A Low-Cost, Wireless, Multi-Modal Sensing Network for Driving Behavior Modeling and Early Collision Warning

Hamad Ahmed, *Member, IEEE,* and Muhammad Tahir, *Member, IEEE*

***Abstract*—Traffic accidents have escalated to an alarming level in the last decade or so, mainly due to inattentive and rash driving behavior. Recent research efforts have resulted in systems for automatic detection of abnormal driving, that can warn the driver as well as notify traffic authority so that the person can be taken off the roads. However, these systems lack large-scale deployment as they are expensive and suffer from false positives. Driving is a complex multi-dimensional interaction of the driver with his environment. Therefore, to effectively capture the driving behavior, information of surrounding vehicles should be considered as well, which is also missing in existing systems. In this paper, we have proposed a multi-modal vehicular sensor network which is low-cost and uses both the vehicle's motion profile and surrounding traffic information for behavior modeling. Heading (pose) and speed of the vehicle are determined from an IMU and vehicle CAN bus respectively, whereas the surrounding traffic is sensed and tracked by a LIDAR. To process and fuse the data of various sensors, algorithms of low-complexity are proposed which are able to run on low-cost microprocessors, thus removing the need of an on-vehicle computer which further reduces the cost of the system. We also provide early warnings of risk of collisions based on the motion of surrounding traffic and the driver's intent inferred from his eye movements which are tracked by a camera. The hardware architecture of the system is also discussed. Experimental results on a test vehicle in real traffic scenario show robustness to false positives.**

***Index Terms*—Driver Safety, Driving Behavior, Driving Events, Aggressive Driving, IMU, Kalman Filter.**

I. Introduction

STEADY development of motor vehicles during the 20th century has greatly influenced human life, providing greater mobility for people and promoting socio-economic development. However the alarming increase in traffic congestion has not been matched with adequate availability of safety measures, making motor vehicle accidents a major cause of death. According to National Highway Traffic Safety Administration (NHTSA), 32,675 people died in U.S. due to road accidents in 2014. Roughly one-third deaths occurred due to drunk driving and an additional 12.5% occurred due to inattentive driving [1]. Furthermore, the first six months of 2015 have already seen an alarming increase of 8.1% in this fatality rate as compared to 2014. In order to address this startling situation and increase road safety, several methods have been used over the years including traffic monitoring cameras and strict penalties on breaking traffic laws. These have somewhat forced the drivers to drive within safety limits but the fatality rate still continues to rise. There is a need to catch drunk drivers, alert inattentive drivers and force rash drivers to drive safely.

It has been shown that drivers tend to drive attentively and are unlikely to perform rash maneuvers when they know that they are being monitored [2]. For example, drivers are less likely to take a red light when they know that cameras are watching as compared to when there are no cameras. 'How am I driving' bumper stickers and call service are utilized for this same purpose. These stickers are installed on trucks, buses and other commercial vehicles and the companies claim that drivers are less likely to engage in unsafe maneuvers when they are being monitored due to these stickers [3]. However as pointed out in [4], it is not practical for someone to actually call the hotline and complain about somemone's driving as using cell phones are not allowed while driving and 76% of workers drive alone to work which means that there is no passenger available to make the call.

To overcome these issues, researchers have developed systems that can be installed on vehicles to automatically monitor driving behavior. These systems serve to alert a driver if they sense inattentive driving or provide feedback to the driver in case of rash driving. The functionality can also be extended to notify traffic authorities if the driver continues rash driving or in case of drunk driving. With the help of certain sensors, speed and heading (pose) of the car are determined and modeling or learning based techniques are applied to determine 'Driving Events' or maneuvers. These events include accelerating, decelerating, lane changes and turns. Based on the nature and frequency of these driving events, driving behavior of the driver is determined.

The measure of success of such an automated system depends upon two factors: the cost-effectiveness of the system and its false positive rate i.e. how often does the system perceive rash driving while actually it is not the case. Regarding the first factor, a high cost of the system will serve as an inhibiting factor towards its widespread and global usage. To decrease the overall cost, it is imperative that only low cost sensors are used in the system, and the algorithms used to determine driving behavior from low-level sensor data are of low complexity so that they can be run on low cost microprocessors and do not require an on-vehicle computer. Existing systems use sensors which are either expensive (high precision GPS, RADAR, LIDAR, camera) or are mobile phone based, which leaves the system to the disposition of the driver. The sensor fusion algorithms are also computationally expensive and are run on the processors of mobile phones or offline on a computer using the data logged by these sensors. Thus creating algorithms which are able to run in real time

without an on-board computer is another challenge. Regrading the second factor, too many false positives will cause the driver to eventually uninstall the system. Different learning and classifying based methods are used to determine driving events, whose accuracy can be affected by a number of factors including noise, outliers or other misleading cues in the sensor data, especially when low cost IMU are used. Therefore, a more robust sensor fusion and modeling method is required which is able to reduce the false positive rate of the system to an acceptable value.

Existing systems also do not use the surrounding traffic information while determining the driving behavior which is perhaps the most influential factor on one's driving. By sensing only the motion of the ego-vehicle and detecting rash driving events, the system flags the driver without taking into account the influence that the surrounding vehicles may have had on the driver's intent to perform that rash maneuver. For example, sudden abnormal behavior by the leading car could have caused the driver to perform an emergency evasive maneuver. Similarly, a driver accelerating rapidly on an empty road will also be termed as driving rashly whereas he is not a threat to anyone. Trying to sense the surrounding traffic information is a very challenging task as expensive equipment like a radar or camera is required which also demands an on-board computer for signal processing thus the overall cost and deploy-ability of the system is severely affected.

This paper proposes a low-cost wireless multi-modal sensor network which can be installed on any vehicle and takes into account both the maneuvers performed by the ego-vehicle as well as the motion of the surrounding vehicles to determine the driving behavior. The proposed system uses only low-cost sensors i.e. low-cost MEMS-IMU, speed of the vehicle from CAN bus and low-cost LIDARs to sense the surrounding traffic. Choosing independent sensors makes the system deploy-able as smart-phones or IMU sensors data on the CAN bus are not required. Heading of the vehicle is determined from IMU using an adaptive Kalman filter based fusion scheme which is robust to both dynamic conditions and environmental disturbances. We improve the learning and classification based method for detecting driving events to further provide robustness against false positives. The scans from LIDARs are processed and surrounding vehicles are detected and tracked using inexpensive algorithms capable of running on low-cost microprocessors. An overall information fusion scheme which collectively evaluates the performed maneuvers and surrounding traffic to provide a meaningful behavior model is also proposed.

Since we have the motion profile of both the ego-vehicle and the surrounding vehicles, we extend the system to also provide early detection of risk of accidents. The system triggers an alarm if it senses the potential of a collision in the near future. To provide a considerable warning time in advance of the collision, a camera is installed to monitor the driver's eye movement and predict his intentions of performing certain maneuvers which can lead to an accidental situation.

Decreasing the cost of the system by using only low-cost sensors with low-complexity algorithms enables the deployment of the system on a wide scale. The rest of the papers is organized as follows: Section II reviews the scientific literature for the previous work done on driving behavior modeling. Section III presents all the algorithms of sensor fusion, processing and modeling as well as the hardware platform of the proposed system. Section IV shows results of the system installed a test vehicle in real traffic situation and section V presents the conclusion.

II. Related Work

Though the aim of our system is to detect abnormal driving behavior which has only been done in a few selected studies, the literature abounds with works that perform maneuver detection or surrounding vehicles tracking for different applications. We provide a review of the literature from 3 aspects: on the basis of sensors used and modalities sensed, on the basis of maneuvers detected and on the basis of algorithms used for detecting maneuvers.

A. Sensors and Modalities

The most crucial factor in the development of such a system is the choice of sensors as they directly impact the cost and complexity of the system. A summary of different sensors used in various studies is presented in Table I.

It is evident that the most widely used sensors are the accelerometer & gyroscope, which collectively form an IMU and data from the vehicle's CAN bus. An accelerometer alone has been used to determine the driving behavior in [9] and [10], by thresholding the longitudinal and lateral accelerometer measurements which indicate acceleration/deacceleration and turning respectively. [11] explores a number of statistical features (mean, median, standard deviation etc.) of the accelerometer data and uses a learning based approach. Detecting turning events from only lateral accelerometer data is prone to a high false positive rate due to the susceptibility of the accelerometer to noise.

An accelerometer in conjunction with the gyroscope is used to detect acceleration, braking and turning events in [13] and [4]. The latter uses the sensors inside a smartphone to develop a smartphone based application called MIROAD. [12] uses both the iPhone's camera and accelerometer to detect lane markings and detect lane drifting. [14], [18] and [19] use vehicle's speed data with the accelerometer for detecting events of interest.

The CAN bus is vastly used for obtaining different sensors' data directly from the vehicle. Speed is available on the CAN bus of all vehicles, while other parameters like throttle, steering wheel angle and braking etc. are available only on some vehicles.

Toledo *et al.* [8] and Liu *et al.* [7] fuse a GPS with IMU to achieve lane level position accuracy and then use it to detect lane changes. The fusion algorithm is quite involved and not feasible for real time implementation on micro-controllers.

As far as detecting the surrounding vehicles are concerned, radar and camera have been mostly used. However, radar is an expensive sensor whereas a camera requires an on-board computer for image processing and cannot function in dark or poor lightning conditions.

TABLE I: Sensors used in various studies

Sensors	Modalities	Used in Our System	Studies
Accelerometer	Attitude	Yes	[5] [6] [7] [8]
	Accl.\DeAccl.	Yes	[9] [10] [11] [12] [13] [4] [14] [15]
	Turning	No	[10] [12] [13] [4] [14] [15]
Gyroscope	Attitude	Yes	[6] [7] [5] [8]
	Heading\Turning	Yes	[9] [13] [4] [7] [15]
Magnetometer	Pose	Yes	[9]
CAN Bus	Speed	Yes	[16] [17] [7] [18] [19] [5] [8] [20]
	Throttle	No	[16] [19]
	Accl.\DeAccl.	No	[18]
	Steering	No	[19] [5] [20]
	Brake	No	[19] [5]
GPS	Position	No	[16] [7] [19] [5] [8]
	Speed	No	[14]
LIDAR	S.Vehicles Positions	Yes	
	S.Vehicles Speed	Yes	
RADAR	S.Vehicle Positions	No	[21] [20]
	S.Vehicle Speeds	No	[21]
Camera	E.Vehicle Position	No	[12] [17] [5] [20]
	S.Vehicle Position	No	[5]
	Eye\Head Tracking	Yes	[20]

B. Maneuvers Detected

Since different studies have different scopes, the maneuvers or events focused upon in them are also application dependent. There are only a handful of driving maneuvers that are perform while driving ,which include accelerating, braking, changing lanes and making a turn. A brief overview of detected maneuvers is given in Table II.

The most widely used events for classifying driving behavior are acceleration and deceleration because they are both easy to determine and give us a direct measure of the rashness of the driver. Lane changing and turns have also been detected in quite a few studies, however only [4] goes on to differentiate them into safe or dangerous lane change/turn. U-turns and round abouts do not need special treatment as they are formed by combining smaller events (U-turn is actually two consecutive turns).

C. Algorithms Used

Mostly supervised machine learning has been used in the literature to classify driving events. A classifier is trained beforehand using collected labeled examples. Mitrovic [14] trained Hidden Markov Models (HMMs) to detect turns, lane changes, curves and round abouts from accelerometer data. Two classifiers were used in [16]: Decision Trees and

TABLE II: Driving Events Detected in Various Studies

Events	Used in Our System	Studies
Acceleration	Yes	[16] [4] [12] [13] [8] [18] [10] [9]
Braking (Deceleration)	Yes	[16] [4] [12] [13] [8] [18] [10] [9]
Turns	Yes	[4] [12] [13] [14] [23] [10] [9]
Safe\Dangerous Turns	Yes	[4] [12]
Lane Change	Yes	[4] [12] [22] [8] [7] [23]
Safe\Dangerous Lane Change	Yes	[4]
Overspeeding	Yes	[4]
Curves	Yes	[14]
U-turn	Yes	[4]
Round Abouts	Yes	[14]

Linear Logistics Regression Based Classifier to detect those maneuvers which affect fuel consumption of trucks. [11] takes all possible statistical measures of the acceleration readings and applies feature ranking to select the appropriate features before training and classifying.

Mandalia *et al.* [22] train Support Vector Machines (S.V.M.) to classify driving behavior. [17] uses a Bayesian Filter on the output of SVMs for further improving lane change detection. [4] and [9] use Dynamic Time Warping algorithm to match the collected samples to a stored template of the maneuver. [18] uses Fuzzy Logic to recognize driving style. [23] presents a new hierarchical PWARX model for hierarchical classification of driving events. Luis *et al.* [12] create a function to score driving behavior depending upon the number of rash maneuver performed in unit time. The authors in these studies have not focused upon the complexity of their algorithms or guarding against false positives.

D. Vehicle Tracking by LIDAR

Vehicle tracking by a LIDAR is the classical 'obstacle detection' problem of robotics and has been studies in detail for various SLAM based applications. Though the ICP algorithm [24] exists to accurately determine shapes from a LIDAR, it is too complex to run on a low cost processor. Apart from it, two classic approaches exist for detecting objects in a laser scan: cluster based approach and model based approach.

In cluster based approach, laser points that are close together are clustered and each cluster is said to represents one object. The temporal distance between the clusters in successive scans yields motion characteristics of the object [25]–[27]. This approach has very low complexity however it is bound to fail in urban environment where complex entities are present and the target is to detect vehicles only. In model based approach, it is assumed that the geometric model of the obstacle is known and it follows some shapes e.g. line, box, circle etc. Thus objects are detected with the help of their model which improves accuracy as it filters out the outliers [28]–[30].

To refine the velocity estimates of the obstacle while tracking it, often a Kalman filter is used to fuse LIDAR

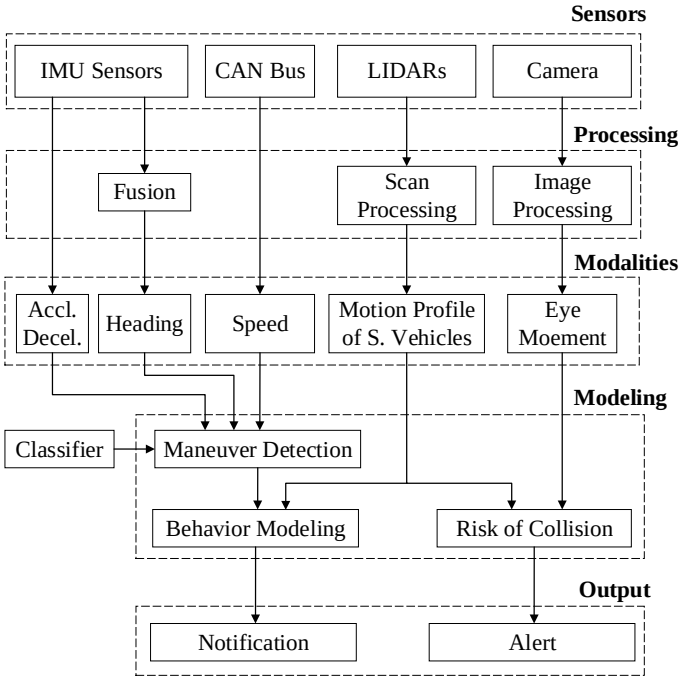


Fig. 1: System Architecture

detected trajectory with a dynamical model [31]. The velocity of obstacles measured using LIDAR is relative to the motion of the LIDAR and can either be used as it is or converted to static frame of reference by applying ego-motion compensation [32].

III. Methodology

To determine abnormal driving behavior, maneuvers are detected using the heading from IMU and speed from CAN bus whereas the surrounding vehicle information is gathered by LIDARs. The system architecture is shown in Fig 1. Each of the algorithms used in the system are detailed below.

A. Heading From IMU

The vehicle axes convention used in this paper is shown in Fig. 2. Pitch (θ), roll (δ) and heading (ϕ) angles are the rotation angles about the x, y and z axes respectively. To detect maneuvers performed by the vehicle, we require the heading angle i.e. rotation about the z-axis only as it indicates how much the vehicle is turning.

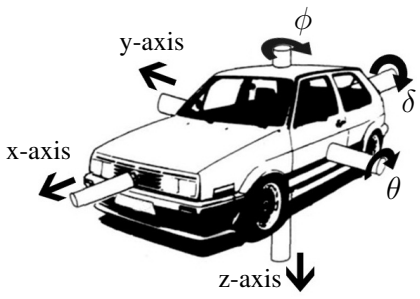


Fig. 2: Vehicle Axes & Orientation Angles

An IMU consists of a tri-axial accelerometer and a tri-axial gyroscope, and an electronic compass consists of a tri-axial magnetometer. Let the outputs of these three sensors at time t be denoted by

$$\mathbf{a}_t = \begin{bmatrix} a_{x_t} \\ a_{y_t} \\ a_{z_t} \end{bmatrix} \quad \mathbf{g}_t = \begin{bmatrix} g_{x_t} \\ g_{y_t} \\ g_{z_t} \end{bmatrix} \quad \mathbf{m}_t = \begin{bmatrix} m_{x_t} \\ m_{y_t} \\ m_{z_t} \end{bmatrix} \tag{1}$$

The accelerometer measures gravitational and linear accelerations about its axes which remain the same if the vehicle rotates about the z-axis, hence the accelerometer cannot give any information about the heading of the car. The gyroscope measures the angular rates of rotation about the x, y and z-axis. Time integration of the z-axis angular rate yields the heading angle.

$$\phi_t = \phi_{t-1} + \Delta t.g_{z_t} \tag{2}$$

However g_{z_t} is the gyroscope measurement of the rotation rate, which is related to the actual rotation rate ω_{z_t} by $g_{z_t} = \omega_{z_t} + n_{z_t}$ where n_{z_t} is the noise and bias in the gyroscope measurement. Substituting into (2), we get

$$\phi_t = \phi_{t-1} + \Delta t.\omega_{z_t} + \Delta t.n_{z_t} \tag{3}$$

We can see that the integration process will also accumulate the noise and bias in gyroscope measurements, causing the heading estimate to drift after some time. Hence the gyroscope alone is not sufficient to accurately compute the heading angle.

The magnetometer measures the magnetic field about its x, y and z-axis. If not under the influence of any external magnetic field, it measures only the Earth's magnetic field which points towards the North pole. Any change in heading of the vehicle will change the modulation of Earth's magnetic field about the x and y axes of the magnetometer, thus we can find the heading by simply taking the angle between the x and y-axis magnetometer measurements.

$$\phi_t = tan^{-1}\left(\frac{m_{y_t}}{m_{x_t}}\right) \tag{4}$$

However this is only true if the motion of vehicle is strictly restricted to the x-y plane. In reality, vehicle motion happens in xyz plane because the road structure causes changes in vehicle pitch and roll orientation as well. Earth's magnetic field is then modulated about the xyz axes and (4) no longer remains valid. To solve this, we have to determine the vehicle pitch and roll angles first, and rotate the magnetometer readings back to the horizontal plane before using (4) to compute the heading.

Pitch and roll angles can be measured by the accelerometer, only when the vehicle is static. Once the vehicle starts moving, the accelerometer measurements are influenced by the linear accelerations of the car. So we compute the pitch and roll angles at the start of the drive i.e. θ_0 ad δ_0 and use them for orientation compensation throughout the drive by assuming the car travels on a flat road with negligible changes in orientation. The pitch and roll angle are given by [33]:

$$\delta_0 = atan\left(\frac{a_{y_0}}{a_{z_0}}\right) \quad \theta_0 = atan\left(\frac{-a_{x_0}}{a_{y_0}/sin\delta_0}\right) \tag{5}$$

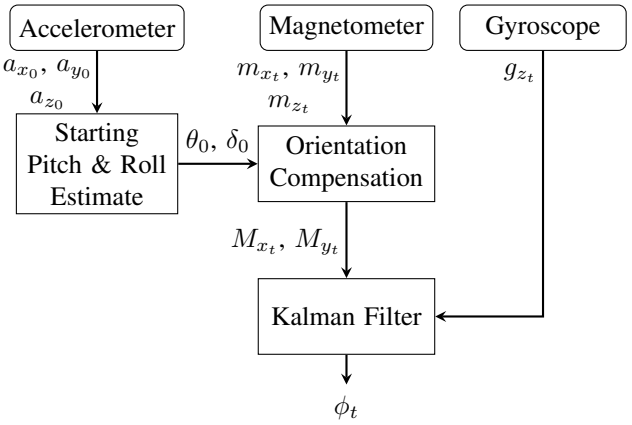


Fig. 3: Sensor Fusion Scheme for IMU

The heading is then computed by [34]:

$$M_{x_t} = m_{x_t} cos\theta_0 + m_{y_t} sin\theta_0 sin\delta_0 + m_{z_t} sin\theta_0 cos\delta_0 \quad (6)$$

$$M_{y_t} = m_{z_t} sin\delta_0 - m_{y_t} cos\delta_0 \quad (7)$$

$$\phi_t = tan^{-1}\left(\frac{M_{y_t}}{M_{x_t}}\right) + \epsilon_t \quad (8)$$

where ϵ_t is a small error in the heading estimate due to the assumption made above and presence of external magnetic fields. We can see that no integration is involved in estimating the heading from magnetometer due to which it does not drift, however it is still sensitive to magnetic disturbances and considerable changes in pitch and roll orientation where our above assumption will not hold. Thus the magnetometer alone is also not sufficient for accurate estimation of heading, however we can fuse the gyroscope and magnetometer based heading estimate by using a Kalman filter to obtain the best possible estimate. The sensor fusion scheme is shown in Fig 3. The Kalman filter formulation is detailed below.

We want to determine the heading hence we define the state vector $\mathbf{x}_t$ as

$$\mathbf{x}_t = \begin{bmatrix}\phi_t\end{bmatrix} \quad (9)$$

The process and measurement equations of a typical Kalman filter are given by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_{t-1} \quad (10)$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \quad (11)$$

where $\mathbf{F}$ is the state transition matrix, $\mathbf{B}$ is the control input model, $\mathbf{w}_{t-1}$ is the process noise, $\mathbf{z}_t$ is the measurement, $\mathbf{H}$ is the observation matrix and $\mathbf{v}_t$ is the measurement noise.

We update the state estimate using gyroscope measurements in the process equation and magnetometer measurements in the measurement equation. The new heading was obtained from the gyroscope through (3). Comparing (3) with (10) we get

$$\mathbf{F} = 1, \qquad \mathbf{B} = \Delta t, \qquad \mathbf{w}_{t-1} = \Delta t.n_{z_t} \quad (12)$$

The process noise covariance $\mathbf{Q}_{t-1}$ is given by $E[\mathbf{w}_{t-1}\mathbf{w}_{t-1}^T]$ and set to $\Delta t^2.\sigma_G^2$ where σ_G^2 is variance of the gyroscope noise.

Heading is computed from magnetometer by (8). Comparing (8) with (11) we get

$$\mathbf{z}_t = tan^{-1}\left(\frac{M_{y_t}}{M_{x_t}}\right), \qquad \mathbf{H} = 1, \qquad \mathbf{v}_t = -\epsilon_t \quad (13)$$

The measurement noise covariance $\mathbf{R}_t$ is given by $E[\mathbf{v}_t\mathbf{v}_t^T]$. Since ϵ_t is time varying and cannot be determined as it depends upon varying external magnetic fields and pitch and roll orientation of the vehicle, we take an adaptive approach and set $\mathbf{R}_t$ according to the residual of the filter.

$$\mathbf{R}_t = A.(\mathbf{z}_t - \mathbf{H}\mathbf{x}_t) \quad (14)$$

where A is a scale factor to tune. We can see from (14) that measurement noise covariance will be high when the magnetometer measurements differ greatly from the gyroscope updated state estimate indicating large error in magnetometer based estimate. Consequently, magnetometer measurements will get lesser weight in the Kalman Filter.

Once the process and measurement models have been defined the general procedure of the Kalman filter is as follows. The $-$ superscript denotes the *a-priori* and $+$ superscript denotes the *a-posteriori* estimate.

$$\mathbf{x}_t^- = \mathbf{F}\mathbf{x}_{t-1}^+ + \mathbf{B}\mathbf{u}_t \quad (15)$$

$$\mathbf{P}_t^- = \mathbf{F}\mathbf{P}_{t-1}^+\mathbf{F}^T + \mathbf{Q}_{t-1} \quad (16)$$

$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_t^-\mathbf{H}^T + \mathbf{R}_t)^{-1} \quad (17)$$

$$\mathbf{x}_t^+ = \mathbf{x}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\mathbf{x}_t^-) \quad (18)$$

$$\mathbf{P}_t^+ = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^- \quad (19)$$

The optimal heading estimate is given by

$$\phi_t = \mathbf{x}_t^+ \quad (20)$$

B. Speed of Vehicle

The speed of the vehicle is obtained from the CAN bus.

C. Maneuver Detection

When a vehicle performs a maneuver, its heading changes in accordance with the maneuver being performed. Fig. 4(a)-(d) show heading response of a vehicle for various maneuvers. In case of a lane change, the driver first steers the vehicle towards the intended lane, causing a sharp change in the heading. The vehicle travels in this direction until it reaches the intended lane. Then the driver steers the vehicle back to drive in the intended lane, causing another sharp change in the heading. The magnitude of change indicates whether the lane change was slow or fast, as shown in Fig 4a and 4b, where a slow lane change causes a heading change of $3-4$ degrees whereas for a fast lane change, this change is $10-12$ degrees. In case of a turn, the change in heading is continuous from the start till the end of the turn. A slow turn takes longer to complete as compared to a fast turn as shown in Fig 4c and 4d, where a slow turn takes 5 seconds while a fast turn takes only 2 seconds to complete.

To detect maneuvers from the changes in heading, we can take its variance over a fixed time window 'W'. In addition to the variance of heading, the speed at which the maneuver is

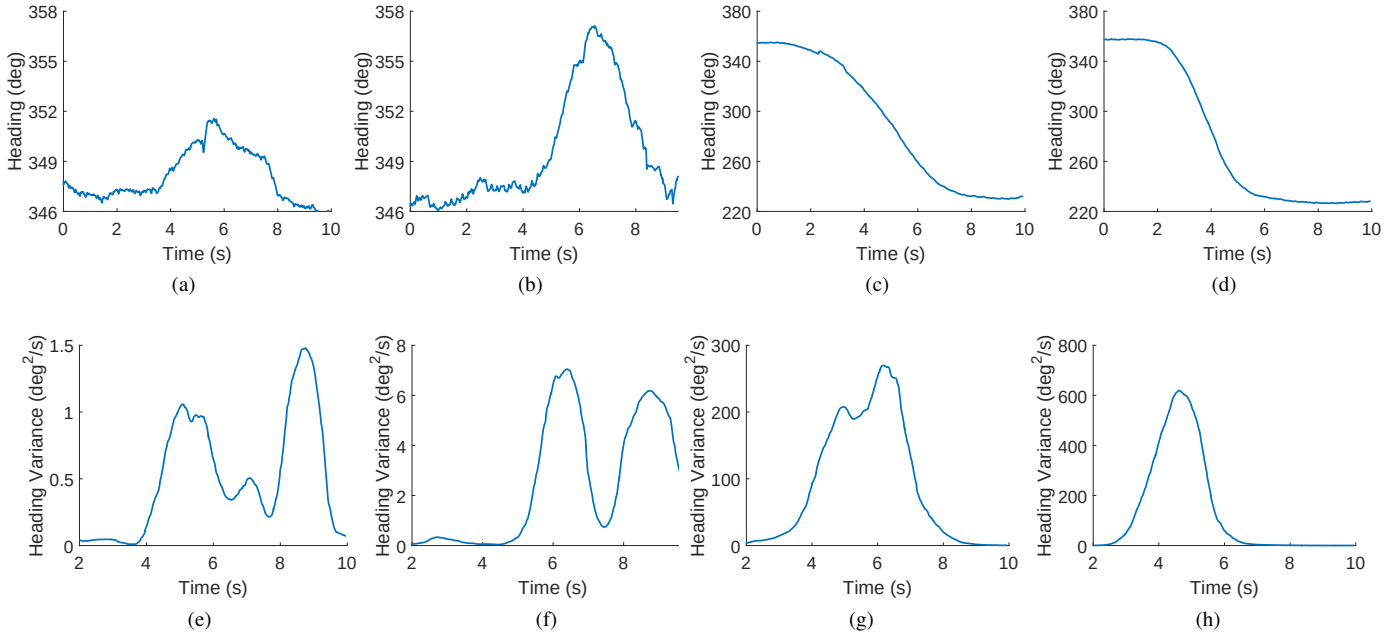
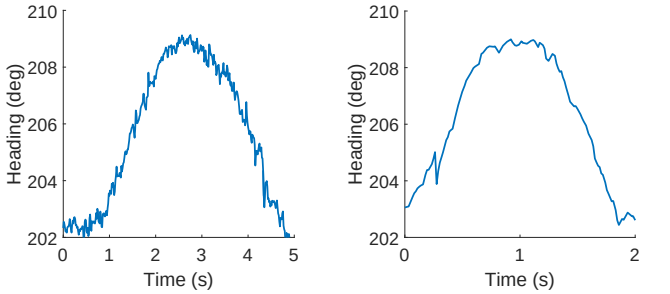


Fig. 4: (a)-(d) Heading during a Slow Lane Change, Fast Lane Change, Slow Turn, Fast Turn (g)-(h) Variance of Heading during a Slow Lane Change, Fast Lane Change, Slow Turn, Fast Turn

performed is also required. This is because at higher speeds, only a small change in heading can cause a significant change in vehicle trajectory. Thus the same change in heading at a lower speed might represent a Slow Lane Change which at a higher speed, represents a Fast Lane Change. This was also experimentally verified and shown in Fig. 5. As we are dealing with data in a window 'W', we can assume that the speed remains constant during this window i.e. equal to its mean value over 'W'. Thus variance of heading and mean of speed are our two features for maneuver detection.

As we want to create a real time system, heading and speed data will be continuously streaming, and the window 'W' will be a sliding window. The length of 'W' is important. If it is large such that the window is able to contain an entire lane change then the heading variance will exhibit only one peak for the whole maneuver and the detection time of the system will increase because the lane change can only be detected when it has been completed and the peak in variance has been observed. If 'W' is small, the window will separately capture the starting and ending of lane change and the heading variance will exhibit two peaks, first one the driver is initiating the lane change and second when the driver is finishing the lane change as shown in Fig 4e and 4f. This not only reduces the detection time but also increases the accuracy of the system by detecting false alarms when a first peak for starting lane change is not immediately followed by a second peak for ending lane change. Thus we selected $W = 1$ second. The variance of heading plots for various maneuvers with $W = 1$ are shown in Figs. 4 (e)-(h).



(a) Heading for a Slow Lane Change at 6.2 mph

(b) Heading for a Fast Lane Change at 32 mph

Fig. 5: Similar heading response of Slow and Fast Lane Change due to different speeds

For each window 'W' of driving data, we create a feature vector '**x**' as discussed before:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{21}$$

where

- x_1 = Mean of the velocity over a time window 'W'
- x_2 = Variance of the heading over a time window 'W'

We need to detect the state of the vehicle for the time duration 'W' based on **x**. The state can be any one of the following:

1. *Going Straight (**x** ϵ GS)*
2. *Performing Slow Lane Change (**x** ϵ SL)*
3. *Performing Fast Lane Change (**x** ϵ FL)*
4. *Performing Slow Turn (**x** ϵ ST)*
5. *Performing Fast Turn (**x** ϵ FT)*

This forms a classical machine learning problem where we need to classify data into the above mentioned five classes. Hence, we use supervised machine learning, in which labeled examples are used to learn a classifying function. This classifying function is then applied to unlabeled data to assign it a

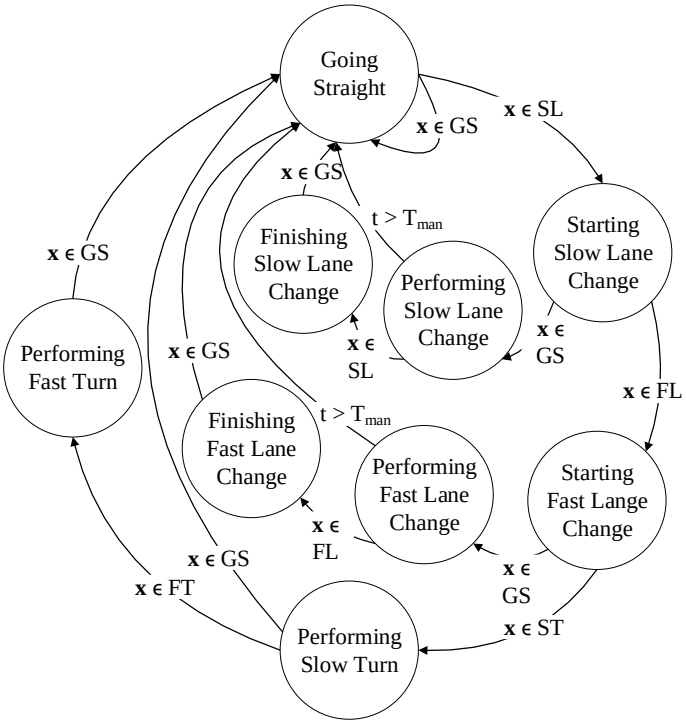


Fig. 6: State Machine

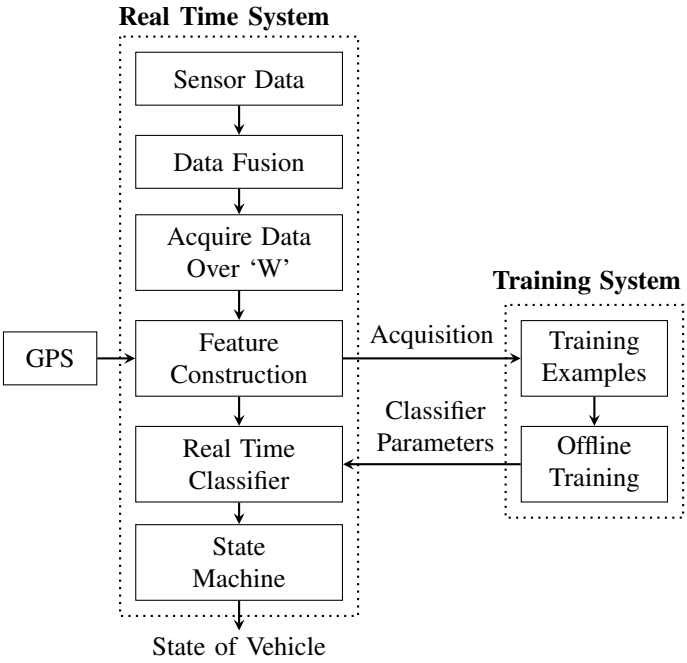


Fig. 7: Flow Chart of Maneuver Detection & Classification

label. The classifying function can be of any order, however we chose a linear classifier to keep the complexity of the system minimum. The set of labeled examples (training set) χ is defined as

$$\chi = \{\mathbf{x}^k, r^k\}_{k=1}^{N} \tag{22}$$

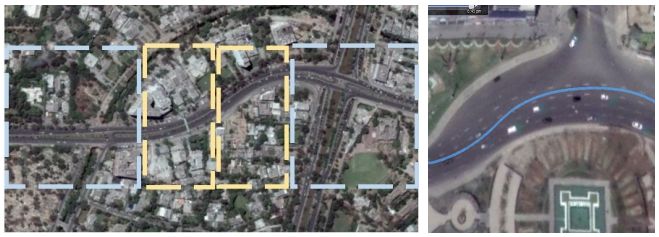
where 'N' is the total number of labeled examples that constitute the training set, $\mathbf{x}^k$ is the feature vector of the 'k_{th}' example and r^k is its label. Since we have five classes, we require four linear classifiers to divide the feature space into five sub-spaces. The state of the vehicle is classified as a certain class if $\mathbf{x}$ lies in the sub-space of that particular class. Each 'i_{th}' linear classifying function $g_i(\mathbf{x})$ has the form

$$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + w_{i0} = \sum_{j=1}^{d} w_{ij}x_j + w_{i0} \tag{23}$$

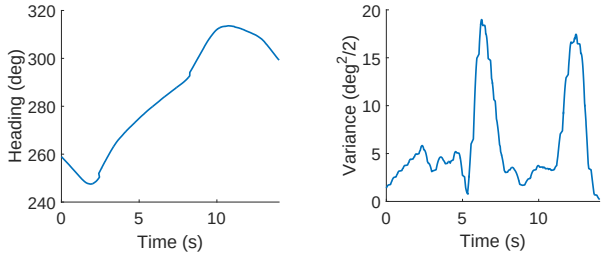
where 'd' is the dimension of the feature vector which is two in our case. The data belongs to 'i_{th}' class iff $g_i(\mathbf{x}) > 0$ && $g_j(\mathbf{x}) < 0\ \forall j > i$.

After the classifying functions have been learnt, they are used for classifying real data. We determine the current state of the vehicle and traverse the state machine shown in Fig. 6. The vehicle leaves 'Going Straight' state as soon as $\mathbf{x} \notin GS$. It sequentially moves along the outer edge of the state-machine as long as variance keeps increases. Once the variance settles, it enters the branch of that maneuver. A maneuver is completed if the variance pattern completes the cycle of the maneuver in the state machine. We can see that lane changes are validated by two peaks. If a timeout condition $t > T_{MAN}$ occurs and the second peak has not been observed then it is detected as a false alarm. The complete maneuver detection algorithm is presented as a flow chart in Fig. 7.

On curved roads, a vehicle traveling in a single lane will exhibit a change in heading due to the curvature of the road (Fig. 8b), which will cause a constant offset in the heading variance (Fig. 8d) and lead to incorrect detection of vehicle state. To mitigate this issue, we use a low cost GPS which is able to provide the location of the vehicle within a few meters accuracy. A low density map table is stored in the device which contains blocks of co-ordinates against the heading variance offset for single lane travel in that region (Fig. 8a). This offset is subtracted from the heading variance whenever the vehicle is traveling through that block.



(a) Map (b) Curved Lane Change



(c) Heading Response (d) Variance of Heading

Fig. 8: Lane Changing on Curved Road

D. Motion of Surrounding Vehicles

A LIDAR sensor or more commonly known as a laser scanner is a distance sensor which measures distance by noting the return time taken by a laser beam to travel to and from a surface at which the beam is pointed. By rotating the LIDAR in angular steps, we can span a field of view and create a 2D distance map of the environment. Detecting the motion of surrounding vehicles using a LIDAR in an urban environment is a complex task, posing a number of challenges: urban entities such as buildings, trees, pedestrians serve as outliers and complicate vehicle detection, movement of detected vehicle causes the impact points returned by it to differ in shape between successive scans and occlusion of vehicles causing multiple vehicles to appear as one cluster.

We adopt a model based approach to detect vehicles from the laser scans in which polygonal shapes are searched within the scan and registered as vehicles if they fulfill certain length criterion. This facilitates in differentiating vehicles from outliers while keeping the algorithm complexity low. The flowchart of vehicle detection and tracking is shown in Fig 9. The LIDAR is installed on a stepper motor for rotating it in angular steps. Hence for each measurement, the LIDAR returns the distance r and the stepper motor gives the angle of the measurement θ. A single complete laser scan will be

$$S = \{r^j, \theta^j\}_{j=1}^{N} \tag{24}$$

where N is the total number of steps in the scan. Two scanner are installed on the hood and boot of the car and rotated to span the 180° field of view in front of and behind the vehicle to detect traffic. To demonstrate each scan processing step, experimentation results are shown in Fig. 10. The actual scene captured by the camera is shown in Fig. 10a whereas the raw scan points returned by the scanner are shown in Fig. 10b. The range of the LIDAR was set to 30m hence all points beyond this range are plotted as 30m.

1) Vehicle Detection: The first step is to determine the number and positions of all vehicles present in the raw laser scan. The scan split into clusters by simple thresholding. We define a range *r_interested*, then traverse the scan and mark all points with distance less than *r_interested* as 1s and the others as 0s. Continuous 1s in the scan give us one cluster. This is shown in Fig. 10c where each cluster has been drawn with a different color. The scan S can now be written as a collection of clusters c:

$$S = \{c_k\}_{k=1}^{N_C}, \quad c_k = \{r^j, \theta^j\}_{j=1}^{N_k} \tag{25}$$

where N_C is the total number of clusters and N_k is the number of points in the k_{th} cluster.

After clustering, we convert the coordinates from azimuth plane to cartesian plane to facilitate the execution of remaining algorithms. Thus for each j_{th} measurement in the azimuth plane we compute its equivalent in cartesian plane by

$$x^j = r^j cos\theta^j \tag{26}$$

$$y^j = r^j sin\theta^j \tag{27}$$

Then we fit each cluster with polygonal segments. This is because the bodies of vehicles return laser points creating

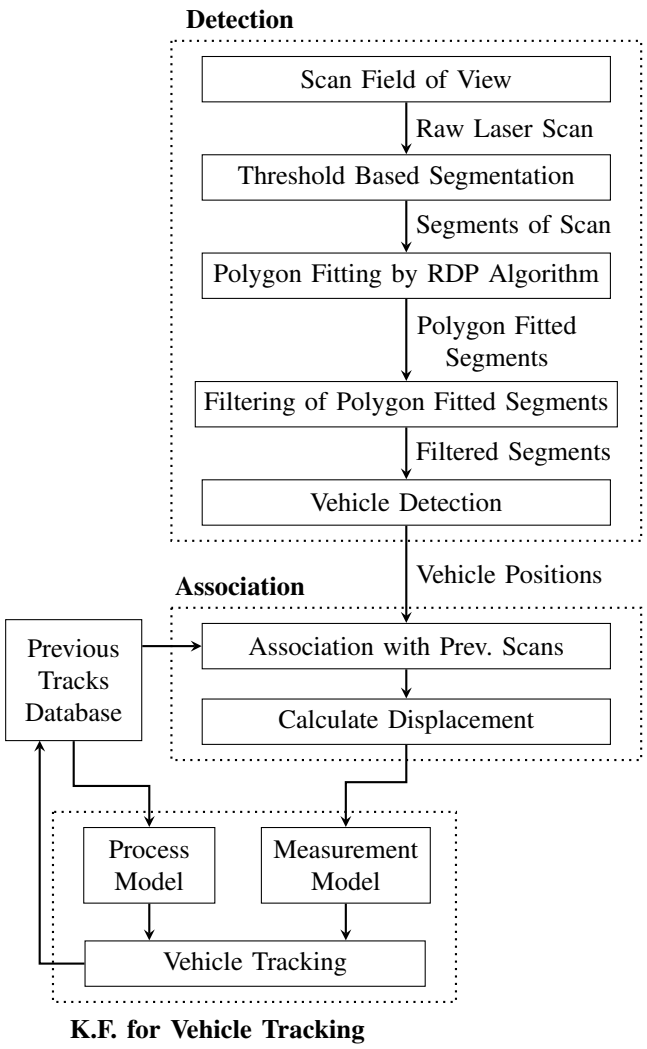


Fig. 9: Flowchart of Vehicle Detection & Tracking by LIDAR

either straight lines or polygonal shapes. To do so, we use the so called RDP (Ramer-Douglas-Peucker) algorithm which is a recursive algorithm [35]. In this algorithm, the first and last points (say A and B) of the cluster are joined through a straight line (AB). Then that point in the cluster is determined from which, the distance to this line segment is the greatest (say C). The line segment AB is broken into AC and CB and then the algorithm is again applied individually on these segments. The algorithm continues until the distance of the farthest point from the segment is less than a certain error threshold. The segments after polygon fitting are shown in Fig. 10d. The scan can now written as a collection of clusters where each cluster is fitted with a number of line segments and each j_{th} segment v^j is defined by its two end points $\{(x_1^j, y_1^j), (x_2^j, y_2^j)\}$.

$$S = \{c_k\}_{k=1}^{N_S}, \quad c_k = \{v^j\}_{j=1}^{N_S}, \quad v^j = \{(x_1^j, y_1^j), (x_2^j, y_2^j)\} \tag{28}$$

where N_S is the total number of segments in the k_{th} cluster.

After polygon fitting we need to detect vehicles present in these polygons. The laser scan contains many outliers in addition to the vehicles. These are surrounding buildings and pedestrians. Separating vehicles from the outliers is a complex

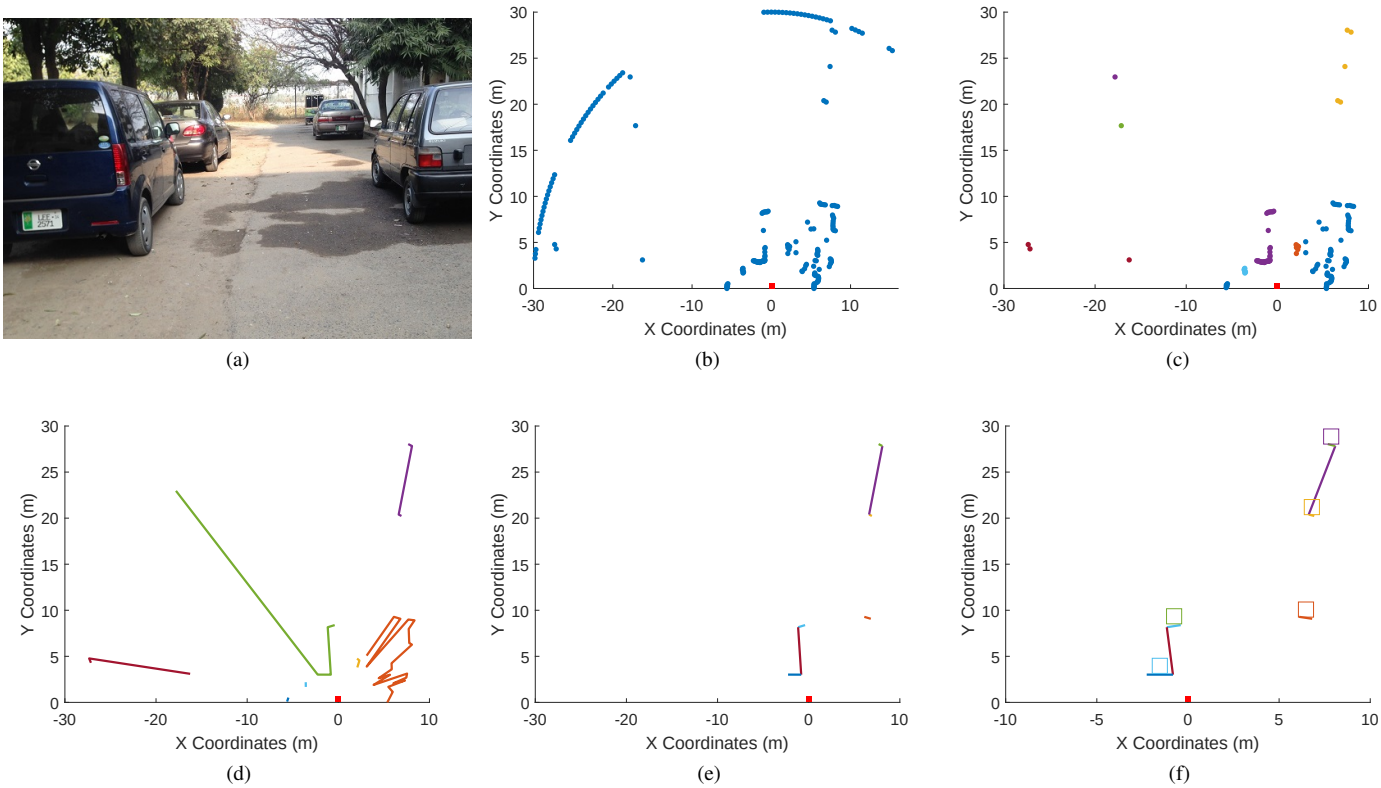
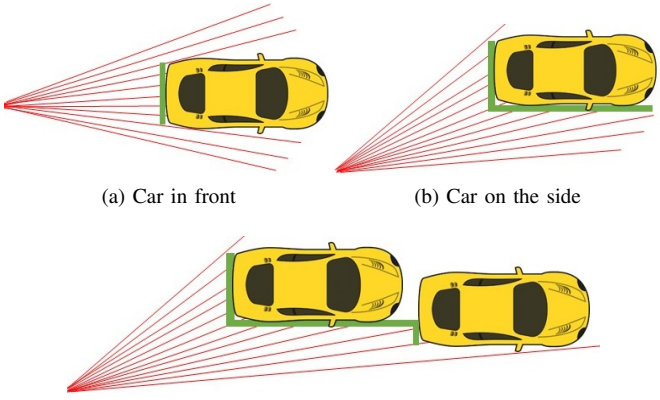


Fig. 10: Stages of Laser Scan Processing (a) Camera view of the scene - 3 vehicles are fully visible, 2 are occluded (b) Raw Scan - Impact points returned by the laser (c) Scan after segmentation - Each segment is drawn with different color (d) Scan after polygon fitting (e) Scan after filtering of segments (f) Position of detected vehicles



(a) Car in front (b) Car on the side

(c) Occlusion at the back

Fig. 11: Shapes of scan points returned from vehicles

job and can only be done upto a certain degree. We observe that the laser points returned from a preceding vehicle are either a straight line (vehicle directly in front of our car) as shown in Fig. 11a or L shaped (vehicle is towards the side of the car) as shown in Fig. 11b. As far as occlusion of vehicles is concerned, we observe that a portion of the occluded vehicle is present in the scan, clustered with the fully observable vehicle as shown in Fig. 11c.

Based on these observations, we define a set of rules. For each segment v^j in a cluster, we calculate three things:

1) Length of the segment

$$l^j = \|v^j\| \tag{29}$$

2) Its angle with the next adjacent segment

$$\psi^j = cos^{-1}\left(\frac{v^j.v^{j+1}}{\|v^j\|.\|v^{j+1}\|}\right) \tag{30}$$

3) The orientation 'O^j' of the segment w.r.t. the laser scanner i.e. whether it is horizontal or vertical. Given that the laser is at the coordinates $(0, 0)$, the orientation is computed as

$$V_j = \left[0 - \frac{x_1^j + x_2^j}{2} \quad 0 - \frac{y_1^j + y_2^j}{2}\right] \tag{31}$$

$$\gamma^j = cos^{-1}\left(\frac{V^j.v^j}{\|V^j\|.\|v^j\|}\right) \tag{32}$$

$$O^j = \begin{cases} \text{Horizontal} & \text{if } |\gamma^j - 90°| > 45° \\ \text{Vertical} & \text{Otherwise} \end{cases} \tag{33}$$

Then based on the set of rules, we filter the segments and discard all those which do not follow the rules. We assume that the length of a vehicle is 'L' meters and width is 'W' meters. For the segments on the left side of the scanner:

1) If a segment does not have any adjacent segments, then its length should satisfy $|l^j - W| < \zeta$ if it is a horizontal segment i.e. we are observing the back side of the vehicle

(Fig. 11a) or $|l^j - L| < \zeta$ if is vertical i.e. we are observing the side of the car. ζ is the leniency allowed in the segment length observation.

2) A horizontal segment is allowed if its adjacent segment is vertical and its angle with this segment satisfies $|\psi^j - 90°| < \xi$ as these two segments represent a vehicle observed from the side as in Fig. 11b. ξ is the leniency allowed in the angle observation.
3) A vertical segment is allowed if its adjacent segment is horizontal and its angle with this segment satisfies $|\psi^j - 270°| < \xi$ as these represent an occluded vehicle cluster (Fig. 11c). The vertical segment is a part of one vehicle and the adjacent horizontal segment is the part of another.

For segments on the right side of the scanner, rules 2 and 3 will be reversed in terms of orientation (vertical/horizontal) constraints as the scan points returning from vehicles on this side are represented by polygons in the reverse manner.

After detecting vehicles, we compute the position $\mathbf{P}$ of each vehicle relative to the scanner in the following manner.

1) If the vehicle is represented by a single horizontal segment v^j then $\mathbf{P} = \left(\frac{x_1^j + x_2^j}{2}, y_1^j + \frac{L}{2}\right)$ since $y_1^j \simeq y_2^j$ for a horizontal segment.
2) If the vehicle is represented by a single vertical segment v^j then $\mathbf{P} = \left(x_1^j + \frac{W}{2}, \frac{y_1^j + y_2^j}{2}\right)$ since $x_1^j \simeq x_2^j$ for a vertical segment.
3) If the vehicle is represented by combination of horizontal and vertical segments v^j and v^{j+1} respectively then $\mathbf{P} = \left(\frac{x_1^j + x_2^j}{2}, \frac{y_1^{j+1} + y_2^{j+1}}{2}\right)$.

Hence now we have the total number of vehicles present in the current scan 'N' and the position (x_P, y_P) of each vehicle.

$$\mathcal{P} = \{\mathbf{P}^i\}_{i=1}^{N}, \quad \mathbf{P}^i = (x_P^i, y_P^i) \tag{34}$$

2) Vehicle Association: After detecting the position of all vehicles present in the current scan, we need to associate them with vehicles in the previous scans i.e. we need to determine which vehicle from the current scan was which vehicle in the previous scans, so that we can compute its speed and predict its future trajectory.

Here we define the term 'track' denoted by T which refer to the motion characteristics of a vehicle that has been tracked over successive scans and its speed has been determined. Each track at time t consists of vehicle position and speed. If the total number of tracks are N_T then

$$\mathbf{T}_t = \{T_t^j\}_{j=1}^{N_T}, \quad T_t^j = \{X_t^j, Y_t^j, V_{X_t}^j, V_{Y_t}^j\} \tag{35}$$

Vehicle association problem is basically the task of associating each vehicle with one of the existing track. To find the appropriate track for a vehicle, we use the nearest neighbor method. When association is carried out, the position and velocity of the tracks have not been updated to the current time 't' and contain estimates of previous time instant '$t-1$'. Therefore, each track is projected one time step forward according to its speed to predict its current position $(\mathcal{X}_t^j, \mathcal{Y}_t^j)$ using

$$\mathcal{X}_t^j = X_{t-1}^j + \Delta t.V_{X_{t-1}}^j, \;\; \mathcal{Y}_t^j = Y_{t-1}^j + \Delta t.V_{Y_{t-1}}^j \tag{36}$$

That vehicle is associated with the track whose position is within $1m$ of the projected position of the track i.e. i_{th} vehicle is associated with the j_{th} track if

$$\sqrt{(x_P^i - \mathcal{X}_t^j)^2 + (y_P^i - \mathcal{Y}_t^j)^2} < 1 \tag{37}$$

If multiple vehicles are in contention for the same track then that vehicle is selected which is closest to the projected track position. Same is the case if one vehicle is in contention for multiple tracks. If a vehicle cannot be associated to any of the existing tracks, (a new vehicle has appeared in the scan) then we create a new track for that vehicle and initialize its speed by 0. If a track does not get associated with any vehicle in the current scan (the vehicle representing that track has moved out of range or become fully occluded) then the track is deleted.

3) Kalman Filter for Tracking: Vehicle positions detected by the LIDAR suffer from errors due to several uncertainties in the detection process. Consequently the tracks of these vehicles are also irregular and can be smoothed with the help of a Kalman filter which fuses the LIDAR detected position of each vehicle with a constant velocity model. One Kalman Filter is required per track. Hence for k tracks, we require k Kalman Filters. The state vector for the Kalman Filter of each track consists of its position at time t.

$$\mathbf{x}_t = \begin{bmatrix} X_t & Y_t \end{bmatrix} \tag{38}$$

The state is updated in the process step by

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \Delta t.\mathbf{V}_{t-1} \tag{39}$$

where $\mathbf{V}_{t-1}$ is the estimate of the track's speed

$$\mathbf{V}_{t-1} = \begin{bmatrix} V_{X_{t-1}} & V_{Y_{t-1}} \end{bmatrix} \tag{40}$$

which is related to the true track speed $\mathbf{v}_{t-1}$ by

$$\mathbf{v}_{t-1} = \mathbf{V}_{t-1} + \boldsymbol{\varepsilon}_{t-1} \tag{41}$$

where $\boldsymbol{\varepsilon}_{t-1}$ is the error in speed estimate. Substituting (41) into (39) and comparing with (10), we get

$$\mathbf{F} = \mathbf{I}_2, \quad \mathbf{B} = \Delta t, \quad \mathbf{w}_{t-1} = \Delta t.\boldsymbol{\varepsilon}_{t-1} \tag{42}$$

The process noise covariance matrix $\mathbf{Q}_{t-1}$ is given by $E[\mathbf{w}_{t-1}\mathbf{w}_{t-1}^T]$ and set equal to $\Delta t^2 \sigma_\varepsilon \mathbf{I}_2$ where σ_ε is the variance of the speed estimate error and tuned between 0 and 1 for performance.

The position of the vehicle associated with this track, as measured by the LIDAR was given in (34)

$$\mathbf{P}_t = \begin{bmatrix} x_p & y_p \end{bmatrix} \tag{43}$$

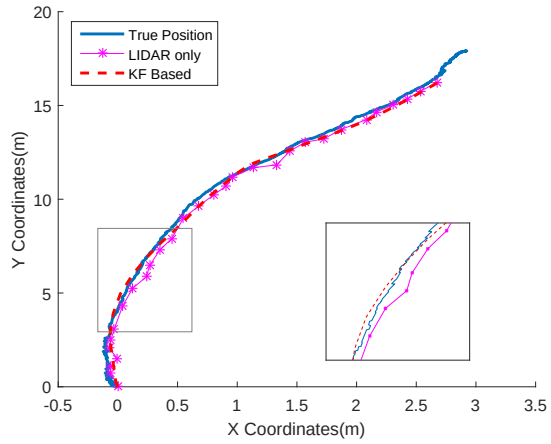
where we have dropped the superscript for simplicity. If the error in this LIDAR detected position is $\boldsymbol{\rho}_t$, then

$$\mathbf{x}_t = \mathbf{P}_t + \boldsymbol{\rho}_t \tag{44}$$

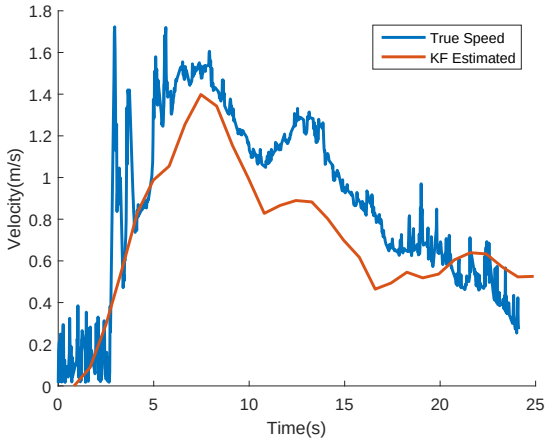
Substituting (44) into (43) and comparing with (11) we get

$$\mathbf{z}_t = \mathbf{P}_t, \quad \mathbf{H} = \mathbf{I}_2, \quad \mathbf{v}_t = \boldsymbol{\rho}_t \tag{45}$$

The measurement noise covariance $\mathbf{R}_t$ is given by $E[\mathbf{v}_t\mathbf{v}_t^T]$. Since $\boldsymbol{\rho}_t$ is time-varying and cannot be determined, we take an adaptive approach to set it. Considering that more scan points are returned from a vehicle that is closer to the scanner



(a) Vehicle Location Estimates



(b) Vehicle Speed Estimates

Fig. 12: Kalman Filter Results

thus giving more accurate position as compared to a distant vehicle that returns less scan points, we set $\mathbf{R}_t$ according to the distance of the vehicle from the scanner:

$$\mathbf{R}_t = \left(B.\sqrt{(0-x_p^2)+(0-y_p^2)}\right)\mathbf{I}_2 \tag{46}$$

where B is a constant to be tuned. Thus more weightage is given to the LIDAR measurement in the Kalman filter if the vehicle is closer the scanner and returning more scan points.

The procedure of the filter is the same as given in Eqs. (15)-(19). Once the a-posteriori state estimate has been determined, the track position is given by

$$X_t = \mathbf{x}_t^+(1), \quad Y_t = \mathbf{x}_t^+(2) \tag{47}$$

and track speed is updated by

$$V_{X_t} = \frac{X_t - X_{t-1}}{\Delta t}, \quad V_{Y_t} = \frac{Y_t - Y_{t-1}}{\Delta t} \tag{48}$$

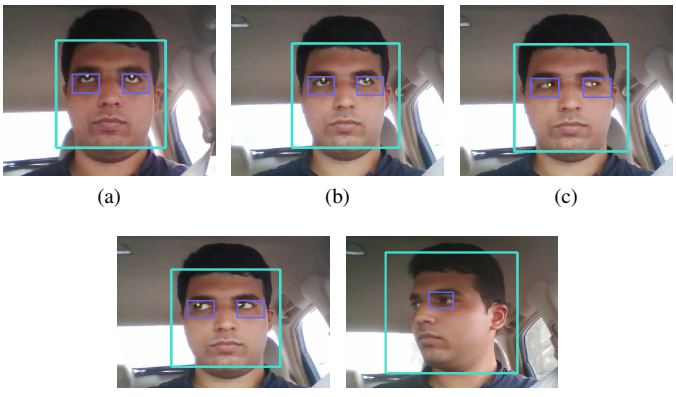
Note that we could have included the speed components into the state vector as well, however the Kalman filter would then be of 4^{th} order and computationally very expensive. To test the performance of the formulated Kalman filter, we installed a second test vehicle with a high precision GPS and recorded its position for ground truth while tracking the vehicle from the LIDAR. The results are shown in Fig. 12 where it can be seen that the Kalman filter performs smooth tracking of the vehicle as compared to using LIDAR only.

E. Tracking Eye Movement

Tracking the driver's eye movement helps in predicting certain maneuvers in advance and allow the collision warning system to generate premature warnings. For example, if the driver is looking to change to the left lane then he might gaze on to the left lane or in the left side mirror. The system after receiving this indication, can check for occupancy of the left lane and generate an alarm if a potential accident situation will be created by this lane change. To track the eye movement, a camera is installed inside the vehicle behind the steering wheel and pointed at the driver's face. To track eye centres, the algorithm given in [36] is used which proposed a multi-stage solution to detecting eye centers. The algorithm is very accurate and has low complexity which enables it to run at a high frames-per-second even on low cost processors.

First, a frame is captured from the video stream and the driver's face is detected from the frame using Viola-Jones algorithm [37] which applies machine learning to detect faces. We noted that the camera will always point at the frontal portion of the driver's face except when he is looking in the left side mirror, in which case his face will be turned towards the left as shown in Fig. 13e. Hence two classifiers were run simultaneously, one for frontal face detection and second for side face detection. After the face region has been identified, eye region is estimated from it using an assumed face geometry. From these eye regions, a minimum gradient algorithm is applied to detect eye centers as given in [36]. The results are shown in Fig. 13.

We used eye movement tracking to identify five scenarios: looking straight (Fig. 13a), looking onto the right lane ahead (Fig. 13b), looking into the right side mirror (Fig. 13c), looking onto the left lane ahead (Fig. 13d) and looking into the left side mirror (Fig. 13e).



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

Fig. 13: Screenshots of Eye Center Localization Process - Eye Centers while (a) Looking Straight (b) Right Lane (c) Right Side Mirror (d) Left Lane (e) Left Side Mirror

F. Behavior Modeling

After detecting the maneuvers performed by the vehicle and the position of the surrounding vehicles, the next task is to model the driving behavior of the driver and determine whether he is driving safely or rashly. To do so, we define the term 'Driving Index' (*D.I.*) which is a score between 0 and 100 assigned to the driver based on his driving. A higher score indicates rash driving behavior whereas a lower score indicates safe driving behavior.

Before explaining *D.I.* we define the term 'safe distance' as the minimum distance that the vehicle must keep with the preceding vehicle. It is also commonly known as braking distance and depends upon the speed of the vehicle v_0 at the instant of applying the brakes. We can derive the equation for braking distance by assuming that the deceleration of the vehicle during braking a_t is constant i.e. $a_t = a$ and using the third equation of motion:

$$v_t^2 = v_0^2 + 2ad_{brake} \tag{49}$$

By setting the final velocity to zero, we can find d_{brake} as

$$d_{brake} = \frac{-v_0^2}{2a} \tag{50}$$

The work done during the braking process $F = ma$ is actually done by the friction $F = -\mu mg$. Comparing these two equations, we get $a = -\mu g$ which substituted into (50) gives us

$$d_{brake} = \frac{v_0^2}{2\mu g} \tag{51}$$

μ is the coefficient of friction and depends upon tyre and road characteristics. A graph of safe distance vs speed is plotted in Fig. 14 with $\mu = 0.7$.

D.I. is initialized by 0 at the start of the drive. A penalty is incurred and *D.I.* is increased if the driver performs any one of the following dangerous maneuvers:

1. Exceeding the speed limit.
2. Violating safe distance with the preceding vehicle.
3. Decelerating and violating the safe distance with the following vehicle.
4. Performing a fast lane change.
5. Performing a slow lane change such that the safe distance is violated with either the preceding or following vehicle in the new lane.
6. Performing a fast turn.
7. Performing a slow turn not from the exit lane, while there was a vehicle following in the exit lane.
8. Needlessly accelerating & decelerating continuously in a short period of time.

Also, *D.I.* is decreased monotonically with time if no dangerous maneuvers are performed. Thus if the driver performs several dangerous maneuvers in a short span of time, *D.I.* will increase and exceed a certain threshold ($D.I._{thresh}$) resulting in the driver getting reported. However, the driver will not be reported if a dangerous maneuver occurs once in a while because *D.I.* begins to automatically decrease as

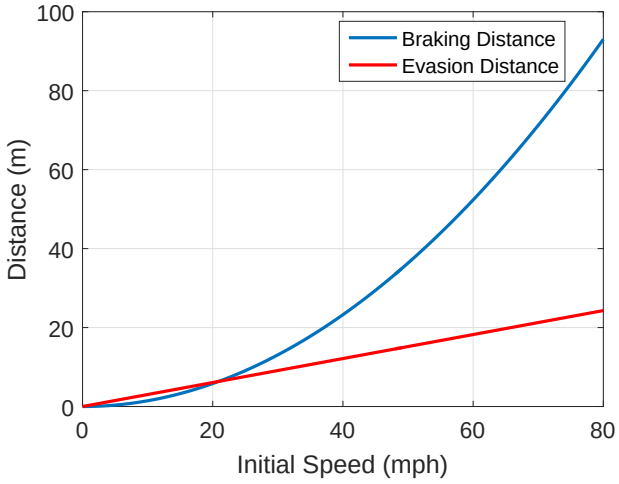


Fig. 14: Safe & Evasion Distance as a function of Vehicle Speed

described above. Thus the mathematical expression for *D.I.* was formulated as

$$D.I._t = \min\left(100,\ D.I._{t-1} - \underbrace{C}_{Decrease} + \underbrace{D * R}_{Increase}\right) \tag{52}$$

where

$$R = \begin{cases} 1, & \text{Dangerous Maneuver} \\ 0, & \text{Otherwise} \end{cases} \tag{53}$$

and C & D are constants which control the behavior of *D.I.* evolution and are upto the system designer. The values set in our system were $C = 10^{-1}$ & $D = 10$.

Once the driver has been detected as driving dangerously, any counter measure can be taken by the system depending upon the provision e.g. stopping the car or notifying traffic authorities.

G. Collision Warning

The system continuously monitors the risk of a collision with any of the surrounding vehicles and generates an alarm if a potential collision situation is determined. 'Safe distance' defined in Section III-F refers to the braking distance that a vehicle must keep and is a good indicator of the quality of driving, however accidents are extreme situations demanding the driver to do much more than braking. Mostly, an evasive maneuver like steering away from the obstacle is performed to avoid an accident. So in addition to the 'Safe distance' defined above, another term called 'Evasion distance' is defined here which is the minimum distance that the vehicle must keep from the preceding vehicle so that it can perform an evasive maneuver in case of an emergency. It is also a function of vehicle speed and can be derived as follows [38]:

Consider Fig. 15 where a vehicle performs an evasive maneuver towards the left by moving along a circle of radius 'r'. The distance of the vehicle from the preceding vehicle is called evasion distance and is given by 'd_s'. The center of the

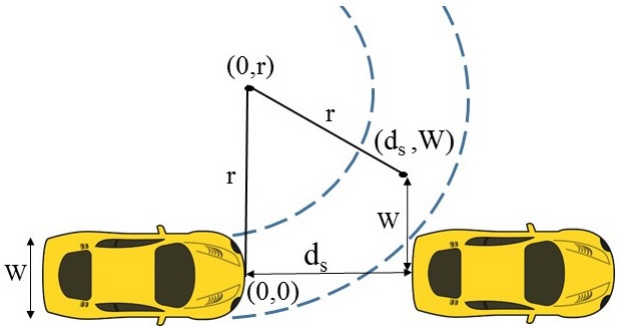


Fig. 15: Evasive Maneuver

circle is located at $(0, r)$. We know that the general equation of a circle with center (h, k) and radius r is given by

$$(x-h)^2 + (y-k)^2 = r^2 \tag{54}$$

Applying this equation and solving for d_s gives us

$$(d_s - 0)^2 + (W - r)^2 = r^2 \tag{55}$$

$$d_s = \sqrt{r^2 - W^2 - r^2 + 2Wr} \tag{56}$$

$$= \sqrt{2Wr - W^2} \tag{57}$$

We know that in circular motion, $\alpha = \frac{v^2}{r}$, where α is the centripetal acceleration which is provided by tyre friction $\alpha = \mu g$. Thus substituting for r

$$d_s = \sqrt{2W\frac{v^2}{\mu g} - W^2} \tag{58}$$

We can see that the evasion distance varies almost linearly in response to the initial speed of the vehicle whereas the braking distance in (50) varied quadratically. The evasion distance has been plotted in Fig. 14 for $\mu = 0.7$ and $W = 1.5m$. Evasion distance is much less than the braking distance at higher speeds.

Apart from the evasion distance we define 'time to collision' as the time remaining for a surrounding vehicle to collide with the ego-vehicle if they continue to move at their current speeds. We check those tracks from the existing tracks whose relative velocity w.r.t our vehicle is negative which means that they are moving towards us. Since the ego-vehicle is at (0,0) time of collision is then determined by

$$t = sol_{t_0}\{P_{obj}(t + t_0) = 0\} \tag{59}$$

We generate an alarm for risk of collision if any of the following three scenarios occur.

1. If the vehicle is keeping less than the evasion distance from the preceding vehicle.
2. If the time to collision with any vehicle is less than 2 seconds.
3. If the vehicle is about to change lane and changing lane will put it in within < 2 seconds of collision.

H. Node Architecture & Communication Protocol

The hardware of the system consists of six sensor nodes which are installed on the vehicle. Each node consists of one or more sensors, a processor and a wireless communication module and is responsible for sensing a specific modality, processing it and transmitting the processed information onto the wireless channel. The architecture of each node alongwith the transmission and reception of shared messages is shown in Fig 16. The placement of nodes on the vehicle is shown in Fig. 17.

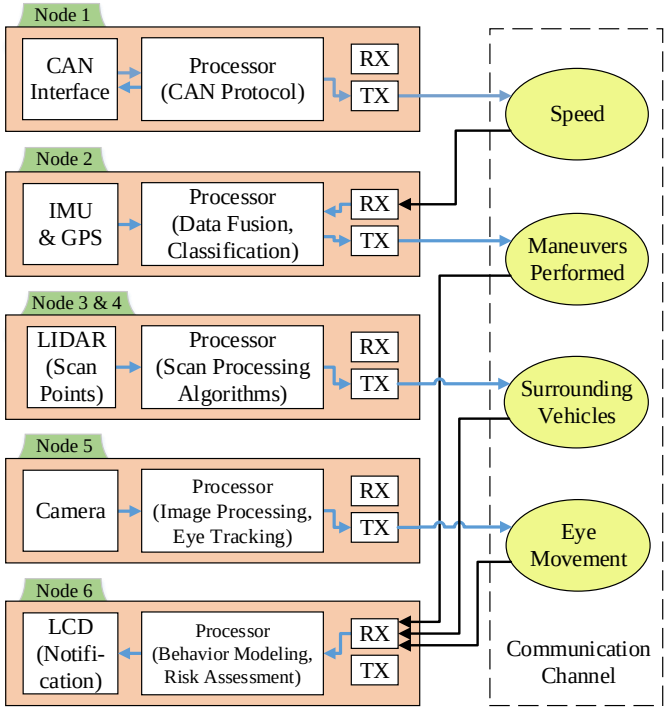


Fig. 16: Node Architecture & Communication Protocol

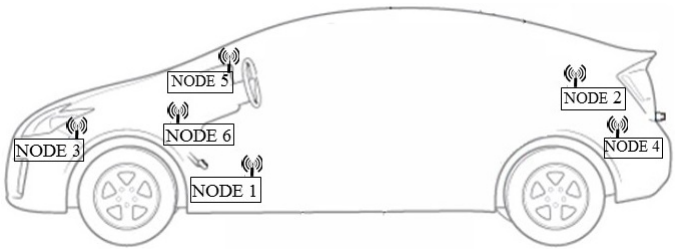


Fig. 17: Placement of Sensor Nodes in the car

The first node is responsible for obtaining the speed of the vehicle from the CAN bus and is installed under the steering wheel where the CAN bus connector is available in most vehicles. It contains a tranciever circuit to communicate with the CAN interface, a processor which requests speed from the CAN interface using CAN protocol and a wireless module to transmit the vehicle speed onto the wireless channel.

The second node is responsible for maneuver detection. It contains inertial sensors and a low cost GPS, and is installed in the boot of the vehicle so that the magnetometer is free from any magnetic disturbances that may arise from the vehicle engine. This node receives the vehicle speed from the wireless channel, fuses the data obtained from inertial sensors, receives vehicle location from the GPS and detects the maneuver performed by the car using the algorithms described in Section

III-C. The detected maneuver is then transmitted onto the wireless channel.

The third and forth nodes deal detecting and tracking the surrounding vehicles. One node is placed at the front of the vehicle, with the LIDAR on the hood and the processor and wireless module inside the hood and the other node is placed at the vehicle with the LIDAR on the boot and the processor and wireless module inside the boot. The processor controls the stepper, receives scan points from the LIDAR. After a scan is complete, it processes the whole scan to detect and track surrounding vehicles as described in Section III-D and transmits the position and trajectory of each surrounding vehicle onto the wireless channel.

The fifth node contains the camera and is responsible for tracking eye movement. It is installed just above the steering wheel where the camera will have a complete view of the driver's face. The processor runs the image processing and eye tracking algorithm of Section III-E and transmits the eye tracking results on to the wireless channel.

The sixth node is placed inside the car. It has an LCD which is mounted just beside the steering wheel. It receives the maneuver information from the Node 2, surrounding vehicles information from Nodes 3 and 4 and eye tracking results from Node 5. It uses this information to apply behavior modeling and collision risk algorithms described in Sections III-F & III-G. It displays information to the driver regarding his driving.

All nodes transmit on the same frequency channel to facilitate easy transmission and reception among the nodes. To avoid collision of messages listen before talk algorithm was implemented. Each node assessed the channel to check if it is free and only transmits if it is free. Each message has a source address label so that the other nodes can check which message is being transmitted. The recipient node can either receive the message if it is relevant or discard it.

IV. Experiments

To experimentally test the system, we installed the wireless nodes on a test vehicle shown in Fig. 18a. The IMU used in Node 2 (Fig. 18b) comprised of STMicroelectronic's LSM303DHLC which contains a tri-axial accelerometer & magnetometer and L3GD20 which is a tri-axial gyroscope. The low cost GPS was SkyNav's SKM53 (Fig. 18d). Nodes 3 and 4 used Pulsed Light's LIDAR-Lite module which is a single beam laser scanner (Fig. 18c). Node 5 used Logitech's HD webcam C290 for eye tracking (Fig. 18e). Node 6 contained a 256x64 pixels JHD12864 Graphical LCD for notification purposes (Fig. 18f). Nodes 1-4 and 6 used STMicroelectronic's STM32F3 microcontroller board which runs at 70 MHz as the processing unit whereas Node 5 used a Raspberry Pi board running at 700 MHz for image processing algorithms. Each node was equipped with Texas Instrument's CC1101 module which is a sub-GHz tranciever operating on 868 MHz channel for wireless communication.

To generate efficient ground truth for comparison, the vehicle was also equipped with Advanced Navigation's Spatial GPS module (Fig. 18g) which a high precision GPS receiver and provides sub-meter accurate position of the vehicle and a Camera to record the proceedings simultaneously (Fig. 18h).

(a) Instrumented Vehicle (b) IMU & GPS node in the boot

(c) LIDAR on the hood (d) Low Cost GPS under rear screen

(e) Camera node (f) LCD for notification

(g) High precision GPS on the roof (h) Camera to record proceedings

Fig. 18: Experimental Test Bed

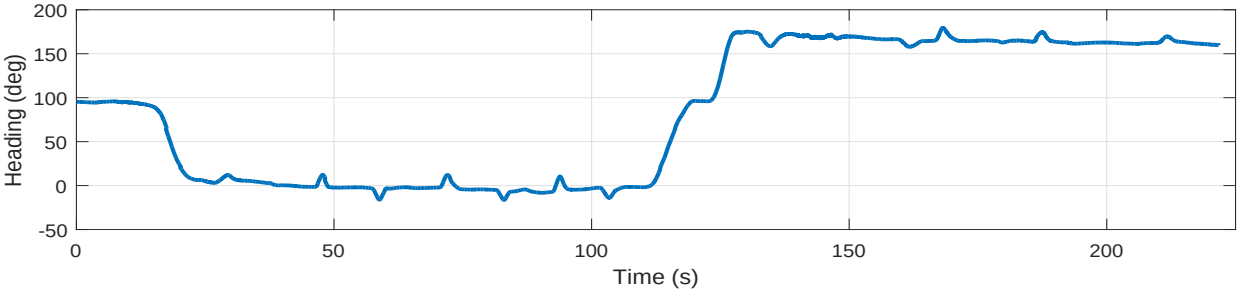
TABLE III: Number of Examples For Each Maneuver

	Maneuver	Number of Examples
1.	Going Straight	33
2.	Slow Lane Change	43
3.	Fast Lane Change	36
4.	Slow Turn	40
5.	Fast Turn	17

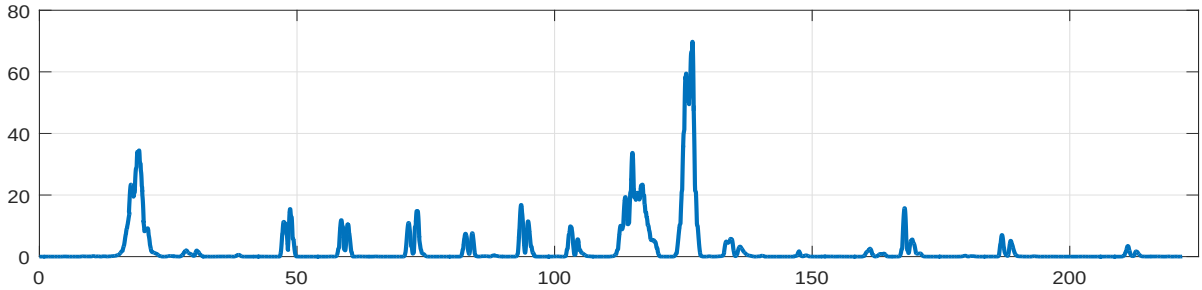
To train the system, several examples of each maneuver were collected. Table III shows the number of examples of each maneuver. To learn the parameters of each classifier, we used the minimum error gradient approach presented in Chapter 10 of [39]. MATLAB machine learning toolbox was used.

The vehicle was driven for 1 km in traffic with the driver randomly performing different maneuvers including slow and fast lane changes and turns. The system kept detecting the maneuvers and tracking the surrounding traffic to determine driving behavior and risk of accident. The results are shown in Fig. 19.

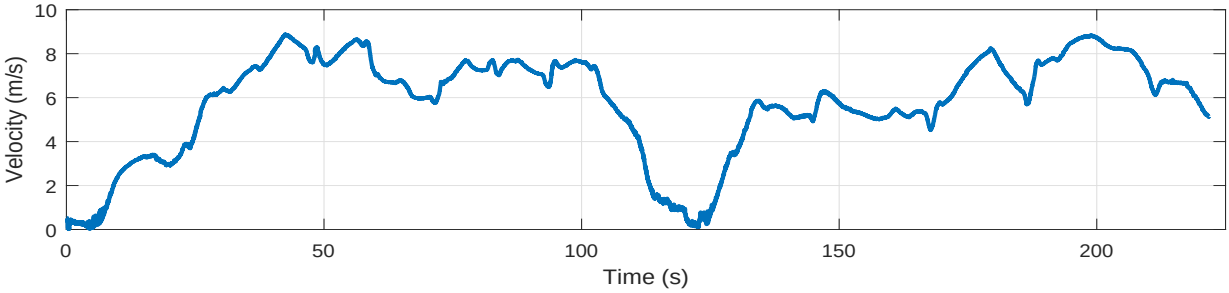
(a) GPS mapping of Test Drive



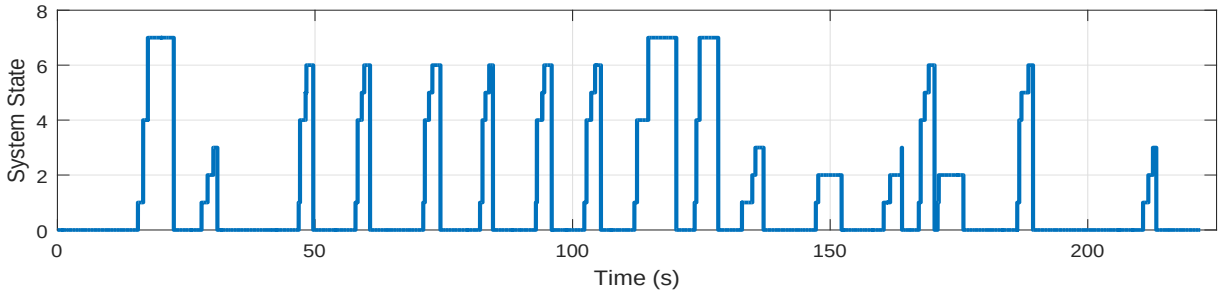
(b) Heading Response during the Test Drive



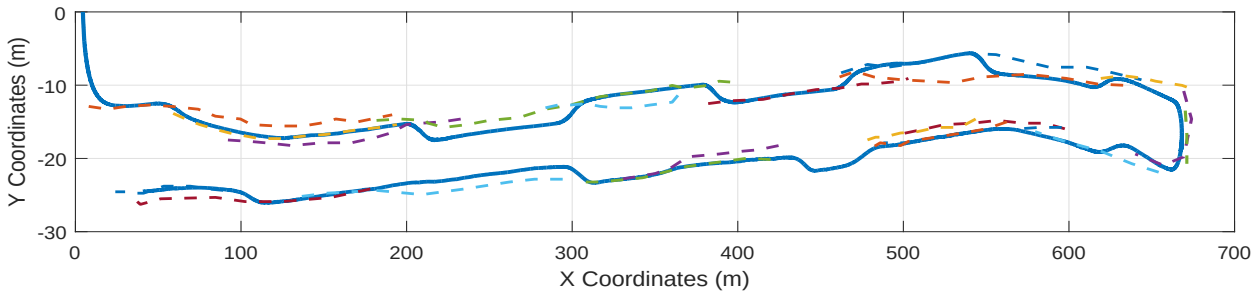
(c) Variance of Heading during the Test Drive



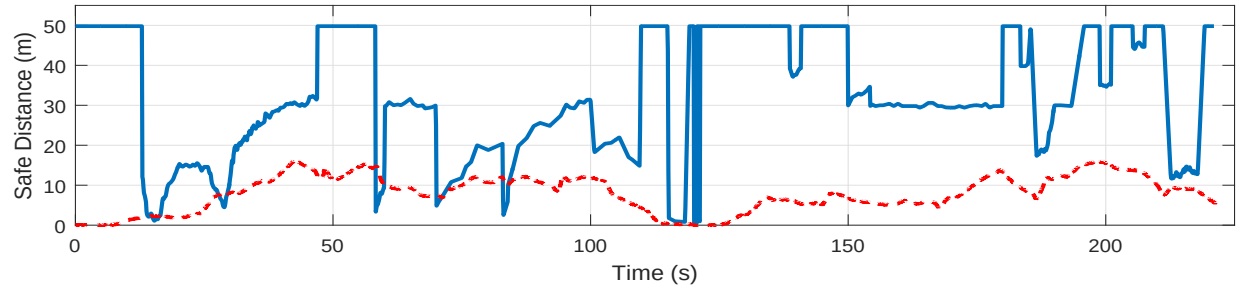
(d) Velocity during the Test Drive



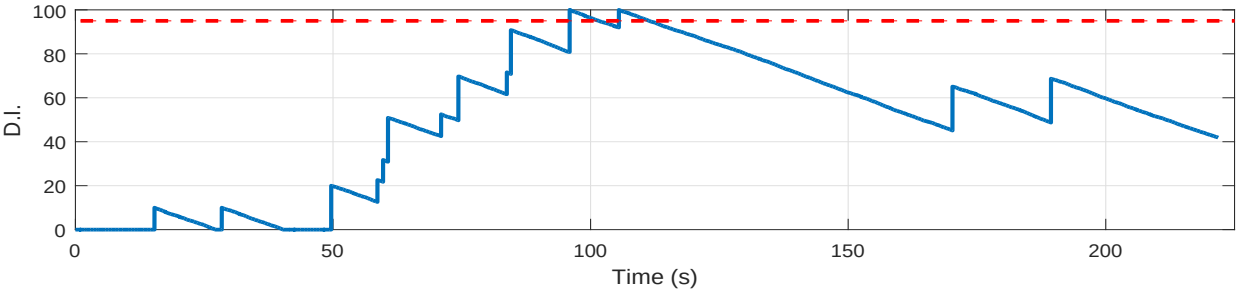
(e) Evolution of States in State Machine, 0: Going Straight, 1: Starting Slow Lane Change, 2: Performing Slow Lane Change, 3: Finishing Slow Lane Change, 4: Starting Fast Lane Change, 5: Performing Fast Lane Change, 6: Finishing Slow Lane Change, 7: Performing Slow Turn, 8: Performing Fast Turn



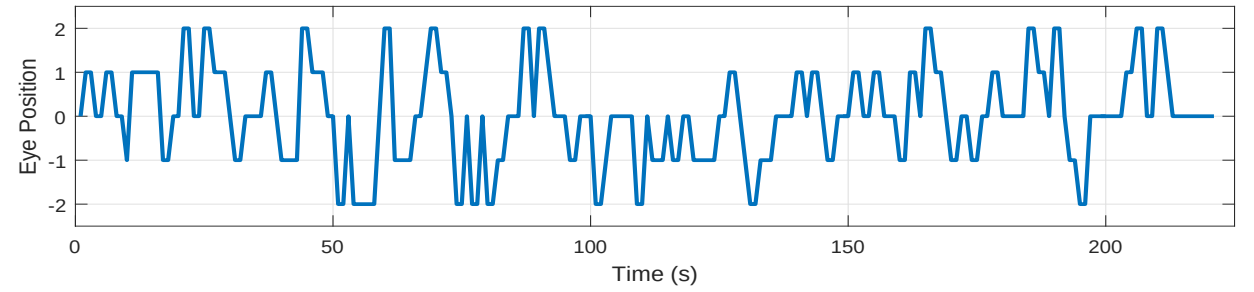
(f) Trajectory of Test Drive and the Surrounding Tracks



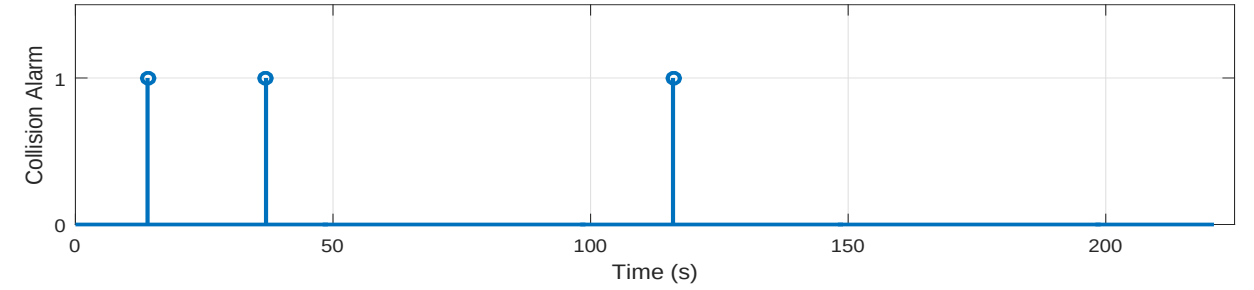
(g) Safe Distance kept by the driver during the Test Drive



(h) Evolution of Driving Index during Test Drive



(i) Eye Tracking Results during Test Drive - 0: Looking Straight, 1: Looking Left Ahead, 2: Looking in Left Side Mirror, -1: Looking Right Ahead, -2: Looking in Right Side Mirror



(j) Risk of Accident during Test Drive

Fig. 19: Test Drive

Fig. 19a shows the path traveled by the vehicle as recorded by the high precision GPS. We can see that all lane changes and other maneuvers are visible in this mapping owing to the high precision of the GPS. Fig. 19f shows this mapping converted into vehicle coordinates and the tracking results of surrounding vehicles are drawn as well. The lane changes are even more evident here due to shrinking of the vertical scale.

Fig. 19b shows the heading angle response of the vehicle throughout the drive. It can be seen that the turns cause a change of almost 90° whereas lane changes induce a ripple in the heading response. The variance of the heading has been plotted in Fig. 19c and the vehicle speed in Fig. 19d. The variance exhibits peaks whenever a maneuver is performed. Using the classifiers learnt by supervised machine learning, the state machine is operated. The system state at each time instant is shown in Fig. 19g. It can be seen that the state sequentially increases from 'Going Straight' to 'Starting Slow Lane Change', 'Starting Fast Lane Change', 'Performing Slow Turn' and 'Performing Fast Turn' if the variance of heading keeps increasing. If the variance stops increasing then the state machine enters the branch of that corresponding maneuver.

The driver also manually recorded that 4 slow lane changes, 8 fast lane changes and 3 slow turns were performed during the drive. Exactly the same was output by the state machine. Also, 2 false slow lane change alarms were detected at 150s and 175s. These were due to slight drift of the vehicle, however the state machine was able to identify them as false alarms instead of detecting them as a maneuver, showing robustness of the system to false alarms.

The surrounding traffic tracked by the LIDAR is shown in Fig. 19f. The graph does not show the traffic w.r.t. time however the braking distance kept by the vehicle is shown in Fig. 19g which also shows the minimum braking distance required according the current speed of the vehicle. We can see that at 14s, 27s, 60s, 72s and 79s. The evolution of the driving index throughout the drive is shwon in Fig. 19h. The D.I. decreases monotonically with time if no dangerous maneuver is performed however a dangerous maneuver incurrs a penalty of 20 points. The threshold of rash driving is set at 95 points. The driver violates the threshold twice during the drive.

The eye tracking results are shown in Fig. 19i and risk of collision is plotted in Fig. 19j. The collision alarm rings at three intervals, at 14, 37 and 116s. The first alarm rings when the vehicle maintains less than the steering distance from the preceding vehicle as shown in Fig. 20a. The second alarm rings when the driver is looking in the right side mirror and slowly drifting the vehicle towards the right whereas this lane change will put him within 2 seconds of collision with a vehicle already present in that lane as shown in Fig. 20c. This scenario clearly signifies the importance of eye tracking as the collision potential was detected premature to the lane departure. The third alarm rings again when the distance is less than the steering distance as shown in Fig. 20b.

(a) Risk Alarm 1 (b) Risk Alarm 2

(c) Risk Alarm 3

Fig. 20: Situations of Collision Warning Alarm

V. Conclusion

In this paper we have proposed a multi-modal wireless sensor network which is able to model the driving behavior of the driver as well as provide early detection of risk of accidents. The emphasis has been on using low cost sensors and developing low complexity algorithms that can be run on low cost microprocessors. The system uses a low cost IMU in conjunction with a low cost GPS to determine the maneuvers performed by the car with the help of machine learning and a state machine. LIDAR sensors are used to detect motion of the surrounding traffic and a camera to detect the eye movement of the driver. The communication protocol is also proposed which ensures seamless flow of messages between related nodes. Using the information of the surrounding traffic and the maneuvers performed, the driving behavior of the driver is determined, and by using the surrounding traffic information and driver's eye tracking, risk of collision is determined.

The system is tested on a real test vehicle. Maneuver detection accuracy is 100% proving robustness to false alarms. Surrounding traffic information has been shown to benefit both driver behavior modeling and accident risk detection. Eye tracking helps to reduce the prediction time of the system and benefit in timely intervention of potential accident situations. All the algorithms ran successfully on low cost microcontrollers thus completing the objectives of the study.

References

[1] NHTSA Press Release. Traffic fatalities fall in 2014, but early estimates show 2015 trending higher, November 24, 2015.

[2] Jeffrey S Hickman and E Scott Geller. Self-management to increase safe driving among short-haul truck drivers. *Journal of Organizational Behavior Management*, 23(4):1–20, 2005.

[3] How-am-i-driving bumper stickers and online driver safety training. Available at dmvreportcard.com/How-Am-I-Driving-Safety-Bumper-Stickers.html.

[4] Derick Johnson, Mohan M Trivedi, et al. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1609–1615. IEEE, 2011.

[5] Ravi Kumar Satzoda, Sebastien Martin, Minh Van Ly, Pujitha Gunaratne, and Mohan Manubhai Trivedi. Towards automated drive analysis: A multimodal synergistic approach. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1912–1916. IEEE, 2013.

[6] Asher Bender, Gabriel Agamennoni, James R Ward, Stewart Worrall, and Eduardo M Nebot. An unsupervised approach for inferring driver behavior from naturalistic driving data.

[7] Jiang Liu, Baigen Cai, Jian Wang, and Wei ShangGuan. Gnss/ins-based vehicle lane-change estimation using imm and lane-level road map. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 148–153. IEEE, 2013.

[8] Rafael Toledo-Moreo, Miguel Zamora-Izquierdo, Antonio F Gómez-Skarmeta, et al. Multiple model based lane change prediction for road vehicles with low cost gps/imu. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 473–478. IEEE, 2007.

[9] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. Estimating driving behavior by a smartphone. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 234–239. IEEE, 2012.

[10] Johannes Paefgen, Flavius Kehr, Yudan Zhai, and Florian Michahelles. Driving behavior analysis with smartphones: insights from a controlled field study. In *Proceedings of the 11th International Conference on mobile and ubiquitous multimedia*, page 36. ACM, 2012.

[11] Vygandas Vaitkus, Paulius Lengvenis, and Gediminas Zylius. Driving style classification using long-term accelerometer information. In *Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On*, pages 641–644. IEEE, 2014.

[12] Luis M Bergasa, Daniel Almería, Jon Almazan, J Javier Yebes, and Roberto Arroyo. Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 240–245. IEEE, 2014.

[13] Minh Van Ly, Sebastien Martin, and Mohan Manubhai Trivedi. Driver classification and driving style recognition using inertial sensors. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1040–1045. IEEE, 2013.

[14] Dejan Mitrović. Reliable method for driving events recognition. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):198–205, 2005.

[15] Jérôme Maye, Rudolph Triebel, Luciano Spinello, and Roland Siegwart. Bayesian on-line learning of driving behaviors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4341–4346. IEEE, 2011.

[16] Claire D Agostino, Alexandre Saidi, Gilles Scouarnec, and Liming Chen. Learning-based driving events recognition and its application to digital roads.

[17] Pranaw Kumar, Mathias Perrollaz, Stéphanie Lefevre, and Christian Laugier. Learning-based approach for online lane change intention prediction. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 797–802. IEEE, 2013.

[18] Dominik Dorr, David Grabengiesser, and Frank Gauterin. Online driving style recognition using fuzzy logic. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1021–1026. IEEE, 2014.

[19] Takashi Bando, Kana Takenaka, Shogo Nagasaka, and Takafumi Taniguchi. Generating contextual description from driving behavioral data. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 183–189. IEEE, 2014.

[20] Anup Doshi, Brendan Morris, and Mohan Trivedi. On-road prediction of driver's intent with multimodal sensory cues. *IEEE Pervasive Computing*, (3):22–34, 2011.

[21] Massimo Canale and Stefano Malan. Analysis and classification of human driving behaviour in an urban environment*. *Cognition, Technology & Work*, 4(3):197–206, 2002.

[22] Hiren M Mandalia and Mandalia Dario D Salvucci. Using support vector machines for lane-change detection. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 1965–1969. SAGE Publications, 2005.

[23] Ryota Terada, Hiroyuki Okuda, Tatsuya Suzuki, Kazuyoshi Isaji, and Naohiko Tsuru. Multi-scale driving behavior modeling using hierarchical pwarx model. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1638–1644. IEEE, 2010.

[24] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[25] Robert Kastner, Tobias Kühnl, Jannik Fritsch, and Christian Goerick. Detection and motion estimation of moving objects based on 3d-warping. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 48–53. IEEE, 2011.

[26] Chieh-Chih Wang, Charles Thorpe, and Arne Suppe. Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 416–421. IEEE, 2003.

[27] Roman Katz, Juan Nieto, and Eduardo Nebot. Probabilistic scheme for laser based motion detection. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 161–166. IEEE, 2008.

[28] Dave Ferguson, Michael Darms, Chris Urmson, and Sascha Kolski. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1149–1154. IEEE, 2008.

[29] Anna Petrovskaya and Sebastian Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139, 2009.

[30] Christoph Mertz, Luis E Navarro-Serment, Robert MacLachlan, Paul Rybski, Aaron Steinfeld, Arne Suppe, Christopher Urmson, Nicolas Vandapel, Martial Hebert, Chuck Thorpe, et al. Moving object detection with laser scanners. *Journal of Field Robotics*, 30(1):17–43, 2013.

[31] Michael Bosse and Robert Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691, 2008.

[32] Takashi Ogawa, Hiroshi Sakai, Yasuhiro Suzuki, Kiyokazu Takagi, and Katsuhiro Morikawa. Pedestrian detection and tracking using in-vehicle lidar for automotive application. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 734–739. IEEE, 2011.

[33] Kimberly Tuck. Tilt sensing using linear accelerometers. *Freescale Semiconductor Application Note AN3107*, 2007.

[34] T Talat Ozyagcilar. Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors, freescale semiconductor application note,(2012), n. *AN4248, rev*, 3.

[35] Fawzi Nashashibi and Alexandre Bargeton. Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 847–852. IEEE, 2008.

[36] Fabian Timm and Erhardt Barth. Accurate eye centre localisation by means of gradients. *VISAPP*, 11:125–130, 2011.

[37] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[38] Jonas Jansson. Collision avoidance theory: With application to automotive collision mitigation. 2005.

[39] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

Hamad Ahmed completed B.Sc. in Electrical Engineering from University of Engineering and Technology, Lahore in 2015 with majors in Telecommunication and Electronics Engineering. Currently, he is working as a research assistant in the Signal Processing & Navigation Algorithms Group at the Department of Electrical Engineering, Lahore University of Management Sciences on the development of low cost alternatives to GPS for vehicle localization. His research interests lie in the areas of Statistical Signal Processing, Optimal Estimation Theory, Wireless Sensor Networks and Machine Learning.

PLACE PHOTO HERE

Muhammad Tahir Biography text here.