On Enhancing Lane Estimation using Contextual Cues

Ravi Kumar Satzoda, Member, IEEE and Mohan M. Trivedi, Fellow, IEEE

Abstract—Vision-based lane detection is a critical component of modern automotive active safety systems. Although a number of robust and accurate lane estimation algorithms have been proposed, computationally efficient systems that can be realized on embedded platforms have been less explored and addressed. This paper presents a framework that incorporates contextual cues for lane estimation in order to further enhance the performance in terms of computational efficiency and accuracy both. The proposed context-aware lane estimation framework (CLEF) takes into account the state of the ego-vehicle, its surroundings and the system-level requirements to adapt and scale the lane estimation process resulting in substantial computational savings. This is accomplished by synergistically fusing data from multiple sensors along with the visual data to define the context around the ego-vehicle. The context is then incorporated as an input to the lane estimation process to scale it depending on the contextual requirements. Detailed evaluation of the proposed framework on real-world driving conditions shows that the dynamic and static configuration of the lane detection process results in computation savings as high as 90%, without compromising on the accuracy of lane estimation.

Index Terms—lane detection, active safety systems, advanced driver assistance systems, computational efficiency, adaptive, scalable, context-aware

I. INTRODUCTION

Among the increasing number of embedded electronic subsystems in modern vehicles, intelligent driver assistance systems form a significant component [1]–[3]. Among these different electronic subsystems, vision-based advanced driver assistance systems (ADAS) form a significant percentage [3], [4]. Cameras are being used to monitor 360° surround of the vehicle [3] for both active safety systems and naturalistic driving data analytics [5], [6]. Although visual data provides a rich set of information, robustness of vision algorithms under varying road and environmental conditions is still a challenging task [7]. In order to achieve high accuracy, data-intensive and computationally-intensive computer vision algorithms are employed, which burden the resource constrained embedded electronic systems in terms of battery power [8]. Additionally, real-time operation is also a necessary requirement for such active safety systems [3], [9].

Most vision based algorithms involve data-intensive operations such as filtering, thresholding, transformations etc. [9]–[11], that are applied directly on pixel intensities and coordinates in the input video frames. According to [9], the amount of visual data that is processed directly to the computational load in the vehicle. Therefore, this necessitates the design of accurate and robust techniques that operate on the required amount of visual data without over burdening the computational resources in the vehicle.

In this paper, we will particularly focus on lane estimation, which has been established in [12], [13] as one of the key functional components in ADAS such as lane departure warning (LDW) systems, lane change decision aid systems [14], lane change intent prediction systems [15] etc. In this paper, we propose a framework that synergistically fuses the contextual information to enhance the efficiency of vision-based lane estimation process. The proposed framework called Context-aware Lane Estimation Framework (CLEF) is shown to significantly reduce data-intensive operations resulting in orders of magnitude of computational savings, without affecting the robustness of the lane estimation process. CLEF uses information from other sensing modalities to determine the state of the ego-vehicle and its surroundings such as traffic and presence of other vehicles. This contextual information is then used to scale the lane detection process such that the required amount of input frame data is processed to extract lanes, which meet the requirements of the context. Based on the survey of existing lane estimation techniques in literature and to the best of our knowledge, the proposed context-aware framework for lane detection is the first of its kind.

The rest of the paper is organized as follows. In Section II, a survey on existing lane estimation methods is presented with a focus on the computational aspects. The outline of CLEF is introduced in Section III. An introduction to existing lane detection method called LASeR is presented in Section IV, which is redesigned to incorporate contextual information in Section V. The proposed context-aware framework CLEF is presented in Section VI, followed by a computational architecture for CLEF in Section VII. A detailed evaluation of CLEF is presented in Section VIII and IX, followed by conclusions and future directions in Section X.

II. SURVEY OF RELATED WORK

Before presenting the proposed framework, a survey of recent related studies in the area of lane estimation is presented in this section. Most surveys on lane estimation such as [12], [13], [22] are comprehensive and complete with detailed analysis of the robustness and functional aspects of the lane estimation algorithms. The brief survey in this paper is particularly aimed at the computational aspects, and computational optimizations (if any) of some of the recent lane estimation algorithms.

Table I lists some recent lane estimation methods. The computational operations for each algorithm are listed under second column. As discussed earlier, all vision-based lane
estimation algorithms listed in Table I involve data-intensive operations. Operations such as 2D filtering, edge detection, gradient computation, kernel template matching, gray-level pixel based feature extraction etc. are pixel-based operations, i.e. these operations are performed on each pixel coordinate using the gray or color intensities. These operations are computationally repetitive and SIMD (single instruction multiple data) processor architectures are most suitable for computing such operations [20]. In addition, computationally demanding image processing algorithms such as Hough transform, RANSAC, model fitting etc. are performed on features such as edge pixels to extract the final lane features. For example Hough transform is used in [16], [23] to determine edge pixels that fall on straight lines which are then extracted as lanes. In addition to the above steps, higher-level operations such as lane tracking using Kalman filtering is a common operation in most lane estimation methods.

The third column in Table I shows the speed of operation and the platform on which the algorithms are implemented. It can be seen that most algorithms listed in Table I are implemented on high-end PC based systems with high clock-speed processors. Although monocular camera based methods such as [13], [19] report real time operation, they are implemented on PC-based hardware resources, which do not lend well for power constrained embedded realization. Similarly [18] is a stereo-based method which has its own computational challenges. Although [20], [21], [24]–[27] have implemented various versions of lane estimation methods on FPGA platforms using different combinations of above mentioned operations, [20], [21] listed in Table I can be considered as complete hardware implementations of the lane estimation algorithms. The other works have reported hardware architectures for lane estimation with varying degrees of completeness. As shown in Table I, [20], [21] show real-time operational speeds of 30 to 40 frames per second on Xilinx Spartan FPGAs. A further study of the constituent algorithms in [20], [21] shows that they involve processing the entire image region of interest (RoI) and there is no architectural optimization performed at the algorithmic level to improve the computational efficiency. Although they achieve real-time operation, the computational complexity of the algorithms in [20], [21] can be an obstacle for conserving power, which is an important metric especially in the case of automobiles that are trending towards electric and hybrid systems.

III. OVERVIEW OF PROPOSED CONTEXT-AWARE LANE ESTIMATION FRAMEWORK (CLEF)

The motivation behind the proposed context-aware lane estimation framework (CLEF) can be explained as follows. Existing lane estimation methods apply the same algorithm without considering the context in which lane estimation needs to be performed. For example, when the traffic is high or when there is a leading vehicle in front of the ego or host vehicle, drivable lanes only need to be estimated [28]. In fact, estimating lanes beyond the obstacles in such scenarios leads to errors [19]. Therefore, adapting lane estimation algorithm to the context aids in: (a) improving the accuracy of the lane estimation process, and (b) avoiding overkill of computation resources.

![Fig. 1](image-url)  
Fig. 1. An overview of the proposed framework. The gray windows indicate the contextual cues that are input to Context Definition Module, which then defines how Lane Detection Module should function.

A. Context Definition Module

The context definition module determines the context in which lane estimation is to be performed. The inputs to this module are of two functional modules: (a) context definition module, and (b) lane estimation module. We will briefly discuss the two modules in this section before describing them in detail in Sections IV and V.
constraints and accuracy specifications. Also, the application of the system can also be used to define the context. For example, a lane change assistance system would require lane detection in far depth of view, whereas a lane keep assistance system will require accurate lane estimation in the near depth of view.

In this paper, we limit the scope to the first set of inputs to the context definition module. The proposed framework in this paper is designed for scaling the lane estimation process based on the real-time dynamics of the ego-vehicle and its surroundings.

B. Lane Estimation (LE) Module

The second functional module in Fig. 1 is the LE algorithm. The context definition module defines the context in which LE must be performed, and sends a set of control signals to the LE module which are used to scale and adapt the LE module. There are a number of lane estimation algorithms in open literature but they are not inherently catered towards adaptability to the contextual information. In this paper, one of the recently proposed lane estimation technique called LASeR (lane analysis using selected regions) [29], [30] is redesigned to include the contextual information. Some algorithms such as [16] could also be developed in similar ways to cater to the contextual inputs.

In the forthcoming sections, we first describe LASeR algorithm in Section IV. Thereafter, in Section V, we will show how LASeR is configured and scaled using the contextual cues that are extracted by the context definition module.

IV. LANE ESTIMATION USING LASeR

In LASeR, selected bands are used to detect lane features as shown in Fig. 2. An image $I$ is first converted into its inverse perspective mapping (IPM) [31] image $h$ resulting in the top view of the road scene. In this IPM image, $N_B$ scan bands, each of height $h_B$ pixels, are selected along the vertical $y$-axis, i.e. along the road from the ego-vehicle. Each band $B_i$ is then convolved with a vertical filter that is derived from steerable filters [32]. The vertical filter is represented by the following equation:

$$G^0(y,x) = \frac{-2x}{\sigma^2}e^{-\frac{x^2+y^2}{\sigma^2}}$$  \hspace{1cm} (1)

where $G(x,y) = e^{-\frac{x^2+y^2}{\sigma^2}}$ is a 2D Gaussian function. In LASeR $G^0(x,y)$ is a $5 \times 5$ 2D filter. Therefore, the filter response $B_i^S$ for each scan band $B_i$ is given by

$$B_i^S = B_i * G^0(x,y).$$  \hspace{1cm} (2)

$B_i^S$ from each band is then analyzed for lane features by thresholding using two thresholds to generate two binary maps $E_+$ and $E_-$ for each band. Vertical projections $p_+$ and $p_-$ are computed using $E_+$ and $E_-$, which are then used to detect lane features using a shift-and-match operation (SMO) in the following way:

$$K = (p_+ \ll < \delta >) \odot p_-$$  \hspace{1cm} (3)

where $\odot$ represents point-wise multiplication, $\delta$ is the amount of left shift (denoted by $<< \delta >$ above) that the vector $p_-$ undergoes. The SMO enables detection of adjacent light to dark and dark to light transitions, which characterize lane features. On applying a suitable threshold on $K$ and road model, we get the positions of left and right lanes in the $B_i$-th scan band denoted by

$$P_{B_i} = \{P_L(x_L,y_L), P_R(x_R,y_R)\}$$  \hspace{1cm} (4)

Therefore, we get $N_B$ such positions in $N_B$ scan bands (as shown in Fig. 2) resulting in $P = \{P_{B_i}\}$, which are associated with each other using a road model, and are also tracked using extended Kalman tracking. More details are explained in [30]. In the forthcoming sections, we will show how LASeR is scaled using the contextual information.

V. CONTEXT DEFINITION USING STATES OF EGO-VEHICLE & ITS SURROUNDINGS

In this section, we propose techniques to define the contextual parameters based on the vehicle dynamics and its surroundings, which will be employed for dynamically scaling the lane detection method. The following three different contexts are considered as part of this study.

A. Proximity of Leading Vehicle

Presence of a leading vehicle in front of the ego or host vehicle is considered as one of the factors for defining how much of lane or road needs to be detected [28]. In KITTI road and lane evaluation datasets [28], road surface that is detected till the leading vehicle is called drivable lane region.
In the proposed context-aware framework CLEF, the distance of the leading vehicle from the ego-vehicle is used to configure the amount of visual data that needs to be processed for lane detection. In order to do this, we use the distance information obtained from the front radar sensor in our testbed. It is to be noted that a large percentage of modern consumer cars are equipped with ACC (adaptive cruise control), which employs a similar radar sensor. As shown in Fig. 2, let \( Y = \{ y_j \} \), where \( 1 \leq j \leq N_B \) be the set of positions of the \( N_B \) scan bands along the \( y \)-axis, i.e. along the road in the IPM image \( I_W \). Given \( d_l \) is the distance (in meters) of the leading vehicle from the ego-vehicle (obtained from the radar), the maximum number of bands in LASeR, i.e. \( N_{bl}^{d_l} \), that need to be processed is given by:

\[
N_{bl}^{d_l} = \arg \max_j \psi(d_l) \leq y_j \tag{5}
\]

where \( \psi(d_l) \) is the transformation function to convert the distance \( d_l \) to IPM \( y \)-axis scale. The scale function is obtained from one-time calibration of the camera setup. Therefore in every frame, the distance of the leading vehicle from the radar is used to determine \( N_{bl}^{d_l} \), and \( N_{bl}^{d_l} \) scan bands only are processed to determine lane features in LASeR.

B. Speed of Ego-Vehicle & Traffic Condition

Vehicle speed is a useful cue especially in highway driving because the vehicles are expected to be moving at a particular speed limit. Therefore, the speed of the ego-vehicle can also be used to determine the traffic condition. This is illustrated using the example shown in Fig. 3, which shows the speed profiles of two different drives that were conducted on a highway. It can be seen that the speed profile of drive 1 with lower speed than drive 2 corresponds to the higher traffic density.

![Fig. 3. Speed profiles for two different drives. Given that the vehicle is on freeway, the speed profile can be used to determine the state of traffic during the drive. Thumbnails show the visual information of the scene during the drive.](image)

From Fig. 3, it can also be seen that in high traffic scenario, the lanes are not visible beyond the nearest vehicle. Therefore, processing the entire image or the entire RoI below the vanishing line for lane estimation is an overkill of computational resources. If \( v_e \) is the speed of the ego-vehicle which is traveling at an acceleration of \( a_e \), the stopping distance within the time to collision (TTC) (denoted by \( T_c \)) is given by:

\[
d_c = v_e T_c + \frac{1}{2} a_e T_c^2 \tag{6}
\]

TTC is a commonly used timing measure in ADAS for active safety. In most literature [33], TTC is commonly set to 1.5 seconds. TTC is also shown as a variable which can be implemented as a look-up table (LUT) for varying speeds and accelerations [33]. Therefore, given TTC, \( v_e \) and \( a_e \), the distance \( d_c \) is calculated using (6), which is then used to determine the maximum number of scan bands \( N_{bl}^{d_c} \) using (5). The resulting \( N_{bl}^{d_c} \) scan bands in LASeR are processed for lane feature extraction.

1) Combining Vehicle Proximity and Speed of Ego-Vehicle:

Now assuming we have both the radar information and the ego-vehicle dynamics, we combine the proximity of the leading vehicle \( d_l \) with the speed of the ego-vehicle \( v_e \) to determine the maximum number of scan bands that are needed for lane detection in CLEF. The relative distance \( d_l \) between the ego-vehicle and the leading vehicle can be used to determine the relative velocity between the two vehicles using the following equation:

\[
v_e^r = \frac{d_l^{t+1} - d_l^t}{\Delta t}. \tag{7}
\]

Acceleration is assumed to be zero in the time interval \( \Delta t \) in the above equation. If \( v_e^r > 0 \), the leading vehicle is moving away from the ego-vehicle, otherwise the ego-vehicle is approaching the leading vehicle from rear. The presence or absence of a leading vehicle, its relative velocity \( v_e^r \) and distance \( d_l \) from the ego-vehicle, and the vehicle dynamics of the ego-vehicle (i.e. \( v_e \) and \( a_e \)) can be combined in the following way to determine the maximum number of scan bands that capture the lanes, which lie between the ego-vehicle and the leading vehicle:

\[
N_{max} = \begin{cases} 
\arg \max_j \psi \left( \frac{v_e^r}{\Delta t} \right) \leq y_j & \text{if } v_e^r < 0 \land \frac{v_e^r}{\Delta t} < d_l \\
\arg \max_j \psi(d_l) \leq y_j & \text{if } v_e^r < 0 \land \frac{v_e^r}{\Delta t} > d_l \\
\min(N_{bl}^{d_l}, N_{bl}^{d_l}) & \text{if } v_e^r \geq 0
\end{cases}
\tag{8}
\]

where \( N_{bl}^{d_l} \) and \( N_{bl}^{d_l} \) are computed using (5) and (6). The first two formulations in (8) determine the number of bands for the case when the ego-vehicle is approaching a leading vehicle from rear. The third formulation refers to the case when the leading vehicle is moving away from ego-vehicle. In such a scenario, the minimum of the two distances \( d_l \) and \( d_e \), i.e. leading vehicle distance and the time to crossing distance from (6), is used. The is because the minimum of \( d_l \) and \( d_e \) forms the region of interest where the lanes need to be detected. In other words, if \( d_l < d_e \), then there is a leading vehicle at a distance lesser than the distance covered within the time to crossing. Similarly, if \( d_e < d_l \), then the distance covered within the time to crossing is lesser than the distance of the leading vehicle. Therefore, taking the minimum of the two distances for determining the maximum number of scan bands gives the required number of scan bands for the context.

C. Curvature of the Road

We now analyze the curvature of the road to define the context for lane estimation. The presence of a curved lane implies larger number of lane feature points that will be required to fit a curved road model. Otherwise, a straight lane model can be used, which requires a minimum of two points to draw the straight lane. In CLEF, the knowledge of the curvature can be used to determine the number of scan bands...
that will be required to give the necessary number of lane feature coordinates, which will then be used for polynomial fitting. For example, in [30], a linear-parabolic model is used for curved lane fitting in LASeR and the accuracy of the curved lane estimation increases if the number of the scan bands is increased.

The curvature $C$ of the lane is defined by the following:

$$C = \frac{d\theta}{ds}$$

where $d\theta$ is the change in angle of the tangential motion vector of the vehicle, and $ds$ is the distance traversed by the vehicle along the curve. $d\theta$ can be determined using the yaw angle of the vehicle, which can be obtained from the in-vehicle sensors. Studies such as [34], [35] have shown how such vehicle dynamics can be accessed either from the in-vehicle CAN bus or external modules such as modern smartphones. The distance traveled in $\Delta t$ is given by $ds$, which is computed using the speed and time information (assuming constant velocity during $\Delta t$). If $C$ is greater than a positive threshold $T_c$, then the road is curving to the right. Similarly if $C < T_c$, then the road is curving to the left.

Given $C$, CLEF changes the number of scan bands for lane detection in LASeR. When $C$ is higher, the number of scan bands is higher and equally spaced out along $y$-axis of the IPM image $I_B$ so that more lane feature points are captured for polynomial curve fitting. In our experiments, we indicate $|C| > 0.05$ to indicate a higher curvature, wherein all the scan bands are made active. If the curvature is lesser than 0.05, alternate scan bands are active to determine the lane feature positions in the scan bands, which are connected using a straight lane model.

1) Note on Other Modalities for Curvature Estimation:
Digital maps are an alternative to estimate the road curvature. Given the GPS coordinates of the ego-vehicle, the road information can be extracted by referring to a digital map. While this is a more straightforward solution, this also requires the availability of GPS signals and detailed digital maps during the drive. Therefore, when there is limited network coverage, dependency on GPS signals will fail the system. Similarly, digital maps are not always sufficiently descriptive, and are dependent on the providers. However, such a system could be deployed in urban environments where GPS signals and maps are available adequately.

VI. CONTEXT-AWARE LANE ESTIMATION FRAMEWORK (CLEF)

In the context-aware lane estimation process, we use the contextual information, that is derived from the state of the ego-vehicle and its surroundings, to do the following:

- The scan bands in LASeR that are required by the context are accessed from the input image.
- The selected bands only are processed further to determine the lane features.

In contrast to the conventional LASeR, the above steps ensure that the required amount of image only is accessed and processed.

Algorithm 1 lists the context-aware lane estimation framework. Given the input image frame $I$, the ego-vehicle dynamics, i.e. velocity $v_e$, acceleration $a_e$, and the distance of leading vehicle $d_l$, the following parameters are computed:

- the maximum number of bands parameterized by the distance of leading vehicle $N_{max}$ using (5).
- the maximum number of bands parameterized by the velocity of the ego vehicle $N_{vmax}$ using (6) and (5).
- the curvature of the vehicle trajectory $C$ using (9)

Algorithm 1: Context-aware lane estimation framework CLEF

1. **Inputs:** Image frame $I$, ego-vehicle dynamics $v_e$, $a_e$, $	heta$, and distance of lead vehicle $d_l$
2. Compute $N_{max}$, $N_{vmax}$ and $C$.
3. Compute $N_{max}$ using $v_e$.
4. if ($|C| > 0.05$) then
5. Set bands list to $B = \{B_i\}$, where $1 \leq i \leq N_{max}$
6. else
7. Set bands list to $B = \{B_j\}$, where $j = \{1, 3, \ldots, N_{max}\}$
8. end if
9. for every band $B_k \in B$ do
10. Generate IPM for $B_k$, apply $G^\theta$ & SMO on $B_k$
11. Determine lane positions $P = \{P_{B_k}\}$
12. end for
13. Use road model for lane visualization and tracking using the lanes feature positions in $P$. 

VII. COMPUTATION ARCHITECTURE FOR CLEF

In this section, we present an SIMD (single instruction multiple data) computation architecture for CLEF. Fig. 4(a) shows the three main computational blocks of CLEF. First, the image is converted to its bird’s eye view using inverse perspective mapping (IPM). This is accomplished by a look-up table because the mapping of an image to its IPM domain is dependent on the pixel coordinates, which can be computed offline and stored as a look-up table. Next, the context definition module computes $N_{max}$ and the set of bands $B$ that must be processed based on the contextual information from the ego-vehicle dynamics and its surroundings. The third block is the...
processing engine (PE) array, which consists of \( N_B \) number of PEs. Each \( P_E_i \) processes the image intensities from one scan band \( B_i \) in the IPM image. In addition to the input from the IPM, the contextual parameters act as power control signals to the PE. If band \( B_i \in B \), then \( P_E_i \) is powered on, otherwise it is powered off. Therefore, the PEs will be operational if and only if the context requires them. After all the PEs process their respective scan bands, the lane marker positions are further processed for outlier removal and road model fitting, which are higher-order operations in the lane estimation process.

The architecture in Fig. 4(a) is an SIMD architecture because each PE is similar in operation but data streams are different for each PE. The PE is described by the architectural block diagram in Fig. 4(b). Given a scan band \( B_i \), for each \( x \)-position (horizontal axis), the pixels are read along the \( y \)-axis in the scan band. The convolution block \( F \) in Fig. 4(b) takes a \( 5 \times 5 \) neighborhood around a given pixel \( I_W(x,y) \) in the scan band to convolve with the \( G^0 \) kernel that was derived in (1). Therefore, the result of this convolution is given by \( B^0_i(x,y) \), i.e., \( B^0_i(x,y) = \sum_{u,v \in k,l} I_W(u,v) G^0(k,l) \), where \( u, v \) define the \( 5 \times 5 \) neighborhood around a pixel in \( I_W \). The comparators \( C_+ \) and \( C_- \) in Fig. 4(b) are then used to threshold \( B^0_i(x,y) \). The results from the comparators are either 1 or 0. Next, the adders \( A_+ \) and \( A_- \) help to accumulate the number of 1’s that are being output from the comparators \( C_+ \) and \( C_- \) for each \( x \)-coordinate. The \( x \)-coordinate is used to generate an address for buffers \( p_+ \) and \( p_- \) (shown in Fig. 4(b)), where the accumulation results are stored.

After scanning the entire scan band, i.e. \( 0 \leq x \leq M_W \), the address generator uses \( x \)-coordinate to create two addresses \( x_+ = x \) and \( x_- = x + \delta \) in order to perform the shift and match operation described in (3). The multiplier \( M \) in Fig. 4(b) multiplies the accumulation results stored in \( p_+ \) and \( p_- \) that are addressed by \( x_+ \) and \( x_- \). The multiplier output is compared against a threshold using \( C_K \) comparator in Fig. 4(b). If the output of \( C_K \) is high (or 1) then the \( x \)-position is considered to have a lane marker, which is stored in \( K_B_i \) buffer for each scan band. This architecture for CLEF can be implemented in multiple ways depending on the resource constraints, i.e. either as a parallel architecture (one PE per scan band) or as a serial architecture.

VIII. METRICS FOR EVALUATION

Considering that lane estimation process requires both computational efficiency and accuracy, the following metrics are introduced to evaluate CLEF.

A. Accuracy Metrics

1) Lane Position Deviation (LPD): In this paper, we use lane position deviation (LPD), which measures the pixel level deviation of the estimated lanes against the ground truth. Referring to Fig. 5, which shows a possible input image scene, let us consider the solid line to be the actual lane boundary in the ground truth for a given image, which is manually marked. Let us consider the left lane boundary and let the dashed line indicate the lane position that is determined by LASeR. The LPD metric determines the average absolute deviation \( \delta_{LPD} \) in the \( x \)-direction between the actual and detected lane positions, between \( y_{max} \) and \( y_{min} \) positions in the image scene, i.e.,

\[
\delta_{LPD} = \frac{1}{y_{max} - y_{min}} \sum_{i=y_{min}}^{y_{max}} \delta_i,
\]

where \( y_{min} \leq i \leq y_{max} \) is the selected region of the road scene in which the lane deviation is measured, and \( \delta_i \) is the absolute deviation between the estimated lane position and the ground truth along \( x \)-axis at \( y = i \). From the above equation and Fig. 5, and depending on the region of interest that is defined by the context between \( y_{min} \) and \( y_{max} \), LPD gives an accurate measure for lane estimation accuracy for both near and far depths of view.

2) Ego-Vehicle Localization: In order to compute ego-vehicle localization \( x_V \), the lane positions determined from the near-view of the vehicle are used to compute the distance...
between the center of the vehicle $x_C$ to the left $x_L$ or right lane $x_R$ position, i.e. $x_V = x_C - x_L$. In most scenarios $x_C$ is the middle of the image frame if the camera is mounted in the middle of the vehicle. $x_V$ can be scaled further to get the drift in vehicle position in centimeters. The ground truth is generated by either manually marking the lane position (say left lane) at the bottom of input video frame or by using a semi-automated method described in [13]. The evaluation determines the average deviation error of the lane position from the center of the car that is obtained from the algorithm against the ground truth for every frame. This metric evaluates the accuracy of the lane positions that are located in the near depth of view of the ego-vehicle. Therefore, this metric is applicable in scenarios where the context is set to the near view of the vehicle such as high traffic density or slow moving ego-vehicle.

B. Computational Cost Metrics

Computational cost is the next metric that will be used for evaluation in this paper. We define computation cost in two different ways. First, without loss of generality, we define computation cost $C_e$ being proportional to the number of pixels $N_p$ that are being processed to estimate the lane features, i.e. $C_e \propto N_p$. This measure for cost is only an estimate and other more sophisticated cost measures can be defined [29]. Most existing methods like [13], [16] process the entire IPM resulting in the cost $C_{exist} = M_W \times N_W$ for an IPM image of size $M_W \times N_W$ pixels, where $N_W$ is the height and $M_W$ is the width of the IPM image. In the case of conventional LASeR algorithm, $N_B$ number of scan bands are sampled at specific positions in the IPM image. If each scan band is of height $h_B$ pixels, then the cost for LASeR is given by $C_{LASR} = (h_B \times M_W)N_B$.

In LASeR and the methods in [20], [21] etc., the computation cost is the same for every frame. However, in CLEF, the computation cost could change in every frame based on the contextual cues. Therefore, given a video sequence with $N_F$ number of frames, we define the following metric for computation cost for CLEF:

$$\begin{align*}
C_{CLEFavg} = \frac{1}{N_F} \sum_{i=0}^{N_F} \frac{(h_B \times M_W)N_{max_i}}{N_{max_i}}
\end{align*}$$

$C_{CLEFavg}$ gives the average computation cost for lane estimation using CLEF for a video sequence with $N_F$ frames. $N_{max_i}$ in the above equation refers to the maximum number of scan bands that the context definition module determines for the $i$-th frame.

The second computational cost metric is more elaborate, and is computed using the number of operations involved in the lane estimation algorithm. For each technique, the number of additions, multiplications, comparisons, divisions and trigonometric operations are determined. We elaborate more on these metrics in Section IX-C. In the forth coming sections, CLEF is compared against LASeR [29] and two hardware-centric methods in [20] and [21] using the different computational cost metrics.

IX. PERFORMANCE EVALUATION

In this section, we will evaluate and demonstrate the proposed context-aware lane analysis process using real-world driving data for varying road scenes. The experiments were conducted on the datasets captured using the LISA-Q and LISA-X testbeds [13] that are equipped with an ACC radar, forward facing camera and additional sensors like accelerometers, gyroscopes etc. The datasets with over 5000 frames of data were collected under different lighting and lane conditions like daytime, nighttime, high and low traffic to demonstrate how the proposed framework can adapt the lane detection process for varying conditions. The scope of the performance evaluation in this paper is set to the data-intensive lane feature extraction part of the lane estimation process because tasks such as lane tracking and road model fitting are dependent on the lane features. Therefore, the cost of other tasks remains the same for all methods.

### TABLE II

<table>
<thead>
<tr>
<th>Number of scan bands</th>
<th>Day Position Deviation (cm)</th>
<th>Night Position Deviation (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h_B = 5$</td>
<td>$h_B = 10$</td>
</tr>
<tr>
<td>$N_B = 8$</td>
<td>9.14</td>
<td>9.73</td>
</tr>
<tr>
<td>$N_B = 16$</td>
<td>6.11</td>
<td>7.12</td>
</tr>
</tbody>
</table>

A. Accuracy versus Computational Efficiency of Different Configurations of LASeR

CLEF uses the number and height of scan bands ($N_B$ and $h_B$) to scale LASeR for varying contextual requirements and computational constraints. We will first evaluate the tradeoff between accuracy and computational efficiency that results from changing $N_B$ and $h_B$ in LASeR.

Table II shows the mean $\delta_{LPD}$ (calibrated in centimeters) that is determined over all the input frames in a given dataset. We show the deviation for two datasets corresponding to daytime and nighttime scenarios, and for different number of scan bands and their heights. It can be seen that as we decrease the number of bands the deviation increases because lesser information is available for lane feature extraction. However, it is noteworthy that processing just 8 or 16 scan bands of height which is no more than 10 pixels gives a maximum lane position deviation of 10cm only (from ground truth). In fact, for 16 scan bands of height 5 pixels, the deviation is less than 7 cm from ground truth, which is less than 5% deviation when compared to the lane width of 3.5 m.

Next, Fig. 6 compares the computation cost between the different configurations of LASeR, that can be chosen based on the contextual constraints. It can be seen that changing the number of scan bands that are processed in LASeR reduces its computational cost by at least 4 times and 8 times when 16 and 8 scan bands are used respectively, as compared to conventional lane detection methods where the whole IPM is processed.

Now let us see the tradeoff between computation cost savings and accuracy of lane analysis. Fig. 7 shows a scatter plot between the computation cost of the various configurations of
Fig. 6. Computation cost savings of LASeR as compared to [13] which processes entire IPM image.

LASeR against their corresponding percentage lane deviation for the daytime dataset. The percentage lane deviation in % is computed as: lane deviation/actual lane width × 100. It can be seen that using 8 or 16 scan bands gives a percentage lane deviation to be less than 2.7%. This shows that the large computation cost savings (> 400%) does not penalize the accuracy of the lane detection process.

The ego-vehicle localization metric helps to understand the effectiveness of the proposed method to accurately determine the lane positions in near-view, which is particularly useful in the case for contexts set by slow moving ego-vehicle and high traffic densities (Sections V-A and V-B).

It can be seen from Table III that LASeR gives better results in terms of mean absolute error (in cm) for day time scenario as compared to VioLET in [13]. In the case of nighttime dataset, the mean absolute error of LASeR is 5mm more than VioLET, which is an error of 1.5% when it is seen as a ratio of the lane width. However, it can be seen that LASeR has lesser standard deviations than VioLET in both daytime and nighttime datasets. CLEF improves upon LASeR in terms of the accuracy. It can be seen from Table III that CLEF gives lesser mean absolute errors and standard deviations for both daytime and nighttime conditions as compared to LASeR. The mean absolute errors are reduced by as much as 0.45cm as compared to LASeR. Fig. 8 shows two sample frames with the lane estimation results from LASeR and CLEF in Columns (a) and (b) of Fig. 8 respectively. It can be seen that the lane estimation in LASeR is affected by the presence of the vehicle where the estimated lanes seem to be incorrectly positioned towards the farther depth of view. CLEF based estimation in Column (b) detects the presence of the vehicle, and therefore estimates the lanes till the leading vehicle only.

B. Accuracy of Lane Detection in Near-view: Ego-vehicle Localization

We compare the ego-vehicle localization metric for lane estimation using CLEF against the full IPM based lane estimation algorithm (VioLET) in [13] and the scan-band based lane estimation algorithm (LASeR) in [29] in Table III. The mean absolute errors are reduced by as much as 0.45cm as compared to LASeR.
C. Evaluation of Computation Operations

In Table IV, we list the number of computation operations such as additions, multiplications etc. for the lane feature extraction step in CLEF, LASeR and the methods in [20] and [21]. The expressions in Table IV are derived based on the different image processing operations for lane feature extraction in the four methods. The expressions are based on $M \times N$ (columns by rows) sized images. $w_G \times w_G$ sized kernels are considered for the 2D filtering operation. In [20], a different $w_M \times w_M$ sized kernel is used for morphological operations. $\beta \in [0, 1]$ is a factor that is used to indicate the percentage of the image that is used for lane estimation, i.e. to indicate the region of interest such as the lower 3/4-ths of the image. An additional factor $\alpha$ is used to indicate the percentage of the image which are edge pixels that will be processed further for lane feature extraction. In [21], Hough transform (HT) is used and the number of angles for which HT is computed is denoted by $N_\theta$. In the case of LASeR and CLEF, $h_B$ and $M_B$ indicate the height of each scan band and the width of the IPM image respectively. The main difference between the expressions in LASeR and CLEF is that the number of scan bands is fixed to the maximum number $N_B$ in the case of LASeR. However, in CLEF this is determined by the context definition module (denoted by $N_{max}$ in Table IV). Also, it is to be noted that the expressions for CLEF in Table IV refer to the number of operations per frame, which will vary for every frame depending on $N_{max}$.

![Comparison of Number of Computations per Frame](image)

**Fig. 9.** Total number of operations for different techniques.

Table V compares the number of computations that are incurred in different methods for estimating lanes in the sequence shown in Fig. 10. As discussed previously, the number of computations to extract lane features for each frame $h_B = 10$ are considered. For CLEF, $N_B = 8$, $h_B = 10.5$ are set to give two different configurations CLEF-1 and CLEF-2 in Fig. 9. The width of the IPM image is set to $M_W = 500$, although lower sized IPM (as low as $M_W = 300$ can be set in LASeR and CLEF).

The y-axis in Fig. 9 is set to log scale in thousands. It can be seen from Fig. 9 that CLEF shows orders of magnitude savings in the number of computational operations as compared to LASeR, [20] and [21]. Such significant savings in the number of operations makes CLEF an ideal choice for embedded realization.

### D. Demonstration of CLEF using Real-world Driving Data

We will now present the results of lane estimation using proposed CLEF that incorporates the contextual information discussed in Sections V and VI. Fig. 10 shows the results of lane detection using the distance of the leading vehicle (from radar sensor) in Fig. 10 (a), and ego-vehicle speed to determine the distance within time to collision (TTC = 1.5 seconds) in Fig. 10 (b). It can be seen in Fig. 10 (a) that as the ego-vehicle approaches the vehicle in front, CLEF is able to scale the lane detection process by reducing the number of scan bands that need to be processed to detect the lanes. In Fig. 10 (b), the lanes are detected within the distance where a collision can occur, which is determined by the TTC. It can be seen in Fig. 10 (b) that the number of scan bands that are processed in the first frame of the sequence is lesser than the scenario when we use distance of the leading vehicle in Fig. 10 (a). This is because the speed of the ego-vehicle is such that it is in a risk of a collision within the limited distance as shown by the lanes estimated by the framework. Fig. 10 (c) shows the number of scan bands that are being processed in every frame of the sequence for the two scenarios in Fig. 10 (a) & (b). It can be seen that using the ego-vehicle velocity and time to collision uses lesser number of scan bands resulting in an average computation cost savings of nearly 50% as compared to the case when leading vehicle distance is used. Additionally, the lane position deviation was found to be less than 7cm in both scenarios.

![Comparison of Computation Cost](image)

**Fig. 10.** Results of lane estimation using CLEF.
is the same in the case of algorithms in [20], [21] and LASeR [29]. CLEF will have varying number of computations per frame for the two contextual parameters that are demonstrated in Fig. 10. Therefore, in the case of CLEF, we compute the mean, maximum and minimum values for the number of computations per frame.

The average number of computations required by CLEF for every frame is nearly 90% lesser than both [20] and [21]. As compared to LASeR, we see the saving is nearly 30% per frame. This is because LASeR in [29] is designed to be computationally more optimized than conventional lane estimation methods in [20] and [21]. In order to remove any further doubts about the computationally efficiency by CLEF, we also list the total number of computations for extracting lane features from the entire sequence. CLEF shows computational savings of over 94% as compared to [20] and [21], and 67% when compared to LASeR algorithm. Therefore, it can be seen that the contextual information about the vehicles around the ego-vehicle and the ego-vehicle’s own dynamics can be used to substantially reduce the computational load without compromising on the accuracy of the lane detection process.

1) Note on Power Savings using CLEF: CLEF is shown to reduce the overall computation costs by at least 60% when compared to existing techniques. In this section, we briefly discuss how the savings in the computational operations in CLEF will also result in reducing power consumption. The dynamic power dissipation in a CMOS circuit can be defined as [37], [38]:

\[
P_D = \left( C_{pd} \times f_1 \times N_{SW} \right) + \sum \left( C_{Ln} \times f_{On} \right) \times V^2_{CC} \tag{11}
\]

In the above equation, each output load capacitance \( C_{Ln} \) and frequency of switching \( f_{On} \) are directly related to the number of operations in an algorithm. As described in Section VII, the architecture of CLEF switches off the processing engines if they are not required based on the contextual information, resulting in lower number of operations. Consequently, the component of dynamic power due to output load capacitance at each node and the corresponding switching frequencies is also eliminated when a processing engine is switched off in CLEF. This could significantly reduce the overall power consumption in the hardware implementation of CLEF. Although actual implementation of CLEF on hardware platforms such as FPGAs or ASICs will incur additional computational costs, the significant computational savings by CLEF presented in this paper contribute to the overall efficiency of the actual implementation of CLEF on hardware.

E. Illustrating Other Contextual Cues

In addition to the formulations in Section V that help determine the effective number of scan bands in CLEF, other contextual information such as timing conditions were used to configure CLEF, and implemented to show the effectiveness of CLEF. Fig. 11 (a) shows the lane estimation results when a light sensor is used to determine the nature of lighting. The magenta and orange boxes on the top left corners in the images in Fig. 11 (a) show the number of scan bands that are being processed: \( \blacksquare \) for active scan band, and \( \square \) for passive band. We can see that during daytime with low traffic, the road scene in the far-view of the ego-vehicle is visible and therefore, all the 16 scan bands are active to capture lane information from as far from ego-vehicle as possible. However during nighttime (Fig. 11(b)), the first four bands of LASeR are switched on to determine the lanes in the near view of the ego-vehicle, because this region is more critical during night time for lane keeping. The framework can configure LASeR to behave in a similar manner for lane changing and lane keeping applications.

In Fig. 11 (b), we show another case of adaptability enabled by CLEF to manage computational cost based on the need for accuracy using three frames \( t_1 \) to \( t_3 \) from a video sequence. There are straight lanes in \( t_1 \) but a leading vehicle is obstruct-
ing the ego-vehicle, followed by a curving lane in $t_2$, and finally a straight lane in $t_3$. The proposed framework selects the first 4 scan bands in $t_1$ because of the obstruction due to the leading vehicle. When the lane starts curving, CLEF detects an increasing curvature value as discussed in Section V-C. Therefore, all the scan bands are switched on so that it gets a good accuracy for curved lane fitting (given that there is no vehicle within the proximity of the ego-vehicle). After passing the curvature, the lane straightens again in $t_3$ with low traffic. Here we show the case when CLEF switches off alternate scan bands (as shown in $t_3$ in Fig. 11 (b)) by setting the lane estimation process to operate in medium level of accuracy for straight lanes.

The above examples show some of the scenarios in which CLEF determines the contextual information for the lane detection process. We demonstrated how CLEF can enable the active safety system (LASeR in this case) to scale and adapt itself to get the required lane information with a high level of accuracy for the context, and lead to computational efficiency (and power savings) also. Table VI illustrates the number of scan bands that can be made active for some typical contexts (a maximum of 8 scan bands are assumed in this illustration). In addition to the different configurations given in Table VI, it is to be noted that the number of active scan bands can be varied dynamically using the distance of the leading vehicle and ego-vehicle dynamics, as shown previously.

X. Concluding Remarks & Future Directions

In this paper, we have proposed a framework called CLEF that determines the contextual information about the ego-vehicle and its surroundings, which is then used to scale lane detection algorithms. CLEF was shown to synergistically fuse the video data with other sensing modalities in the vehicle, the different conditions about the ego-vehicle, the road, vehicle in front of the ego-vehicle, and the surrounding environment, so that lanes are detected based on the context. CLEF was demonstrated and evaluated using real-world driving data, and it was established that it improves the computational efficiency significantly without compromising on the accuracy of the lane detection. The detailed analysis presented in this paper shows that context-awareness can contribute positively to computational efficiency, and hence energy management of vision-based active safety systems in resource constrained embedded electronic subsystems in automobiles.

The work in this paper is one of the earliest contributions in context-aware processing for embedded automotive systems, and is limited to specific contexts related to the ego-vehicle state and some of its surroundings. Given that the real-world driving scenarios can vary significantly and there is a genuine need for energy-efficient in-vehicle electronic subsystems, there is a large window of opportunity to develop techniques that can contextualize ADAS tasks such as lane detection for varying real-world driving conditions. We have presented a detailed architectural description of CLEF. An actual implementation of CLEF on hardware platform could incur additional computation costs. However, the significant computational savings in CLEF will positively contribute in reducing the hardware complexity and power consumption of the hardware implementation on platforms such as FPGAs and ASICs.

ACKNOWLEDGMENTS

We would like to first thank the reviewers for their insightful comments and suggestions. We would also like to thank UC Discovery Program and our sponsors, particularly Toyota Collaborative Safety Research Center. We also thank our colleagues from Laboratory for Intelligent and Safe Automobiles (LISA) for their support.

APPENDIX

The different datasets that are used for evaluating CLEF will be made available for future comparisons. Some lane estimation datasets are already made available at http://cvrr.ucsd.edu.

REFERENCES

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2406171, IEEE Transactions on Circuits and Systems for Video Technology