

Application of Inverse Perspective Mapping for Advanced Driver Assistance Systems in Automotive Embedded Systems

Vighnesh N. T.¹, Rachana Anil², Rohith Kumar D³, Sanjana Sharvana⁴, Rajeshwari Hegde⁵, B S Nagabhushana⁶

^{1,2,3,4,5}Department of Telecommunication Engineering, BMS College of Engineering, India

⁶Department of Electronics & Communication Engineering, BMS College of Engineering, India

Article Info

Article history:

Received Aug 05, 2017

Revised Oct 06, 2017

Accepted Oct 20, 2017

Keywords:

Blind Spot Detection (BSD)
Hardware Abstraction Layer (HAL)

Inverse Perspective Mapping (IPM)

Kahunen-Loeve Transform (KLT)

Lane Departure Warning System (LDWS)

ABSTRACT

In the recent times vehicle manufactures and automotive suppliers are progressing towards building vision based subsystems for provisioning driver assistance while targeting the automotive safety critical needs. While the acquired images constitute the fundamental input for any vision based system, transforms on images become essential to derive and gain insight into certain specific features. These derived features are used and reused at multiple places for varied automotive applications. This situation warrants a scalable and flexible image processing platform for a class of automotive applications. An attempt is made in this Research work to propose architecture that, specially, includes a layer of image transformations and to implement a prototype image processing platform. Inverse Perspective Mapping (IPM), a widely used class of transforms is emphasized in the present architecture alongside other nominal transforms. Lane departure warning system is implemented on this platform for the purpose of illustration and to analyze the effectiveness of the proposed architecture.

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Vighnesh N. T.,
Department of Telecommunication Engineering,
BMS College of Engineering,
PO Box no 1908, Bull Temple Road, Bangalore, India.
Email: vighnesh.nt@gmail.com

1. INTRODUCTION

Automotive subsystems oriented towards safety are broadly classified as passive (secondary) safety systems and active (primary) safety systems. An active safety system acts prior to an eventuality whereas a passive safety system acts alongside an eventuality. Having been backed by mature technologies, passive safety systems are commoditized whereas active safety systems are, continuously, researched and hence are evolving partly owing to exploration of newer techniques and partly owing to the innovative automotive requirements. In terms of their sphere of influence, active safety systems can be broadly categorized as internal and external. Internal active safety systems are internal to the vehicle and are distributed in nature wherein safety mechanisms are a part of ECUs like the Powertrain control module, Electronic Throttle Control (ETC), Anti-lock braking systems and others. On the other hand, subsystems like Lane Departure Warning Systems (LDWS) [1], pedestrian detection systems, accident avoidance systems, drowsiness detection systems and such similar systems [1] constitute external active safety systems. Although these systems existed for over two decades, state of the art technologies have, significantly enhanced these systems to augment their utility and effectiveness with respect to automotive safety critical requirements. An attempt is made in this paper to present our research investigations with respect to building external active safety systems. Section 2 describes state-of-the-art technologies potential of supporting the development of external

safety systems. Section 3 deals with theoretical framework for Inverse Perspective Mapping (IPM). A generic architecture to build active safety critical systems that houses these technologies is conceptualized and presented in Section 4 of this paper. Development platforms for a class of applications can be derived from this architecture. Section 5 deals with implementation. The analysis for real time performance is presented in section 6. The approach of platform-based development is well practiced in the automotive industry. Development of LDWS using the proposed architecture is presented as a case study in Section 5 of this paper. Conclusions are drawn and presented in Section 7 of this paper.

2. ADVANCED TECHNOLOGIES FOR EXTERNAL SAFETY SYSTEMS

Image processing has been the widely accepted technology for building external safety systems for automotive. Filtering, restoration, feature extraction, data compression and such requirements are well addressed by the Research community and as such, versatile solutions are available and employed to build applications [3]. Image processing for building applications based on Augmented Reality (AR) and Virtual Reality (VR) paradigms are comparatively new and solutions are being built. Although AR and VR are attributed to gaming and such other applications, mission critical applications based on AR and VR have also been identified and automotive is no exception. Such applications and also few conventional applications pose challenges in terms of memory and computational power that obviate their ease of deployment, especially for real time systems. High end microcontrollers, Field Programmable Gate Arrays (FPGAs), IP cores and integrations of these entities into a hardware platform are expected to alleviate the challenges indicated above with respect to computational power. Hardware architectures wherein microcontrollers work in tandem with FPGAs are witnessed to be effective in building image processing hardware platforms. Obviously, in such architectures, time critical tasks are scheduled on FPGAs and algorithmic tasks are scheduled on the microcontrollers. However, more realistically, with the present day advancements, value additions brought by FPGAs for image processing is quite nominal.

Addressing real time requirements at the hardware layer shall, by itself, not suffice for the overall throughput betterment since the latency is also contributed due to system software at varied levels. Image processing technology has identified this lacuna and addressed this challenge through software libraries of which OpenCV is a representative one. It is the experience of the authors that use of OpenCV reduces the latency by over 75% approximately while being worked on a multi-tasking operating system. Further, since OpenCV is written in C++, the ease of integration with applications and/or middleware developed using Object Oriented philosophy is quite promising.

While OpenCV is quite versatile within its own sphere of influence, it does not address additional techniques whose implementations would supplement basic image processing primitives supported by OpenCV. These additional techniques, primarily, are 2-dimensional extensions of transform based techniques which have proved well with its 1-dimensional counterpart in signal processing. Examples of such transforms leaving aside the conventional Fourier Transforms, Fast Fourier Transforms, Discrete Cosine Transform include one or more of the following and such other transforms:

- a. Inverse Perspective Mapping
- b. Discrete Karhunen-Loeve transform (KLT)
- c. Walsh transform
- d. Hadamard transform
- e. Walsh-Hadamard transform

3. THE ORITICAL FRAMEWORK FOR INVERSE PERSPECTIVE MAPPING

Understanding of the image formation process and image analysis involves transformation and/or inverse transformation. This technique of transformation is extended to map images across coordinate systems of which the duo, perspective transformation and inverse perspective transformation is found to be highly useful. However, transformations over coordinate systems get complicated owing change in dimensionality. Specifically, translation from 3-dimensional to 2-dimensional and viceversa is the fundamental requirement in the context of perspective transformation. Perspective transformation takes an image of a point or a set of points in the 3 dimensional world and these points are mapped to the imaging plane which is a 2 dimensional one. The inverse perspective transformation just does the reverse process where the uniqueness of the transformation needs a detailed analytics. The analytics of IPM is briefed in the following paragraphs. Figure 1 shows the imaging geometry wherein the world coordinate system is aligned with the camera coordinate system.

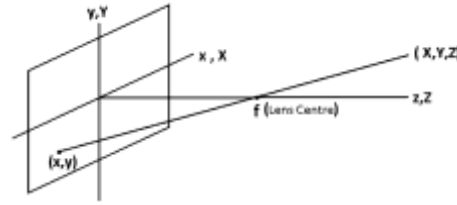


Figure 1. Alignment of world coordinate system with the camera coordinate system in the imaging geometry

A 3D point (X, Y, Z) creates an image (x, y) on the image plane. Further, if λ is the focal length of the camera lens, the lens center is $(0, 0, \lambda)$. To determine a relation between the point (X, Y, Z) and the corresponding image point which is (x, y) , as a first step, the Cartesian coordinate system is converted to a homogeneous coordinate system. Analytical treatment of perspective projection geometry needs the mathematical abstraction of infinity which is provisioned by the homogeneous coordinate system. The homogeneous coordinates for the 3D point (X, Y, Z) is given by (kX, kY, kZ, k) where k is an arbitrary nonzero constant. The corresponding homogeneous coordinates for the image point (x, y) is obtained by defining a perspective transformation matrix P defined as:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (-\frac{1}{\lambda}) & 1 \end{bmatrix} \quad (1)$$

Using this transformation, the homogeneous coordinates of the image point is obtained by using the homogeneous coordinates of the 3D point as follows:

$$c_h = Pw_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (-\frac{1}{\lambda}) & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ -k(\frac{Z}{\lambda}) + k \end{bmatrix} \quad (2)$$

The homogeneous coordinates of the image point is converted into cartesian coordinates by dividing each of the elements of the column matrix ch by its fourth element. With this, the cartesian coordinates of the image point is derived as

$$c = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \lambda X / (\lambda - Z) \\ \lambda Y / (\lambda - Z) \\ \lambda Z / (\lambda - Z) \end{bmatrix} \quad (3)$$

From a practical view point, it is essential to be able to determine the world coordinate for the 3D point after having known the corresponding image point. This task is accomplished by inverse perspective transformation, as briefed below.

The inverse transformation of the image point to the 3D world point is given by

$$w_h = P^{-1} c_h \quad (4)$$

Where P^{-1} is the inverse perspective transformation matrix is given as

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (\frac{1}{\lambda}) & 1 \end{bmatrix} \quad (5)$$

Consider a point (x_0, y_0) to be a point on the image plane. For any arbitrary value of k , $(kx_0, ky_0, 0, k)$ would be the homogeneous coordinates of the point (x_0, y_0) . The homogeneous coordinates of the 3D world point is given by

$$w_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \quad (6)$$

Using this homogeneous coordinates for the 3D world point, one can find out its Cartesian coordinate counterpart as

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} \quad (7)$$

However, the issue here is that any point on the 3D space would map onto the image plane with its z coordinate being zero.

Consider the world coordinate system and image coordinate system aligned to each other as shown in the Figure 2. Consider the image point $(x_0, y_0, 0)$. Consider a straight line from this point passing through the optic center of the lens. It is obvious that any point on this line maps onto the point $(x_0, y_0, 0)$. Thus this point has multiple mapping with the 3D world. Obviously, inverse perspective transformation applied on the point $(x_0, y_0, 0)$ will not lead to an acceptable world coordinates for the 3D world point.

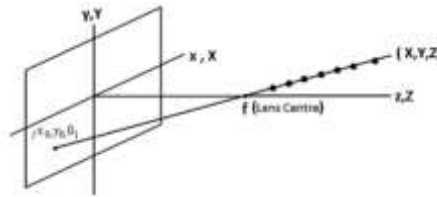


Figure 2. Image point $(x_0, y_0, 0)$ on the 3D space mapped onto the image plane

In order to meet the engineering requirement that we need to be able to determine the world coordinate points given the corresponding image point, uniquely, we relax the formalism of the imaging geometry by assuming the image point to have a fictitious z -coordinate instead of zero. Thus, the image point in terms of Cartesian coordinates would be (x_0, y_0, z) which translates to its homogeneous coordinates as (kx_0, ky_0, kz, k) where both z and k are arbitrary numbers. Proceeding further, taking inverse perspective transformation on this modified image point we get the homogeneous coordinates of the 3D world point as

$$w_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix} = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \left(\frac{kz}{\lambda}\right) + k \end{bmatrix} \quad (8)$$

Further, converting w_h into Cartesian coordinate system, we get

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \lambda x_0 / (\lambda + z) \\ \lambda y_0 / (\lambda + z) \\ \lambda z / (\lambda + z) \end{bmatrix} \quad (9)$$

So,

$$X = \lambda x_0 / (\lambda + z)$$

$$Y = \lambda y_0 / (\lambda + z)$$

$$Z = \lambda z / (\lambda + z)$$

Using the above three equations, we can eliminate the fictitious variable z . This process will give us

$$X = (x_0 / \lambda) (\lambda - Z) \text{ and}$$

$$Y = (y_0 / \lambda) (\lambda - Z)$$

This work around has led to a situation by which it is possible to map an image point onto a 3D point provided we have an idea about its z -coordinate (Z). Techniques by which this information is made available differentiate the mapping algorithms which have been devised. Few variants of the technique are explained in the following sections.

4. PROPOSED ARCHITECTURE

The motivation for devising an architecture for image processing applications for automotive is a three-fold one. The foremost motivation spurs out of a generic need for architecture to provision the design and implementation of a well-structured solution along with a gamut of benefits that include standardization, usability, and comprehensibility. Although one gets tempted to include factors like flexibility, scalability, maintainability, quality, performance, they are, in a way; do not fall within the purview of architecture. The second motivation, having said that the architecture leads to platform software, is owing to the contextual need of automotive industry. All subdomains of automotive are endowed with platform software upon which the application gets developed. This platform software, typically, vests with Tier-1 suppliers who would use, reuse and expand platform software meeting the requirements of multiple vehicle programs across vehicle OEMs. Development of vehicular infotainment is a grand success story to demonstrate how platform software could be grown to meet the state-of-the-art requirements, just in time. Vehicle infotainment has, in fact, has grown from its stage of infancy with just the AF/FM/RDS functionality to the present day infotainment which fulfills multitude of requirements that include USB, Bluetooth, GPRS, phone, satellite radio and others. The third and, probably, the recent motivation for architecture has spurred out of multitude of applications oriented towards active safety and which are built on the image-processing paradigm. Specially, applications catering to the requirements of connected vehicles, autonomous driving and others are far more safety critical and involve exhaustive image processing compared to nominal active safety mechanisms like LDWS, pedestrian detection, parking assist, collision detection and warning and others[4]. Such multitude of applications, in the near future, shall redefine platform based development in automotive by which there shall be a dire need for architecture from which platform software shall have to be derived. As indicated above, an attempt is made in this research work to conceptualize such architecture for the development of automotive applications that, specially, involve high levels of image processing. Figure 3 depicts the proposed architecture for automotive image processing applications, which can be adapted for building multitude of applications.

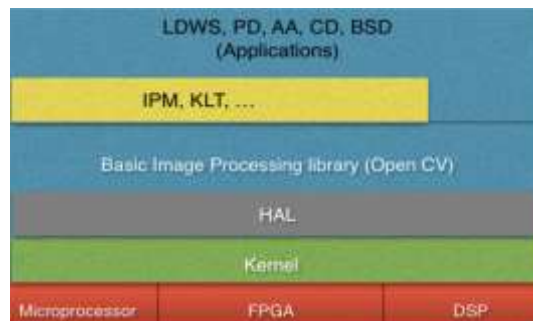


Figure 3. Proposed architecture

The first layer is the hardware layer, which is built around a microprocessor, an FPGA, and/or a DSP. The choice of one or more of these hardware options depends on the expected system performance, primarily, in terms of the computational speed. The hardware, typically, could involve a combination of all the three options to, mostly, achieve real time implementation. The second layer is the kernel, which shall, in the context of automotive applications need be a real time kernel. In the present implementation, Linux kernel is employed. Employing Linux kernel obviates the need of a dedicated Hardware Abstraction Layer (HAL) whenever the host computer is employed thus easing out the developmental need. However, a real life implementation warrants a dedicated hardware and hence the HAL. The HAL, with the necessary OS support, abstracts the underlying hardware from the software layers above it. The Basic Image Processing Library provides primitives for image processing and is capable of interfacing with the hardware through the HAL. For the proposed architecture and the representative application implementation in the present Research work, OpenCV is employed as indicated in the Figure. However for implementing more complex domain tasks, OpenCV may not suffice. However, the OpenCV software is feature rich and highly optimized for embedded applications and hence is potential of being retained as an architectural block. Introducing a dedicated layer titled “Image Processing Algorithms” as shown in the Figure 1 caters the complex image processing tasks. This layer of software implements more domain oriented, large-grain image processing functionality by using the support provisioned by OpenCV. In the proposed architecture, implementation of Inverse Perspective Mapping (IPM), Kahunen-Loeve Transform (KLT), Hadamard and such similar transform based techniques constitute this newly built layer. Upon this layer lies a host of applications catering to the varied needs of automotive. Also, a clearly differentiated functionality of the application might need a provision to use OpenCV. This facility is also provisioned as indicated in Figure 1 by which the application software can call OpenCV routines, directly. Specifically, in the present research, IPM is developed afresh and employed along with other transforms provisioned by OpenCV[5[7]] to build a novel Lane Departure Warning System[5] as a case study to demonstrate the versatility of the proposed architecture.

5. IMPLEMENTATION

An implementation of the proposed architecture has been completed on a host computer, which runs Ubuntu 12.04 operating system. The OpenCV library is used as the basic image processing software for implementation. The algorithm, which we used to implement IPM, was 4-point correspondence. This algorithm helped reduce the computation time by computing complex matrices without actually computing the entire matrix, but performing many to one mapping of the pixels in the region of interest obtained by the 4-point correspondence[7]. This algorithm is executed on a pre-recorded dash board camera video and also on live feed input from the camera[8]. The real time analysis of the complete processing of the images is prioritized over the spatial analysis of the output image in the present study and is detailed in the following section. Figure 4 shows output of 4-point correspondence.

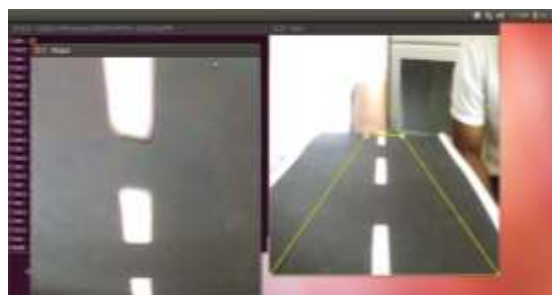


Figure 4. Output of 4-point correspondence

6. ANALYSIS FOR REAL TIME PERFORMANCE

In view of the time criticality of the active safety systems coupled with the computationally intensive image processing tasks, it is essential to conduct a formal analysis of the system performance for having achieved or not of the real time performance. Non-fulfillment of the real time performance, fundamentally, defeat the purpose of the application itself. As such, a formal analysis of the software

execution was conducted for the LDWS application built on the architecture briefed in the previous section. The highlights of this analysis are explained in the following paragraphs:

The camera employed for the image acquisition works at 30fps (Frame Per Second), which means that the camera is rendering a frame every 30ms and hence the maximum time available for achieving real time performance is to ensure that the computation time falls within the time duration (approx.30ms) available. In view of the safety criticality and also the time required for the control action, as the case might be, maximum allowed time for computation is fixed (by design) to be 20ms. With the present implementation, the computation time, on an average, was found to be 15-20ms, which is quite in order for achieving real time performance. However, this metric is only a partial fulfillment for claiming real time performance since this metric does not, anyway, address the speed of the vehicle and the time available for corrective action by the driver. As such, as a further extension to the above analysis, computation time per frame was evaluated for a 15 minute run of a real life capture of the video for the LDWS [9] system and the results are as shown in Figure 5 and the graphical representation as in Figure 7.



Figure 5. Output of analysis in real time

A second analysis was conducted on the processed video signal by which it was possible to detect the lost frame the graph for which is shown below. All those frames for which the computation time gets reported as zero were found to be the frames, which have not been processed but are lost. As expected the number of lost frames increased with the increase in speed of the vehicle. Ideally, it is essential to keep the number of lost frames to be zero for all vehicle speeds. However, in the present implementation, the number of lost frames at the vehicle speed of 90kmph was of the order of 5-10 frames for every 15minutes of vehicle run.

We had to broadly classify our analysis into three major categories

- a. Variable speed (emulating the real life situation)
- b. Rapid increase in speed (to assess the impact of rate of change of speed)
- c. Rapid decrease in speed (to assess the impact of rate of change of speed)

The parameters employed to base our analysis are as follows:

- a. Capture Rate.
- b. Time taken for the computation on the host system.
- c. Loss of frames at different speeds.

Figure 6 clearly shows that the capture rate was constant throughout since the functionality of the camera is, completely, independent. However, the computation time interval from accepting a frame till displaying the corresponding output was ranging between 10-20 milliseconds. This concluded that for vehicle speed up to around 80kmph, no frame would be lost from the processing loop. Since at 80kmph the computation time just grazes the 20ms (the design limit), speeds beyond just 80kmph would result in lost frames. Infact, this graph is extracted from the output of a real road test (of 10minutes duration) which is depicted in Figure 7. Differences in computation times throughout the run are again a clear depiction of the dynamics of the code, which result out of varying execution path of the application software.

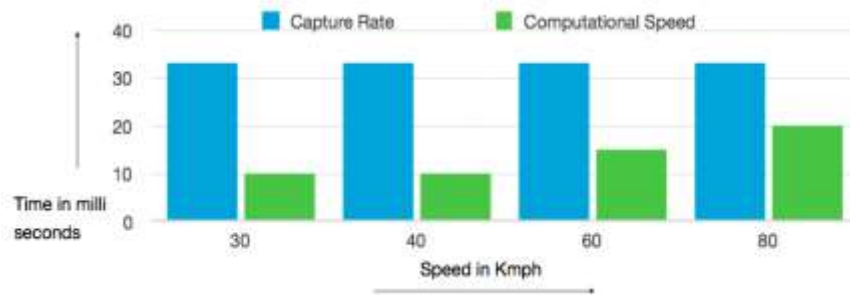


Figure 6. Capture rate as seen for speed vs. Time

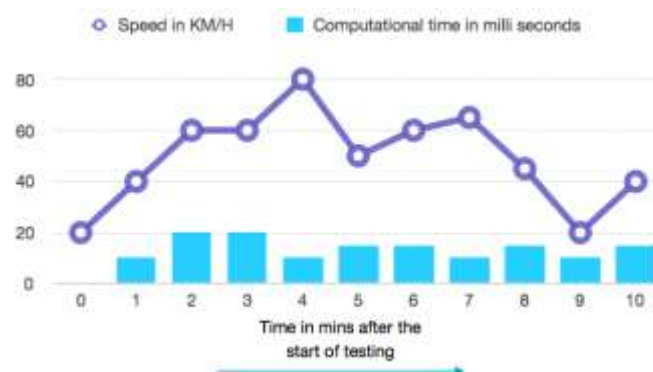


Figure 7. Computational Time and system response for varying speeds for a test time of 10 minutes

Further analysis was carried out to assess the impact of rate of change of speed on the processing dynamics. Using relevant data points of the test run, briefed above, a simulation test bench is created. Variation in computation time for an instantaneous increase in speed as well as for an instantaneous decrease in speed is captured as presented in Figure 8 and Figure 9, respectively.

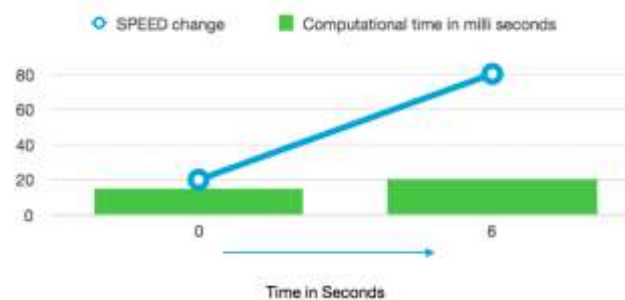


Figure 8. Computational time (in milliseconds) calculated for rapid increase in speed in Km/h.

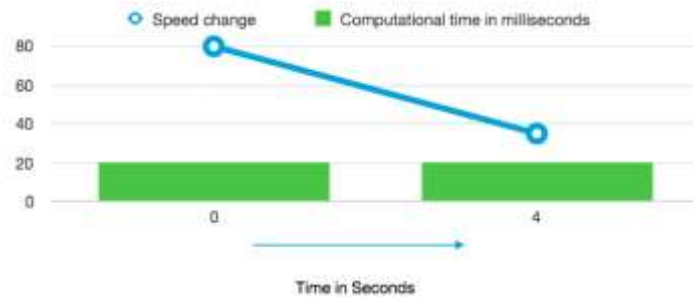


Figure 9. Computational time (in milliseconds) calculated for rapid decrease in speed in Km/h

Analysis with case studies as above become important since catastrophes result in automotive owing to sudden increase or sudden decrease in vehicle speeds. Although, the computation times for our case study was found to be within the design limit, with different speed profiles the situation might differ. Enhancement to the simulation setup indicated above shall be the future work of the authors. These enhancements shall also resolve, specifically, the bottleneck associated with the inability to capture missing frames[10].

7. CONCLUSION

Future vehicle programmes are expected to include far more improvisations and sophistications predominantly in terms of autonomy. This situation would warrant the deployment of specialized subsystems targeting safety. Additionally, safety shall be all pervasive in terms of mandatory add-ons for the prevalent ECUs. Since these subsystems, defacto, employ image processing technology, a generic image processing platform architecture is presented in this paper. Using this architecture, software development platforms can be built upon which a class of applications can be built. Lane Departure Warning System is developed on one such custom built platform as a case study. The authors envisage significant add-ons to the proposed architecture in terms of video data analytics and also a wide spread utility of the architecture for building development platforms as an enabler for the development of safety critical automotive applications in the future.

REFERENCES

- [1] Ioana Maria Chira, Ancuta Chibulcutean, Radu Gabriel Danescu, "Real time detection of road marking for driving assistance applications", *2010 International Conference on Computer Engineering and Systems (ICCES)*, pp 158-163, 2010.
- [2] Darko Juric, "On- Road Night Time Vehicle Light Detection and tracking ods Overview", Available at: https://www.fer.unizg.hr/_download/repository/KDI_Darko_Juric.pdf
- [3] Vikram Appia, Hemant Hariyani, Shiju Sivasankaran, Stanley Liu, Kedar Chitnis, Martin Mueller, Umit Batur, Gaurav Agarwal, "Surround view Camera for ADAS on TI's TDAX SoCs" available at: <http://www.ti.com/lit/wp/spr270a/spr270a.pdf>
- [4] Mohamed Aly, "Real time detection of lane markers in urban streets", *2008 IEEE Intelligent Vehicle Symposium*, pp 7-12, 2008.
- [5] S.Tuohy, D.O'Cualain, E. Jones, M.Glavin "Distance determination of an Automobile Environment using Inverse Perspective Mapping in OpenCV", *IET Irish Signals and Systems Conference (ISSC 2010)*, pp.1-3, June 2010.
- [6] Jun Wang, Tao Mei, Bin Kong, Hu Wei, "An approach of lane detection based on inverse perspective mapping", *17th International IEEE conference on Intelligent Transport Systems*, pp-35-38, 2014
- [7] Miguel Oliveira, Vitor Santos, Angel D. Sappa, "Multi modal Inverse Perspective Mapping" available at: http://www.cvc.uab.es/people/asappa/publications/J_Elsevier_IF_Vol_24_July_2015_pp_108-121.pdf
- [8] X. Clady, F. Collange, F. Jurie and P. Martinet, "Tracking with a pan tilt zoom camera for ACC", available at: <http://www.isir.upmc.fr/files/2001ACTI598.pdf>
- [9] Massimo Bertozzi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection", *IEEE transactions on Image Processing*, Vol7, issue 1, pp 62-81, 1998
- [10] Dongwook Seo, Khangyun Jo, "Inverse Perspective Mapping based Road Curvature Estimation", *2014 IEEE/SICE International Symposium on System Integration*, pp 480-483, 2015.

BIOGRAPHIES OF AUTHORS

Vighnesh N.T received his Bachelor of Engineering in Telecommunication Engineering from BMS College of Engineering, Bangalore. His interests include Automotive Embedded Systems, Image processing and Computing networks and systems mainly focusing on Automotive domain. He is currently serving as Developer for Tata Consultancy Services, Bangalore, India and is serving Jaguar Land Rover account, developing Human Machine Interface(HMI) for the infotainment system.



Rachana Anil received her Bachelor of Engineering in Telecommunication Engineering from BMS College of Engineering, Bangalore. Her interests include Embedded Systems and Image Processing. She is currently serving as a Developer in Robert Bosch Engineering and Business Solution, developing Run Time Environment(RTE) for controllers used in Mobile Hydraulics.



Sanjana Sharavana received her Bachelor of Engineering in Telecommunication Engineering from BMS College of Engineering, Bangalore. Her interests include Embedded Systems and is currently serving as a Service Engineer in Microland as Application Packager and Tester.



Rohith Kumar. D received his Bachelor of Engineering in Telecommunication Engineering from BMS College of Engineering, Bangalore. His interests include Automotive Embedded Systems. He has worked as an intern in GipzonICs techno labs in Embedded Systems for 5 months and is currently working as an Associate Software Engineer in Accenture, Bangalore, India, developing and maintaining the SWIFT application for payments of CYBG.



Rajeshwari Hegde received her Bachelor of Engineering in Electronics and Communication Engineering from National Institute of Engineering, Mysore, Master of Engineering in Electronics from BMS College of Engineering, Bangalore and PhD from Bangalore University. She is currently an Associate Professor & Head, Department of Telecommunication Engineering, BMS College of Engineering, Bangalore, India.



B.S.Nagabhushana received his Bachelor of Engineering in Electronics from Bangalore University, M.Tech. in Industrial Electronics from Univ. of Mysore and Ph.D. from Indian Institute of Science (IISc), Bangalore. He is presently associated with the Department of Electronics and Communication Engineering of BMS College of Engineering, Bangalore, India. His Research interests include embedded systems with focus on automotive electronics.