Efficient Lane and Vehicle detection with Integrated Synergies (ELVIS)

Ravi Kumar Satzoda and Mohan M. Trivedi
Laboratory for Intelligent and Safe Automobiles, University of California San Diego
rsatzoda@eng.ucsd.edu, mtrivedi@ucsd.edu

***Abstract*—On-road vehicle detection and lane detection are integral parts of most advanced driver assistance systems (ADAS). In this paper, we introduce an integrated approach called Efficient Lane and Vehicle detection with Integrated Synergies (ELVIS), that exploits the inherent synergies between lane and on-road vehicle detection to improve the overall computational efficiency without compromising on the robustness of both the tasks. Detailed evaluations show that the vehicle detection component of ELVIS shows at least 50% lesser false alarms with equal or better detection rates, and reducing the computational costs by over 90% as compared to state-of-the-art vehicle detection methods. Similarly, the lane detection component shows more reliable lane feature extraction with average computation costs that are at least 35% lesser than existing techniques.**

***Keywords*-vehicle detection, lane detection, integrated system, computational efficiency**

I. Introduction

In recent years, there has been an significant increase in the number of the embedded electronic processors in automobiles from 1% in 1980 to over 22% in 2007 [3]. The presence of such systems directly impacts the overall power consumption, which is especially critical in present times when the automotive industry is moving towards battery-powered hybrid and electric vehicles.

Among the different embedded sub-systems, vision-based advanced driver assistance systems (ADAS) are becoming more popular in recent times because of the availability of low-cost, high-resolution and pervasive cameras [15]. In order to meet the increasing demand in vision-based ADAS, manufacturers such as Texas Instruments etc. are releasing newer embedded platforms that are specifically catered for implementing vision algorithms in automobiles [7]. Although such platforms do provide an advantage over conventional processors and hardware systems, the efficiency (both power and real-time operation) are constrained by the implementation of the constituent vision algorithms [14].

Various computer vision algorithms have been explored for ADAS tasks such as detection of lanes, vehicles, pedestrians etc. in [11], [4] etc., wherein a variety of feature extraction methods, classification techniques, and tracking methods have been proposed for improving accuracy. While robustness is critical in such vision-based active safety systems, exporting such data-intensive algorithms on embedded platforms is still a challenging task [14], [10]. Although implementations such as lane detection in [6] demonstrate the realization of ADAS algorithms on embedded platforms, designing the algorithm itself to cater for more efficient embedded realizations is less explored in current literature [6], [10]. Similarly, vehicle detection is also mainly explored for improving accuracy, which primarily focus on multi-scale sliding window approach [11], [5]. There are a number of such techniques in literature [13]; however, integrating the two tasks, i.e. lane detection and vehicle detection, is less explored [12].

Figure 1. (a) Sliding window of multiple sizes used across the image, (b) Relationship between sliding window and the position of the leading vehicles.

In this paper, we propose an integrated approach called ELVIS (**E**fficient **L**ane and **V**ehicle detection using **I**ntegrated **S**ynergies) that incorporates the lane information to detect vehicles more efficiently in an informed manner using a novel two-part based vehicle detection technique. The vehicle detection outputs are further used in ELVIS to improve the efficiency of the lane feature extraction process. We will present detailed evaluations that show significant gains in computational efficiency using ELVIS, without compromising on accuracy of both lane and vehicle detection.

II. ELVIS: Proposed Integrated Approach

In this section, we will first propose the vehicle detection component of ELVIS followed by the lane feature extraction method that benefits from vehicle detection.

A. Vehicle Detection using Lane Information

Let us consider a typical road scene as shown in Fig. 1. Conventional classifier-based vehicle detection techniques apply a sliding window of multiple scales (or the image is scaled keeping the size of the window same) across the input frame as shown in Fig. 1(a) to detect vehicles

2160-7516/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPRW.2014.108

of multiple sizes. Although such techniques exhaustively look for vehicles, they also result in false positives and can be computationally overwhelming for embedded realization with a few hundreds of thousands of windows for processing. There is an additional computational overhead for eliminating false positives using tracking, the cost of which increases with higher number of false positive windows.

In ELVIS, we consider the following observations about on-road vehicle detection. The on-road vehicles appear in an orderly manner along the road surface, with sizes that are changing in an orderly way, especially in highway and urban driving. Therefore, as shown in Fig. 1(b), if the image is scanned along the road surface, the vehicles appearing first in the direction of the road will appear bigger than the vehicles that appear later.

In the proposed method, we use the lane information and the contextual information described above to detect the vehicles in an informed manner. The first part of the proposed method is a one-time process to generate an LUT (look-up table) of the positions and sizes of the windows which should be used to detect the vehicles. In order to do this, we use the inverse perspective mapping (IPM) of the image, which is can be readily derived based on the camera calibration [1]. Considering that IPM generation is an integral part of most lane detection methods such as [8], [2], [10], [1], it will reused for lane detection in ELVIS later.

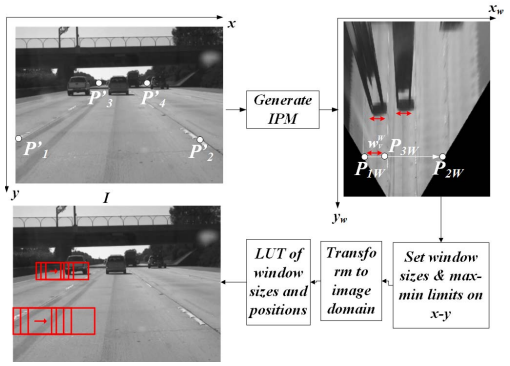

Figure 2. Generating the LUT of windows and positions using lane and IPM information.

Fig. 2 illustrates the LUT generation process. Given an input image I, the IPM image I_W is generated using the homography matrix $\mathbf{H}$ [1]. Therefore, every point $P(x, y)$ in I is transformed to P_w in I_W using $\mathbf{H}$, i.e.,

$$\begin{bmatrix} x_w & y_w & 1 \end{bmatrix}^T = k\mathbf{H} \begin{bmatrix} x & y & 1 \end{bmatrix}^T \quad (1)$$

where k is the calibration constant. Therefore, the four points P'_1 to P'_4 in Fig. 2 in the image domain correspond to the minimas and maximas in the IPM domain along the $x_w - y_w$ coordinates. The following parameters for each row I_W are determined: (1) the start position of road surface in the row, i.e. P_{1W} (Fig. 2), (2) the end position of the road surface P_{2W}, and (3) point P_{3W} such that $P_{3W} - P_{1W} = w_V^W$, where w_V^W is the width of the vehicle as seen from top view. Considering most consumer vehicles usually have a standard axle length, w_V^W can be pre-determined. Given P_{1W}, P_{2W} and P_{3W}, we use the inverse of $\mathbf{H}$, i.e. $\mathbf{H}^{-1}$ to determine the corresponding points P_1, P_2 and P_3 in the image domain. For each row index y in I, we now have the following:

$$x_{min} = x_1 \, , x_{max} = x_2 \, , w_V = x_3 - x_1 \quad (2)$$

where w_V is the width of the window that should be used for vehicle detection in the y-th row of I, and x_{min} and x_{max} are the minimum and maximum indices along the x-axis in I where the windows will be processed. The height h_V of the window was set to $1/1.2$ times w_V, which was found suitable to detect most consumer vehicles. An LUT is generated with the above variables for each row of the input image I. Therefore, unlike the conventional way of running the sliding window all over the image, we now have a defined search space and specific scales of the window for every row in the LUT to detect the vehicles.

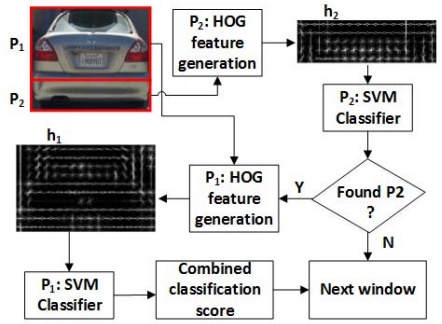

Figure 3. Two-part based vehicle detection method in ELVIS.

In addition the LUT generation, we will now propose a two-part based vehicle detection scheme in ELVIS that incorporates the next contextual information about vehicles moving in the road scene. In an ADAS application that requires on-road vehicle detection such as collision avoidance or lane change assistance, it is evident that the vehicles are moving on the road surface. Therefore, the windows are applied in the direction of the decreasing y-coordinate (referring to the axes in Fig. 2), i.e. from the front of the host-vehicle towards the vanishing line. Next, the window I_V is divided into two parts $\mathbf{P_1}$ and $\mathbf{P_2}$ such that if the window is placed on the vehicle, they capture the two parts of the vehicle as shown in Fig. 3. For an $M \times N$ (rows by columns) sized image patch I_V, $\mathbf{P_1}$ and $\mathbf{P_2}$ are defined as follows:

$$\mathbf{P_1} = \mathbf{I_V}\left(1 : \frac{2}{3}M, 1 : N\right) \quad \mathbf{P_2} = \mathbf{I_V}\left(\frac{2}{3}M : M, 1 : N\right) \quad (3)$$

In ELVIS, we first detect the presence of lower part of the vehicle, i.e. $\mathbf{P_2}$ because while traversing along the road, the

lower part is expected to be seen first. The HOG feature [4] of part $\mathbf{P_2}$ in I_V denoted as h_2 is computed first, and classified using SVM classifier for $\mathbf{P_2}$. If the classification of the lower part is positive, then HOG feature h_1 for $\mathbf{P_1}$ is computed and classified using the SVM classifier for $\mathbf{P_1}$. The classifier scores from the SVMs of parts $\mathbf{P_1}$ and $\mathbf{P_2}$ are used in the following way to give the final classification result:

$$\mathbf{I_V}(x, y) = vehicle \text{ if } p_1(x, y) \times p_2(x, y) > T_p \tag{4}$$

where p_1 and p_2 are the probability scores of the SVMs for parts $\mathbf{P_1}$ and $\mathbf{P_2}$ respectively, and T_p is the threshold for overall classification score. Non-maximal suppression is applied to remove overlapping windows.

B. Improving Lane Detection using On-road Vehicle Information

In this section, the performance of lane detection is improved using the results obtained from the vehicle detection algorithm. For most ADAS applications such as lane departure warning, lane change assistance, collision avoidance etc., lanes are required to either localize the ego-vehicle in the lane or to maneuver the ego-vehicle between the host lane and the adjacent lanes. Therefore, if there is a leading vehicle in front of the ego-vehicle, detecting the lanes between the two vehicles is sufficient. Also, presence of a leading vehicle obstructs the view in front of the ego-vehicle and lanes are not visible in the image plane resulting in false lane features that affect the accuracy of lane estimation [12]. We propose to use the positions of the vehicles that were determined by the proposed vehicle detection scheme in ELVIS to determine the regions in the input image where the lanes must be detected.

We demonstrate this using one of the recently proposed lane detection methods [10], [9] called LASeR (lane analysis using selective regions). In this paper, we limit the scope of the proposed lane detection approach to the host lane detection for the sake of explanation. Also, LASeR algorithm in [10] is designed for host lane detection, which will be used to demonstrate the proposed integrated method. In LASeR, N_B number of scan bands at positions $Y = \{y_j^B\}$ along the y_w axis of the IPM image are processed where $1 \leq j \leq N_B$. The bands are shown in Fig. 4 where the band closest to the ego-vehicle is indexed as 1. Given that we have the positions of the vehicles at $P^V(x^V, y^V)$ in the image domain I from ELVIS, we find their positions in the IPM domain using (1) to get $P_W^V(x_W^V, y_W^V)$, i.e. $P_W^V = k\mathbf{H}P^V$. Therefore, with the leading vehicle at y_W^V, the index of the maximum band along y_W axis in the IPM image that must be processed in LASeR is computed using the following equation:

$$N^{max} = \arg\max_j y_W^V \leq y_j^B \tag{5}$$

Referring to Fig. 4, the conventional LASeR algorithm would have processed all the 5 scan bands shown in the IPM image in Fig. 4. Due to the presence of vehicles in bands B_4 and B_5, false lane features could be generated. However, the integration of the vehicle detection method into the lane detection process allows choosing $N^{max} = 3$ bands, that are processed by the next steps of LASeR algorithm. We will demonstrate the advantages of the proposed integration between lane and vehicle detection in terms of computational complexity and accuracy in the next section.

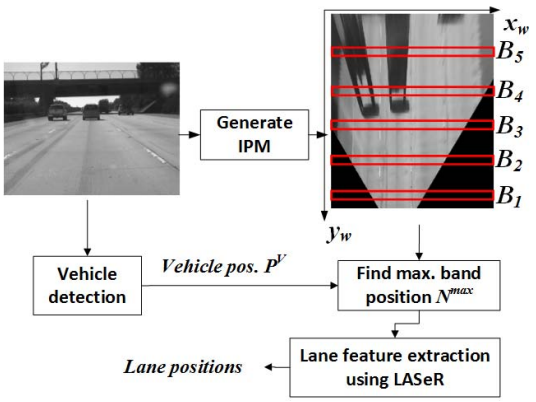

Figure 4. Improving lane feature extraction using using vehicle positions.

III. Performance Evaluation

A. Vehicle Detection in ELVIS

1) Accuracy Analysis: The accuracy analysis setup involved a training set comprising 1700 positive samples and 2500 negative samples. Each sub-image was divided into two parts as shown in Fig. 3 and two different classifiers were trained using the two sets of training samples. The testing of the proposed vehicle detection component in ELVIS was performed using three different datasets that do not contain any of the training samples - Caltech 1999 (126 images), LISA dataset 2 (300 images) and LISA dataset 3 (300 images) [11]. We use the following metrics described in [11] for evaluating and comparing accuracy with results in [11]:

$$TPR = \frac{\text{True detections}}{\text{Total num. of veh.}} \; ; \; FDR = \frac{\text{False positives}}{\text{True det. + False pos.}} \tag{6}$$

$$\frac{FP}{frame} = \frac{\text{False Positives}}{\text{Total num. of frames}} \; ; \; \frac{TP}{frame} = \frac{\text{True positives}}{\text{Total num. of images}} \tag{7}$$

where TPR is true positive rate and FDR is false detection rate. Fig. 5 plots the receiver operating curves (ROCs) showing the TPR versus FDR on applying ELVIS to the three datasets. It can be seen that the proposed two-part based method performs well with the Caltech dataset giving over 90% TPR for less than 5% of false detection rate. Coming to the LISA 2 dataset, the proposed method detected the single vehicle in all cases giving 100% true positive rates for all data points considered. In LISA 3 dataset ELVIS gives over 95% TPR for less than 10% FDR.

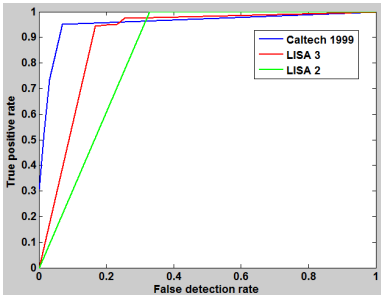

Figure 5. ROCs for detecting fully visible vehicles using the proposed two-part based method.

Table I
COMPARISON BETWEEN ELVIS AND ALVERT IN [11]

	TPR	FDR	TP/frame	FP/frame
Caltech '99 [11]	85%	5%	NA	NA
LISA 2 [11]	83.5%	79.7%	1	4
LISA 3 [11]	98.1%	45.8%	3.16	2.7
ELVIS - Caltech '99	95%	7%	0.95	0.07
ELVIS - LISA 2	100%	53.1%	1	1.13
ELVIS - LISA 3	97.5%	26.7%	2.92	1.06

Table I compares the measures for the four different metrics and three different datasets between the two-part based vehicle detection technique in ELVIS and the passive learning Adaboost based classifier in [11]. It can be seen that in most cases the proposed method gives atleast 1-2% better TPRs with more than 15% lower FDRs as compared to [11]. It should also be noted that the figures in Table I for [11] involves tracking for LISA 2 and 3 datasets, whereas we have reported detection rates for ELVIS without tracking. We can see that FPs/frame have reduced by 2-3 times in ELVIS, which implies that tracking will be more effective in reducing the false positives further. Fig. 6 shows the detection of fully visible vehicles on sample images from different datasets that were evaluated in Table I.

Figure 6. Sample results of vehicle detection in ELVIS.

2) Computational Efficiency: In order to compare the computational efficiency of the proposed vehicle detection method in ELVIS against variants of existing sliding window based techniques, the following assumptions are made. Firstly, the evaluation is performed for feature extraction and classification steps only, and the cost of peripheral steps such as non-maximal suppression and tracking is assumed the same for all methods. Next, all techniques are assumed to use the same kinds of feature vectors and classifiers. Therefore, the computational cost is evaluated based on the number of windows that are processed in each method, which is a more generic and platform-independent metric. Computation cost can also be directly translated to computation time if required.

Let us first look at the computation cost of existing recent techniques such as [11], [5]. In such methods, a sliding window of multiple scales is traversed all over the image to detect vehicles in every frame. Therefore, considering an $m \times n$ (rows by columns) sized image frame I, if the window slides over every s_m and s_n pixels along the rows and columns respectively, the total number of windows that will be processed in existing methods is equal to:

$$C_1 = n_s \frac{m \times n}{s_m \times s_n} \tag{8}$$

where n_s is the number of scales that are used for detection.

In the proposed method, the scale of the window for each row in the image frame, and the start and end positions to slide the window in each row are determined by the formulations shown in equations (1) and (2) in Section II. Therefore, the total number of windows which is equivalent to the computational cost C_2 of the proposed method is equal to:

$$C_2 = \sum_{j=0}^{\frac{y^{max}}{s_m}-1} \frac{x^{max}_{s_m j+1} - x^{min}_{s_m j+1}}{s_n} \tag{9}$$

where x^{max} and x^{min} are the maximum and minimum x-coordinates in the image plane for the s_{mj+1}-th row. The above equation is the summation of the number of windows for each row in the image plane and within y^{max}-th and y^{max}-th rows of the image where the step size of window traversal is s_m and s_n pixels along the rows and columns respectively. It is to be noted that the number of scales n_s is not a factor in C_2. Additionally, the above equation gives the worst case scenario for the proposed method because (9) assumes that the entire window is processed throughout the image. However, as discussed in Section II, we process $1/3$-rd of the window first, and the next $2/3$-rd of the window is processed only if the lower part is detected.

Fig. 7 shows the comparison of total computation cost in terms of the number of windows for four different types of methods. In order to compute the results in Fig. 7, $m = 720$, $n = 1280$, $n_s = 10$, $s_m = 1$ and $s_n = 3$ are considered. The first two methods (methods 1 and 2 on the x-axis of Fig. 7 refer to the conventional sliding

window based methods such as [11]. Method 2 refers to approaches where no knowledge of the region of interest is considered and the windows are applied on the entire image. In method 1, we assume that the vanishing line is available. We consider $y_{max} = 682$ and $y_{min} = 382$ for the results shown in Fig. 7, which are derived from some of the test cases we considered during evaluation. Method 3 and 4 refer to the proposed method. As discussed previously, we are assuming the worst case in Method 3 by considering both parts $\mathbf{P_1}$ and $\mathbf{P_2}$ of all windows are computed. In Method 4, we consider 10% of the windows only for processing both parts, whereas the remaining 90% are processed for the lower part only (10% is significantly higher than the actual percentage we observed during our evaluation). Also, for Methods 3 and 4, we consider the same region of interest as in Method 1 in terms of y-axis. The x-axis limits are determined by the proposed LUT generation step. Fig. 7 shows that the proposed method involves only 1/10-th the number of computations as compared to 1 and 2. When the part-based computations are also accounted in Method 4, we see that the computations are further reduced. With such significant savings in computations, ELVIS is more suitable for embedded realization.

B. Lane Detection in ELVIS

In this section, we will evaluate the performance of lane feature extraction step in ELVIS. Fig. 8 shows sample results of a sequence on which the conventional LASeR algorithm and ELVIS are applied. The first column of Fig. 8 shows that the LASeR algorithm tries to detect lane features beyond the obstructing lead vehicle in the ego-lane. In such scenarios, the leading vehicle tends to introduce features that are often mistaken for lane features. For example, lanes are estimated as curving at the far end of the field of view. Therefore, in the presence of an obstructing vehicle, the lane features that are detected by LASeR are not *credible* enough to extend the lanes beyond the leading vehicle [12].

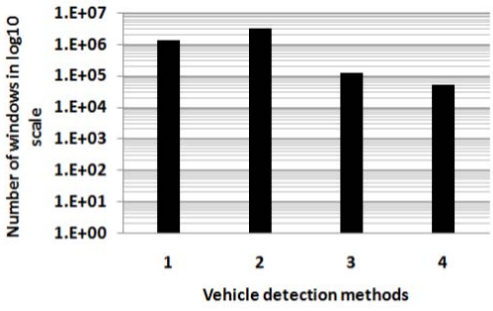

Figure 7. Comparison of computation costs in terms of number of windows (in log scale) on y-axis and the different techniques on x-axis: sliding window approach [11] bound by y_{max} and y_{min} in (1), on the entire image [11] (2), proposed method with both parts of all windows computed (3), and computed for 10% of the total windows (4).

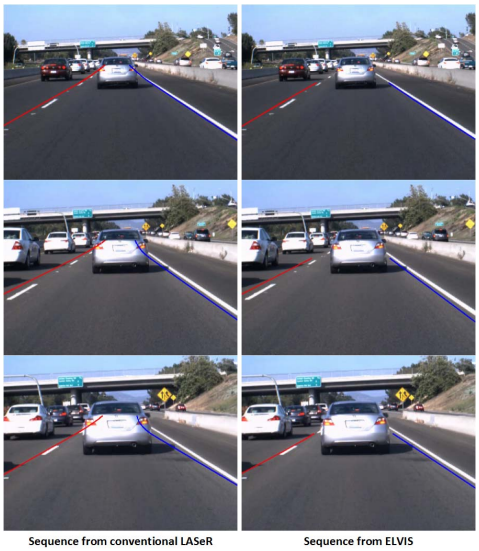

Figure 8. First column: lane detection using the conventional LASeR algorithm. Second column: lane detection using the proposed integrated approach.

The second column in Fig. 8 shows the lane detection results from ELVIS, wherein the lanes are detected till the position of the leading vehicle. It can be seen that the features of the vehicles do not interfere in the process of lane detection. It can be argued that the lanes are not being detected in the full field of view. It should be noted that in most ADAS applications, detecting the lanes between the ego-vehicle and the leading vehicle (if present) is more critical and necessary. If there is a requirement to detect the lanes beyond the obstructing vehicle, it is more accurate to extrapolate the lanes from existing lanes rather than depending on the lane features that are not visible due to obstructing vehicles. We compare the accuracy of lane estimation using lane position deviation metric (LPD), which computes the average error between the positions of the lanes in ground truth and those estimated by a lane detection algorithm [9] for every frame. Table II shows the mean, standard deviation and maximum absolute LPDs for the right lane from a set of 200 frames of the sequence shown in Fig. 8. In the case of ground truth marking and ELVIS algorithm, if the lane markings are not visible due to the leading vehicle, the lanes are extrapolated as a tangent till the maximum y-coordinate that is used by conventional LASeR. It can be seen that the lanes estimated by ELVIS are more aligned with the ground truth. In a similar manner, the vehicles detected in the image scene can also be used to eliminate false lane features that are detected due to vehicles in neighboring lanes.

Let us now consider computational efficiency. If we consider ADAS applications that involve both lane and vehicle

Table II
PERFORMANCE EVALUATION OF LANE DETECTION

Comparison of Lane Position Deviation (LPD)			
Method	**Mean Absolute LPD**	**Stan. Dev. of Absolute LPD**	**Maximum Absolute LPD**
ELVIS	8.3 pixels	3.3 pixels	18.7 pixels
LASeR [10]	9.3 pixels	3.3 pixels	19.1 pixels

Comparison of Computational Complexity			
VioLET	LASeR 2013	This work 2014	
2006 [8]	[10]	Average	Worst-case
300000	50000	32500	50000

detection tasks, it can be seen that the proposed ELVIS will be computationally more efficient than LASeR. Considering that the number of bands changes dynamically based on the presence or absence of the leading vehicles in ELVIS, the computational cost can be compared based on a given test sequence. Using the formulation of computation cost for LASeR from [10], the cost in LASeR is equivalent to $C_{L1} = N_B w_B h_B$ where each band is w_B pixels wide and h_B pixels high. In other IPM based lane detection algorithms such as [8], the cost is given by $C_{L2} = w_B y^W_{max}$, where y^W_{max} is the height of the IPM image [10]. However, in ELVIS, the average computational cost per frame for detecting lanes is given by:

$$C_{L3} = \frac{1}{N_f} \sum_{j=1}^{N_f} N_j^{max} w_B h_B \tag{10}$$

For the test sequence that is evaluated in this paper, Table II lists the average computational cost per frame. It is assumed that the lane detection is being performed for applications that employ vehicle detection also, therefore the cost of vehicle detection is assumed to be the same for all methods. It can be seen that for the sequence considered, the ELVIS gives an average computation cost savings of nearly 35% as compared to LASeR and reduces the cost of the conventional lane detection method such as VioLET by atleast 90%. The worst case computation cost of the proposed approach is maximum when there are no leading vehicles and all bands in LASeR are processed. Therefore, the worst case average computation cost of the integrated approach will never be more than conventional LASeR.

IV. CONCLUDING REMARKS

In this paper, a novel integrated approach called ELVIS is presented to detect on-road vehicles and lanes. We have shown that using the information about the lanes and the vehicle detection in a synergistic manner can help in detecting both vehicles and lanes in a computationally more efficient and robust manner. The proposed techniques have been shown to detect fully visible vehicles with lower false positive rates and nearly 90% lesser computations as compared to existing methods. Similarly, the lane feature extraction method is also shown to benefit with significant cost savings by integrating the vehicle detection information. The proposed techniques need to be further extended for occluded vehicles, which is the next step in the proposed research.

REFERENCES

[1] M. Bertozzi and A. Broggi. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. on Image Proc.*, 7(1):62–81, Jan. 1998.

[2] A. Borkar, M. Hayes, and M. T. Smith. A Novel Lane Detection System With Efficient Ground Truth Generation. *IEEE Trans. on Intell. Transp. Sys.*, 13(1):365–374, Mar. 2012.

[3] S. Chakraborty, M. Lukasiewycz, C. Buckl, S. Fahmy, P. Leteinturier, and H. Adlkofer. Embedded systems and software challenges in electric vehicles. *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 424–429, Mar. 2012.

[4] N. Dalal and W. Triggs. Histograms of Oriented Gradients for Human Detection. *CVPR*, 1(3):886–893, 2004.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on PAMI*, 32(9):1627–45, Sept. 2010.

[6] P.-Y. Hsiao, C.-W. Yeh, S.-S. Huang, and L.-C. Fu. A Portable Vision-Based Real-Time Lane Departure Warning System: Day and Night. *IEEE Trans. on Vehi. Tech.*, 58(4):2089–2094, 2009.

[7] Z. Lin, J. Sankaran, and T. Flanagan. Empowering automotive vision with TIs Vision AccelerationPac. *Texas Instruments White Papers*, pages 1–7, 2013.

[8] J. McCall and M. Trivedi. Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation. *IEEE Trans. on Intell. Transp. Sys.*, 7(1):20–37, Mar. 2006.

[9] R. K. Satzoda and M. M. Trivedi. Selective Salient Feature based Lane Analysis. In *2013 IEEE Intell. Transp. Sys. Conf.*, pages 1906–1911, 2013.

[10] R. K. Satzoda and M. M. Trivedi. Vision-based Lane Analysis : Exploration of Issues and Approaches for Embedded Realization. In *2013 IEEE CVPR Workshops on Embedded Vision*, pages 604–609, 2013.

[11] S. Sivaraman and M. M. Trivedi. A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking. *IEEE Trans. on Intell. Transp. Sys.*, 11(2):267–276, June 2010.

[12] S. Sivaraman and M. M. Trivedi. Integrated Lane and Vehicle Detection , Localization , and Tracking : A Synergistic Approach. *IEEE Trans. on Intell. Transp. Sys.*, pages 1–12, 2013.

[13] S. Sivaraman and M. M. Trivedi. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Trans. on Intell. Transp. Sys.*, 14(4):1773–1795, Dec. 2013.

[14] F. Stein. The challenge of putting vision algorithms into a car. *2012 IEEE CVPR Workshops on Embedded Vision*, pages 89–94, June 2012.

[15] M. M. Trivedi, T. Gandhi, and J. McCall. Looking-In and Looking-Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety. *IEEE Trans. on Intell. Transp. Sys.*, 8(1):108–120, Mar. 2007.