicles from background is color [35,36]. RGB color system [36] and L^*A^*B color system [37] are more conventional. LED lighting is used for detecting and tracking vehicles at night [6].

3. The Proposed Method

Different image features which are of high importance for vehicle detection were briefly introduced and reviewed in several studies in the previous section. Indeed, multiple low-level features were used in this study. The researchers tried to group and cluster these features to identify vehicles. ROI (Region of Interest) feature is determined based on Gaussian probability distribution function. Fig.2 depicts the block diagram of the proposed method. The components of the proposed method are described consequently.

3.1 Feature Extraction

As discussed earlier, there exist many horizontal lines at the back of vehicles such as shadow lines, window lines, and top and bottom lines of a vehicle. This feature is relatively stable to the changes of light and scale. Indeed, it should be noted that, in some frames, vehicles might not be exactly in front of the camera i.e. they may be in margins; hence, horizontal or vertical lines are not clear. Thus, an algorithm used for edge detection methods such as canny [38] is regarded as more appropriate. In contrast with other methods which only used horizontal or vertical lines, a fully edge detection algorithm was utilized in this study. Fig.1.b illustrates the result of edge detection on the image.



Fig.1. Original image and extracted features (a) original image (b) image edge extracted by canny algorithm (c) corners extracted by harris method (d) adaptive boundingbox

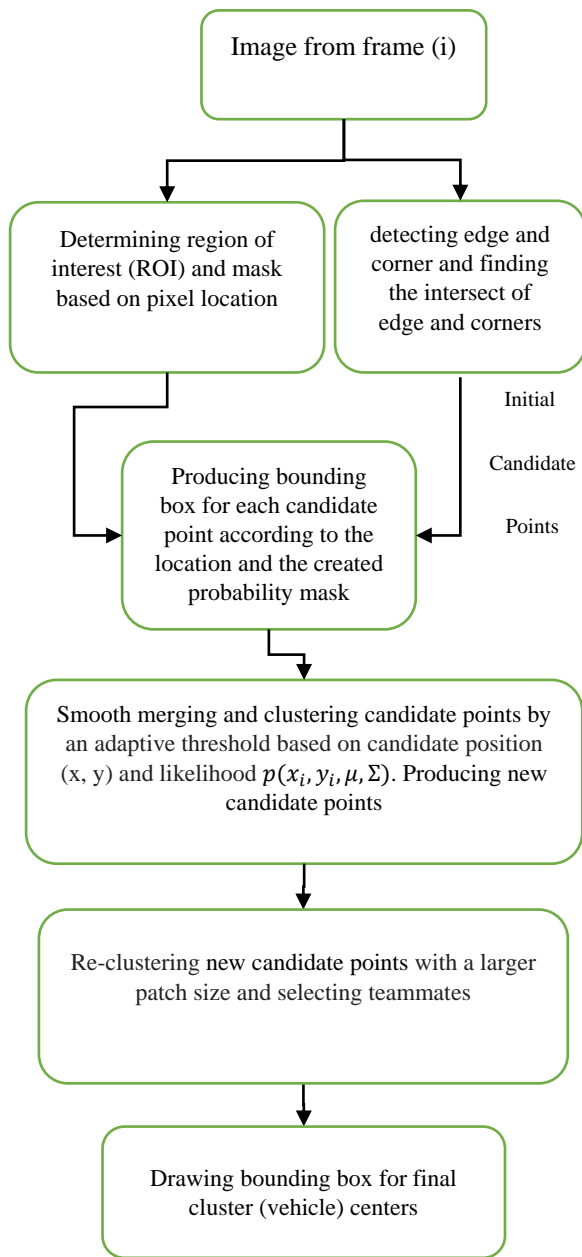


Fig. 2. Block diagram of the proposed method

Corner is considered to be an important feature of video images which is commonly used in vehicle detection systems. The most common corner extraction method is based on harris features [39]. In this study, image corners in combination with edges were used as low level features for detecting vehicle. Fig.1.c depicts corners on an image from LISA dataset.

3.2 Focus on Probability

As shown in Fig.1.a, each frame has numerous details around the road. Hence, determining ROI for further processing is essential for two aspects. Firstly, appropriate ROI placing has a direct impact on the accuracy of detection. Secondly, as a given area is determined more precisely, less time will be spent for processing in the next level. According to camera settings and its

circumstance and location on the ego_vehicle, ROI will vary in size and details.



Fig. 3. Angular installed camera effect on captured image, more angular (left side); less angular (right side)

In this paper, certain parameters such as ROI determination and marginal detail parameters including camera parameters were specified. As illustrated in Fig.3.a, in case the installed camera is angular, the captured image will include more details and the sky will be the greatest part of image. However, in case camera position and angle relation to the horizon is less, the image will include more vehicles but less marginal details. Fig.3.b illustrates this fact. λ and θ refer to ego-vehicle height and camera angle with respect to the horizon. The values of the parameter are converted to interval [0,1].

The probability that each image pixel is a vehicle pixel is computed according to its position $X(x, y)$ and the bivariate Gaussian distribution function mentioned in eq.2. For obtaining more adaptation in this distribution function, the parameters λ and θ are used to create the covariance matrix (eq.3).

$$p(X; \mu, \Sigma) = \frac{1}{(2\pi)^{|\Sigma|} \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(X - \mu)^T \Sigma^{-1} (X - \mu)\right) \quad (2)$$

$$\Sigma = \begin{bmatrix} C * \lambda & 1 \\ 1 & R * \theta \end{bmatrix} \quad (3)$$

In equation 3, C and R denote the size of image columns and rows respectively. Since the Gaussian distribution is symmetrical in relation to the mean of variables, the positive part was only used. Fig.3.c and Fig.3.d indicate the probability mask of each region as a gray level in the image with the specified parameters. Fig.3.e and Fig.3.f are produced by applying these masks on the original images. Inasmuch as bright parts of the masks are ROI, they must be processed further in the next levels.

3.3 Clustering of Initial Center Points

As discussed earlier, edge and corner features were used in the present study to detect vehicles. Consequently, candidates for further processing were selected from the intersection of dilated edges and corners according to equation 4.

$$C = \text{corners} \cap (\text{edge} \oplus \text{se}) \quad (4)$$

Indeed, edges and corners are features which were extracted in the previous stage. *se* stands for dilation mask which is like a circle with 5-pixel radius. Further processes were conducted on *C*.

$$C = \{c_i = (x_i, y_i) \mid i = 1, 2, \dots, K\}, \quad (5)$$

C includes coordinates of candidate points. It should be noted that the image sizes of vehicles near camera are large and those of distant vehicles are small. Hence, applying a uniform threshold for drawing bounding box and clustering centers are not possible. Consequently, the following adaptive threshold based on candidate position (*x*, *y*) and the likelihood $p(x_i, y_i, \mu, \Sigma)$ was used in the study.

$$AD_i = \text{patch}_{\text{size}} * p(x_i, y_i, \mu, \Sigma) \quad (6)$$

For drawing a bounding box around the initial central points and clustering them, Adaptive_{Distance} (AD) was used which is shown in eq.5 threshold. Thus, the center which is probably considered to be a vehicle has an even greater margin (Fig.1.c).

3.4 Clustering of Candidate Centers

For obtaining more accurate results, centers were clustered using the Euclidean distance measure and the AD threshold was computed for each center. It should be pointed out that the number of clusters is determined while clustering.

$$D(c_i, c_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (7)$$

$$TD(i, j) = \min(AD_i, AD_j) \quad (8)$$

$$G(c_i, c_j) = \begin{cases} 1 & \text{if } D(c_i, c_j) \leq TD(i, j) \\ 0 & \text{o. w} \end{cases} \quad (9)$$

In this grouping method, the number of clusters is variant which is determined while clustering according to the cluster merging. After clustering, cluster centers should be specified according to the following equation for subsequent processing.

$$\underbrace{CL}_k \quad (10) \\ = \{[\text{mean}(x_i) \quad \text{mean}(y_i)] \mid \forall i \ G(c_i, c_k) == 1\}$$

In this equation, *N* denotes the number of pre-generated clusters. After initial clustering, the final step to vehicle determination is carried out by re-clustering with a larger patch size and averaging based on the number of members. Neighbor points are selected by choosing teammates based on correlation and average computation.

In this stage, pairwise correlation among all central points is measured. Then, a multivariate data adjacency matrix is produced. Correlations between points are computed based on gray level differences and Euclidean distances between center coordinates as bellow:

$$CR(c_i, c_j) = \frac{1}{(|v_i - v_j|)^{\rho} + (\text{dist}(c_i, c_j))^{\gamma} + 1} \quad (11)$$

Where, v_i and c_i refer to the *i*'th center point gray level and coordinate respectively.

CR has a value within the interval [0, 1]. The self-adjacency value for each center point is 1. γ and ρ parameters determine the effectiveness of the Euclidean distance and gray level differences respectively. By increasing the Euclidean distance between centers and the difference between gray levels, CR value tends towards zero. After constructing multivariable adjacency matrix, the recruitment is conducted for grouping the same samples. The following question arises in this stage: how many centers with how much adjacency can form a cluster? For answer this question, an instance candidate center is selected and it is assumed to be a cluster. CR values of this center and those of other centers are computed. The closest center is added to the cluster. The mean of CR values in the cluster is computed and the average is weighted based on the number of samples it takes. The weight grows while the number of samples increases. All the processes including selecting, averaging and weighting is referred to as teammate selection which is depicted in fig.4.

In fact, sampling and teammate selection and weight increase continue as far as monotony property of averaging is not violated in recruiting. By imposing this constraint, nearby points form a cluster. Even though distant point selection increases averaging weight, it is not enough to enhance the total amount of weighted average with respect to the previous value. Hence, our objective was to recruit more so that the adjacency value of members would not be less than that of a threshold. As mentioned earlier, after applying weighted averaging and recruitment, an adaptive clustering step based on different threshold was used to obtain the final results.

```

Compute CR matrix based on eq.10;
foreach ( unselected candidate center point i in CR matrix)
  Num=1;
  Len=number of disjoint centers;
  K=1;
  raw_mean=0;
  While (K<=Len)
    Max_v=Choose maximum value CR in i'th row;
    Update raw_mean according Num and Max_v value;
    New_mean=weighting raw_mean according to Num value;
    If ( new_mean >=old_mean)
      Add Max_v to i'th cluster and remove from CR matrix;
      Num++;
    else
      Break;
    End {if}
    K++;
  End {while}
End {foreach}

```

Fig. 4. The proposed teammate selection clustering algorithm

4. Results

4.1 Dataset Description

The proposed method was evaluated by means of LISA dataset available online at <http://cvr.ucsd.edu/LISA/index.html>. The evaluation data set included three different sets with different traffic congestions and different levels of identification complexity. The first dataset included 300 frames captured on March with an SUV vehicle in a curved road. Also, pedestrians moved along and across the street. The second data set also included 300 frames which were taken in different lighting conditions on April. This data set consisted of multiple vehicles which were moving regularly. The third data set including 1600 frames was taken on January. This data set was complicated because it was taken at the most crowded time and included shadows, high-motion maneuvering and five movement lines. All these data sets were hand-labeled and a box was drawn around each vehicle.



Fig. 5. True labeled vehicles shown by green boxes and blue boxes in the proposed method

There are some well-known performance metrics such as precision, recall [40], average false positive per frames, average false positive per object and average true positive per frame [8] which were used in the study as the evaluation criteria.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{Average FP/frames} = \frac{FP}{\text{total number of frames processed}} \tag{14}$$

$$\text{Average FP/objects} = \frac{FP}{\text{true vehicles}} \tag{15}$$

$$\text{Average TP/frames} = \frac{TP}{\text{total number of frames processed}} \tag{16}$$

Table 1. Experimental results

Criteria	Description	#1	#2	#3
ATPpF	Average TP per Frames	1	2.966	2.67
AFPpF	Average FP per Frames	2.133	1	2.22
ATPR	Average True Positive Rate (recall)	1	0.989	0.621
AFDR	False Detection Rate	0.519	0.223	0.399
AFNpF	Average FN per Frames	0	0.033	0.193
AVpF	Average Vehicle number per Frame	1	3	4.38
AFPpObj	Average FP per Objects	2.133	0.333	0.507
ATPpObj	Average TP per Objects	1	0.989	0.621
AFNpObj	Average FN per Objects	0	0.011	0.044
Precession		0.48	0.768	0.60

Table 2. Comparison of the proposed method with Elvis [41] and active learning method [8].

	Best value →	Depends on #vehicles	Small	small	large
tracking system	Criteria dataset	TP frame	FP frame	FDR	TPR
		The Proposed method	#1: 1 #2: 2.966 #3: 2.67	2.133 1 2.22	0.519 0.223 0.399
Elvis[41]	#1: 1 #2: 2.92 #3:	1.13 1.06	0.531 0.267	1 0.975	
Active learning method[8]	#1: 1 #2: 3.16 #3:	4 2.7	0.797 0.458	0.835 0.981	

The obtained results are given in table (1) and the results of comparing the proposed method with two other methods are given in Table (2). As shown in Table (2), the proposed method has good results with respect to all of the computed measures. The problem was that only the rears of vehicles were tagged. In other words, in case a vehicle appears with a part other than the rear part, then, the detection will fail and the algorithms detecting these vehicles will be registered as an error in terms of system evaluation. Since a correctly detected vehicle (TP) will be considered as the system error (FP), hence, the system performance will decreased. Fig.4 shows a frame of each data set in which the proposed system detected a vehicle or a pedestrian which are not labeled in the data set. As a case in point, in the first dataset, a pedestrian crossing the street was detected but it was not labeled in the data set.

The algorithm used in the present study had some parameters which depended on the capturing condition and image size. For example, in dataset#2, $\theta = 0.9$ indicates that the camera is installed angularly and the sky is included in the majority of the image. The patch size and λ and θ were selected according to the image size and the

conditions. Nevertheless, γ and ρ were selected expertly and based on trial and error. Due to the incomplete labeling of the datasets, Average FP per objects (labeled) was too high, particularly in the first dataset.

Table 3. The parameters used in the proposed method

Parameter	Brief description	#1	#2	#3
λ (normalized)	The ego-vehicle's height	0.4	0.1	0.6
θ (normalized)	The camera's angle relative to the horizon	0.7	0.9	0.4
γ	determine effectiveness of the Euclidean distance	0.7	0.5	0.5
ρ	determine effectiveness of the gray level differences	0.5	0.5	0.8
Patch size #1	Initial patch size for bounding box determination in first phase	100	100	100
Patch size #2	Initial patch size for bounding box determination in second phase	300	250	200
τ	Threshold for neighboring	10	10	10

5. Conclusion and Future Works

The study reported in this paper focused on the significant challenge and issue of detecting vehicles with

a camera embedded inside a car. The underlying approach was based on a new clustering method which is referred to as teammate selection. In this clustering method, a new merging measure based on the distances of cluster centers and gray scale values was used. Several general features such as "edges" and "corners" were taken into consideration to robustly characterize vehicles. The detection was carried out hierarchically at several stages. After extracting feature, masks were used to determine ROI based on the probability of the position of each pixel. Then, feature clustering was performed according to the listed parameters. Next, a weighted average based sampling and teammate selection was measured and the final clustering was implemented by the new parameters.

It can be argued that the *significant* contribution of this study was to determine vehicles as clusters without having any information about the number of them in each frame. The utilized algorithm had some parameters which were functions of the capturing condition and image size. The results of the study indicated that the high TPR (recall) is a notable benefit of the proposed method. Due to the complexity and difficulty of dataset#3, the obtained results were not satisfying. Indeed, it can be maintained that parameters for complex environments can be improved by optimizing algorithms such as genetic algorithms.

References

- [1] S. Nadimi and B. Bhanu, "Multistrategy fusion using mixture model for moving object detection," in *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, 2001, pp. 317-322.
- [2] J. Sochman and J. Matas, "Waldboost-learning for time constrained sequential detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 150-156.
- [3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I-511-I-518.
- [4] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012, pp. 975-982.
- [5] A. Takeuchi, S. Mita, and D. McAllester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 1014-1021.
- [6] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, 2011, pp. 583-595.
- [7] S. Samadi and F. M. Kazemi, "A Multi-Agent Vision-Based System for Vehicle Detection," *World Applied Sciences Journal*, vol. 15, 2011, pp. 1722-1732.
- [8] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, 2010, pp. 267-276.
- [9] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study," *Machine Vision and Applications*, 2011, pp. 1-13.
- [10] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, 1994, pp. 201-221.
- [11] C. H. Lampert and J. Peters, "Active structured learning for high-speed object detection," in *Pattern Recognition*, ed: Springer, 2009, pp. 221-231.
- [12] S. Sivaraman and M. M. Trivedi, "Real-time vehicle detection using parts at intersections," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012, pp. 1519-1524.
- [13] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *Image Processing, IEEE Transactions on*, vol. 15, 2006, pp. 2019-2034.
- [14] H. Cheng, N. Zheng, and C. Sun, "Boosted Gabor features applied to vehicle detection," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, pp. 662-666.
- [15] S. Kim, S.-Y. Oh, J. Kang, Y. Ryu, K. Kim, S.-C. Park, et al., "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*, 2005, pp. 2173-2178.
- [16] R. K. Nath and S. K. Deb, "On road vehicle/object detection and tracking using template," *Indian Journal of Computer Science and Engineering*, vol. 1, 2010, pp. 98-107.
- [17] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," in *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, pp. 310-314.
- [18] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, 2006, pp. 694-711.
- [19] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Engineering Practice*, vol. 4, 1996, pp. 473-479.
- [20] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol. 12, 2000, pp. 69-83.
- [21] N. Srinivasa, "Vision-based vehicle detection and tracking method for forward collision warning in automobiles," in *Intelligent Vehicle Symposium, 2002. IEEE*, 2002, pp. 626-631.
- [22] M. Bertozzi, A. Broggi, and S. Castelluccio, "A real-time oriented system for vehicle detection," *Journal of Systems Architecture*, vol. 43, 1997, pp. 317-325.
- [23] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, 2006, p. 13.
- [24] T. Ten Kate, M. Van Leewen, S. Moro-Ellenberger, B. Driessen, A. Versluis, and F. Groen, "Mid-range and distant vehicle detection with a mobile camera," in *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, pp. 72-77.
- [25] T. Kalinke, C. Tzomakas, and W. v. Seelen, "A Texture-based Object Detection and an adaptive Model-based Classification," 1998.
- [26] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, 2001, pp. 3-55.
- [27] A. Kuehne, "Symmetry-based recognition of vehicle rears," *Pattern recognition letters*, vol. 12, 1991, pp. 249-258.
- [28] T. Zielke, M. Brauckmann, and W. Vonseelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, 1993, pp. 177-190.
- [29] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, 2012., pp. 831-842
- [30] J. Diaz Alonso, E. Ros Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *Vehicular Technology, IEEE Transactions on*, vol. 57, 2008, pp. 2736-2746.
- [31] M. Perrollaz, J.-D. Yoder, A. Nègre, A. Spalanzani, and C. Laugier, "A visibility-based approach for occupancy grid computation in disparity space," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, 2012, pp. 1383-1393.
- [32] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674-679.
- [33] J. Choi, "Realtime On-Road Vehicle Detection with Optical Flows and Haar-Like Feature Detectors," *Urbana*, vol. 51, p. 61801, 2007.
- [34] H. Mori and N. M. Charkari, "Shadow and rhythm as sign patterns of obstacle detection," in *Industrial Electronics, 1993. Conference Proceedings, ISIE'93-Budapest., IEEE International Symposium on*, 1993, pp. 271-277.
- [35] J. D. Crisman and C. E. Thorpe, "Color vision for road following," in *1988 Robotics Conferences*, 1989, pp. 175-185.
- [36] S. D. Buluswar and B. A. Draper, "Color machine vision for autonomous vehicles," *Engineering Applications of Artificial Intelligence*, vol. 11, 1998, pp. 245-256.
- [37] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 249-254.
- [38] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1986, pp. 679-698.
- [39] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 1988, p. 50.
- [40] D. L. Olson and D. Delen, *Advanced data mining techniques [electronic resource]*: Springer, 2008.
- [41] RK. Satozoda and MM. Trivedi. "Efficient lane and vehicle detection with integrated synergies (ELVIS)." In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014 IEEE Conference on, 2014, pp. 708-713.

Moslem Mohammadi Jenghara is Instructor of Computer Engineering at Payame Noor University (PNU), Miandoab, Iran. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering at the University of Kashan, Kashan, Iran. He had received his M.Sc. degrees in Computer Engineering (Artificial intelligence) in 2008, from Iran University of Science and Technology, Tehran, Iran. He has published several articles and presented several papers in international and national journals and conference about data mining, text mining and image processing. His main area of research includes Data mining, Pattern Recognition, text mining, temporal graph mining and applications of Artificial Intelligence in Bioinformatics.

HosseinEbrahimpour-Komleh is currently an Assistant Professor at the Department of Electrical and Computer Engineering at the University of Kashan, Kashan, Iran. His main area of research includes Computer vision, Image Processing, Pattern Recognition, Biometrics, Robotics, Fractals, chaos theory and applications of Artificial Intelligence in Engineering. He received his Ph.D. degree in Computer Engineering from Queensland University of Technology, Brisbane, Australia in 2006. His Ph.D. research work was on the "Fractal Techniques for face recognition". From 2005 to 2007 and prior to joining the University of Kashan, he was working as a Post-doc researcher in the University of Newcastle, NSW, Australia and as a visiting scientist in CSRIO Sydney. Hossein Ebrahimpour- Komleh has B.Sc. and M.Sc. degrees both in Computer Engineering from Isfahan University of Technology (Isfahan, Iran) and Amirkabir University of Technology (Tehran, Iran,) respectively. He has served as the invited keynote speaker, editorial board member and reviewer of several journals and international and national conferences