Int. J. ITS Res.
DOI 10.1007/s13177-015-0116-5

# Detection of Driving Events using Sensory Data on Smartphone

**Chalermpol Saiprasert[1] · Thunyasit Pholprasit[1] · Suttipong Thajchayapong[1]**

Received: 28 February 2014 / Revised: 14 September 2014 / Accepted: 30 June 2015
© Springer Science+Business Media New York 2015

**Abstract** In a fast-paced environment of today society, safety issue related to driving is considered a second priority in contrast to travelling from one place to another in the shortest possible time. This often leads to possible accidents. In order to reduce road traffic accidents, one domain which requires to be focused on is driving behaviour. This paper proposes three algorithms which detect driving events using motion sensors embedded on a smartphone since it is easily accessible and widely available in the market. More importantly, the proposed algorithms classify whether or not these events are aggressive based on raw data from various on board sensors on a smartphone. In addition, one of the outstanding features of the proposed algorithm is the ability to fine tune and adjust its sensitivity level to suit any given application domain appropriately. Initial experimental results reveal that the pattern matching algorithm outperforms the rule-based algorithm for driving events in both lateral and longitudinal movements where a high percentage of detection rate has been obtained for 11 out of 12 types of driving events. In addition, a trade-off between the detection rate and false alarm rate has been demonstrated under a range of algorithm settings in order to illustrate the proposed algorithm's flexibility.

✉ Chalermpol Saiprasert
chalermpol.saiprasert@nectec.or.th

Thunyasit Pholprasit
thunyasit.pholprasit@nectec.or.th

Suttipong Thajchayapong
suttipong.thajchayapong@nectec.or.th

[1] National Electronics and Computer Technology Center, 112 Phahonyothin Road, Klong Neung Klong Luang District, Pathumthani 12120, Thailand

**Keywords** Driving events detection · Smartphone · Accelerometer · Driving behaviour analysis

## 1 Introduction

In a frenzied and competitive environment in our society today, commuting from place to place in the shortest possible time seems to be a necessity. As a result, safety issues when traveling on the road are not always our first priority. Hence, aggressive driving behaviours such as fast lane change, tailgating and sudden braking which often lead to accidents are likely to occur. It has been found that when a driver is monitored and driving events are recorded the chances of aggressive and dangerous driving behaviour are reduced [1]. There are a number of commercial products available in the market using in-vehicle data recorders equipped with a wide variety of sensory devices such as GPS receiver and often a video camera [2]. Examples of these products are used in fleet management systems and taxi operators where every driver can be traced to ensure that they follow designated routes and do not violate the speed limit.

Many application domains such as logistics and intelligent transport systems benefit from this network of sensory devices [3]. Examples of these can be found in [4] where real-time driving data from controlled test crashes is stored and analysed in order to detect possible collisions and also assess the level of damage of the potential crash. Car manufacturers have taken this idea and developed an advanced driver-assistance systems (ADAS) such as collision prevention and avoidance systems [5]. At present, this can only be found in high end models as the sensors required for ADAS system are expensive making it very unlikely to be included in lower priced vehicles.

Published online: 23 July 2015

Human error is one of the three key contributing factors to road traffic accidents, the other two being vehicles capabilities and road infrastructures. The fundamental elements leading up to aggressive driving behaviours are from different driving manoeuvres or events that occur during a journey such as harsh braking and acceleration, rapid turning and rapid lane change. Therefore, it is essential to be able to detect these fundamental driving events and classify whether or not they are aggressive. As a result, we are able to recognise when a driver is engaging in behaviours that are indicative of aggressive driving, which in turn, would indicate potential crash risk.

All of the driver monitoring systems discussed so far make use of in-vehicle data recorders which possess the ability to store relevant driving data [2, 4, 5]. However, these recorders are attached to one vehicle only and are not removable to be taken off and used in other vehicles. An alternative is to replace in-vehicle data recorder with a smartphone since it is easily accessible, widely available and low cost. In addition, modern smartphone models at present are embedded with multi-sensors on-board which enables the capability similar to in-vehicle data recorders. Considering all these options, it is clear to see that smartphones are a good candidate to be deployed as a tool to be used to collect, process and analyse driving data as well as detect and classify aggressive driving behaviours in order to alert drivers when they are being reckless.

The multi-sensing capabilities of smartphones available in the market enable us to collect a rich vein of raw data. Accelerometer data provides an insight into the longitudinal and lateral movement of the phone while the on-board GPS receiver provides us with location data in terms of latitude and longitude. In the literature, smartphones have already been deployed as a tool to collect data for the analysis of driver's behaviour and external road conditions in [6] and [14]. With all the data and sensors we are able to detect vehicle's movement when a smartphone is placed inside the vehicle of interest. As a result, typical driving events such as turning left and right, braking and accelerating can be detected. It is important to detect these typical driving events as they are fundamental to the evaluation of driver behaviour. This would be highly beneficial to many application domains in the road safety perspective such as an automated advanced warning system.

This paper proposes a self-contained, vehicle-independent tool in the form of a smartphone which enables users to detect driving events. The contribution of this work is two-fold and are listed as follows:

- Three algorithms are proposed in order to detect and classify driving events all with different target applications. The first detection algorithm is a rule-based algorithm which requires only data collected from GPS receiver making it suitable for implementation on low end smartphones. The second detection algorithm deploys pattern matching technique. The third algorithm is a more robust and improved approach to the second algorithm. Algorithms two and three utilise accelerometer sensor which make them more suited for a mid range smartphone and above.
- The second contribution of this paper is the comparison of the three proposed algorithms in order to assess their features and performances.

This paper is an extension of our conference paper at the $20^{th}$ ITS World Congress [7]. Additional material has been included in order to create a more in-depth research paper. This includes a fully customisable algorithm for a more robust driving event detection and additional experiments and analysis for the assessment of the proposed algorithms.

The paper is organised as follows. Related work is discussed in section two of this paper. The overall framework of this paper is described in section three. The fourth section describes the three proposed algorithms and the theories behind them. Section five discusses the experimental setup. Experimental results and analysis are presented in the sixth section. Finally, the last section concludes this paper.

## 2 Related Work

In the literature, the work on driver monitoring system can be classified into two main categories which are in-vehicle data recorder based and smartphone based. This section discusses existing work in the two main categories as follows.

### 2.1 Driver Monitoring System using In-vehicle Data Recorder

At present, in vehicle data recorders with the ability to record vehicle's data during a trip are becoming more common in the field of transportation. This is especially true in vehicles used for public transportation and fleet and logistics where it is crucial to be able to track and identify where the vehicle is in real-time. A number of examples exist in the literature with regards to utilising in-vehicle data recorders and the data they collect to evaluate driving behaviour. Lotan and Toledo proposed a system called DriveDiagnostic in order to record and analyse driver behaviour for crash and pre-crash events [8]. In addition, a safety index is calculated based on the collected data. However, there is no clear explanation of how the index was derived.

The data from these data recorders can provide us with very useful information once it is properly analysed. An example of this is the work in [9], where a correlation of car accidents and the risk index obtained from collected driving data is established. Ueyama et al collected and monitored pre and post accidents data from data recorders installed in vehicles in Japan [10]. Their findings revealed that there is a relationship between driving behaviour and the likelihood of accidents.

As well as longitudinal data recorded using the in vehicle recorders, video and voice technology have also been considered. Arai et al incorporated video sensor to the already existing data from the recorder in order to reconstruct accidents in greater detail [11]. In addition to visual data, Green et al introduced the use of audio sensors including horns, clashing metal sounds and squealing tires for the analysis of crashes at an intersection [12]. The objective of [12] is to detect patterns of accidents for relevant improvements.

Han and Yang's work discused an analysis of driver's characteristic for the detection of dangerous driving using data from automobile data recorder [13]. Dangerous driving actions that lead to reckless driving were clearly defined with detailed description of the algorithms to detect each event. However these data and analysis were not transformed into an index or a rating in order to provide a measure of driver's behaviour.

### 2.2 Driver Monitoring System using Smartphone

Recent technological advancements in smartphones capabilities coupled with an increasing smartphone adoption rates worldwide has initiated the development of many new ITS related applications. The work in [15] proposed a low cost lane departure warning system implemented on a smartphone in order to bring advanced technology to a device which is widely available. The main idea of the work in [15] is to apply image processing techniques to the images captured from smartphone camera. In addition, their proposed algorithm is optimised such that it can tolerate low quality images and is robust to run on smartphones with lower processing power.

Mohan et al. proposed a system where smartphones are utilised as a means to monitor road and traffic conditions [16]. This was achieved by using sensors onboard smartphones such as accelerometer and GPS sensor to detect potholes, bumps as well as vehicles braking and honking. The system has been implemented and tested where promising results in terms of the effectiveness of sensing functions have been reported. Similar to [16], the approach proposed in [6] deploys a smartphone app which collects data from multi-sensors onboard to analyse road conditions obtaining at the same time high accuracy results in classifying different road defects, vehicle condition and driving events. Compared to the work in [6], the scope of this paper is particularly focused on investigating driving events where algorithms are developed to identify aggressive driving behaivours.

Johnson and Trivedi proposed an approach in order to classify different driving styles based on data collected from smartphones [14]. In their approach, driving styles can be in the form of normal, aggressive and very aggressive. The results from their work reveal that various sensors on smartphones can provide good source of information for an accurate measure and classification of different driving styles. However, the algorithm proposed in [14] relies on accelerometer and gyroscope where the latter is only available on high-end smartphones. Unlike [14], the algorithms proposed in this paper use measurements from only accelerometer so they can also be deployed with low-end smartphones.

It is noted that sudden acceleration and sudden brake events are explicitly omitted in [14] with the main reason that they can be measured directly using GPS signal. However, GPS signal itself can be lost in certain areas (e.g. in city with tall buildings or under bridges) which can subsequently undermine the detection capability of these two important events. Therefore, to address this issue, the proposed algorithms are designed to detect sudden acceleration and sudden brake events using accelerometer.

Furthermore, the work presented in [17] discusses about the use of smartphones to report and detect car accidents in real-time. Similar to [16], the approach in [17] also utilises GPS receiver and accelerometer data in order to detect car accidents. While the algorithm in [17] is very useful in reporting an accident once it takes place, the algorithms proposed in this paper proactively detects aggressive driving behaviours which can be used to prevent or minimize the chance of accidents. Hence, it would be interesting to combine the algorithms in this paper with the one proposed in [17] to completely investigate pre-accident conditions, accident sequence and post-accident conditions.

## 3 Platform Overview

This section discusses the high level overview of the platform that we deploy for the detection of driving events. An Android based smartphone is used as our target platform to collect raw data from two on-board sensors, 3-axis accelerometer and GPS receiver.

### 3.1 Measurement of Raw Data

In this paper, two sensors from a smartphone are considered. Firstly, the 3-axis accelerometer measures the force of acceleration whether caused by the phone's movement or gravity. The three axes correspond to lateral, longitudinal and vertical accelerations. In this work we are only interested in

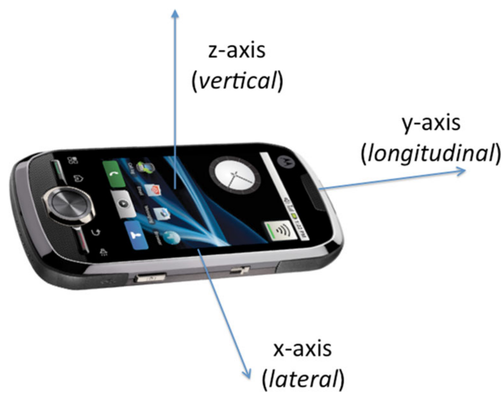



**Fig. 1** 3-axis Acelerometer

movements along the lateral and longitudinal axes which refers to side to side movement and forward and backward movement respectively. In real-world scenarios lateral acceleration or side to side movements represent driving events such as turning left and right and lane change while longitudinal acceleration corresponds to vehicle braking and accelerating.

Figure 1 shows a typical smartphone with relevant axes in accordance with the measurement from accelerometer. The lateral movement is denoted by the x-axis while the longitudinal movement is denoted by the y-axis.

A GPS receiver which provides positioning and speed data of the vehicle that the smartphone is attached to. Overall, accelerometer data is sampled at a rate of 5Hz where one sample is recorded every 200ms. Data from GPS receiver is sampled at 1Hz.

### 3.2 Vehicle Movement

Based on the measurement of raw data in the previous section, fundamental driving events can be established and is summarised in Table 1. It can be seen from the table that 12 types of driving events are considered in total, 8 in the lateral movement domain and 4 in the longitudinal domain. These include standard and aggressive events.

**Table 1** Driving events

| Lateral | Longitudinal |
|---|---|
| Right/Left turn (normal and aggressive) | Braking (normal and aggressive) |
| Right/Left lane change (normal and aggressive) | Acceleration (normal and aggressive) |

## 4 Proposed Algorithms

In this paper, three algorithms for the detection of driving events are proposed. The main difference between the first and second algorithms are the source of raw data from the sensors onboard a smartphone to be processed and the theories behind each algorithm. The third algorithm implements an additional step for a trigger to initiate the algorithm when a certain condition is met. Details of the three proposed algorithms are given in the following subsections.

### 4.1 Rule-based Algorithm

The first algorithm proposed in this paper takes a stream of raw data from a GPS receiver onboard a smartphone as an input. The following data items are recorded.

- $Speed$: a measurement of instantaneous speed at which the vehicle is travelling at for each data sample.
- $Position$: a geographical location of the vehicle described by the latitude and longitude for each instance of a data sample.
- $Heading$: a measure of direction of travel in degrees East of True North.

Data is measured and recorded at a sampling rate of one sample per second. The fine sampling frequency of 1Hz creates a time series of measurement in speed and position over a period of time. Under this algorithm, the recorded data from GPS receiver is used to detect and classify our driving events of interest for both the lateral and longitudinal movements.

Figure 2 shows the flow of the Rule-based Algorithm. An initial data preprocessing is required at the start of the algorithm as raw data from GPS receiver has a tendency to be incomplete with missing data in certain time periods of the time series due to weak GPS signals. A simple linear interpolation is used to treat this incomplete data set. At this stage the time series data is ready to be processed by the algorithm. In order to detect and classify each of the driving events thresholding technique is utilised with the constraints according to Table 2.

It can be seen from Table 2 that aggressive turning occurs when the rate of change of vehicle's heading ($\delta Heading$) is above 30 degree/s. In addition, a driving event is classified as sudden acceleration and braking when the absolute value of acceleration is greater than 0.3G where G denotes the acceleration due to gravity ($9.8m/s^2$) [18–20]. This thresholding constraints are deployed in the classification algorithm to classify the detected driving events. Finally, the algorithm outputs the predicted driving event. This process

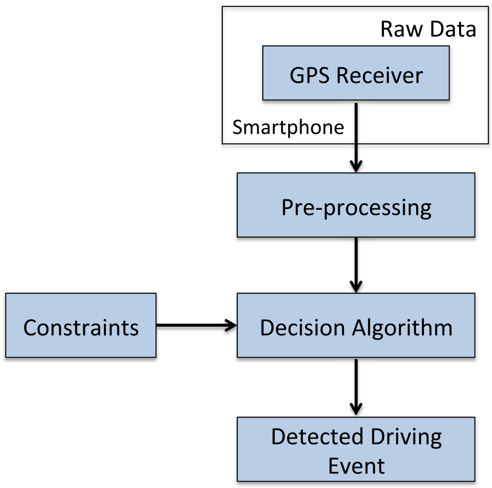



**Fig. 2** Rule-based algorithm

is repeated for the duration of the journey resulting in a series of detected events.

In terms of the implementation on smartphone, the rule-based algorithm can be deployed on the majority of smartphones in the market as only GPS receiver is required. Hence, it covers a wide range of models from the lower end of the price range to high end models. However, GPS signals require a good line-of-sight between transmitter and receiver which means that signal strength is not guaranteed to be at an acceptable level depending on weather conditions, obstructions such as trees, high-rise buildings and wires. Therefore, raw data from GPS receiver is likely to contain incomplete data set resulting in missing data points and also inaccurate locations.

### 4.2 Pattern Matching Algorithm

Unlike the rule-based algorithm, the pattern matching algorithm makes use of raw data from accelerometer sensor onboard a smartphone. One of the advantages over rule-based algorithm which utilises data from GPS receiver is that accelerometer sensor is independent of external factors that create the limitation of missing data set as all measurements are recorded within the sensors onboard the phone. An example of this is that the lost of GPS signals due to no line-of-sight will result in data loss. The accelerometer provides measurements of acceleration of the vehicle that the smartphone is attached to in 3-axis domain, vertical, longitudinal and lateral as shown previously in Fig. 1. Data from accelerometer sensor is recorded at a rate of 5Hz in this work in order to form a time series of acceleration of the smartphone.

The proposed pattern matching algorithm is based on the Dynamic Time Warping (DTW) technique. Dynamic Time Warping was originally implemented in order to perform speech recognition by Sakoe and Chiba [21]. It has then been utilised extensively in the field of computer sciences such as the approach in [22] where DTW was used to find patterns in time series. An approach in [23] utilises DTW to predict short-term traffic congestion based on the speed of the probe vehicles.

In general, Dynamic Time Warping provides a similarity measure between two signals, namely the incoming and the reference signals. The main feature of DTW is that it allows for stretched and compressed portions of the two signals to be compared by compensating for length differences in the two signals while taking into account of the non-linearity of the length differences between the incoming signal and the reference signal. This feature is not possible with a traditional pairwise comparison between the two signals using the Euclidean distance. In this paper, the concept of Dynamic Time Warping will be used for the detection of driving events. A brief description of the DTW algorithm is given below.

Consider two time series $X$ and $Y$ with length $n$ and $m$ respectively, where each time series is represented by $X = \{x_1, x_2, ..., x_i, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_j, ..., y_m\}$. An $n \times m$ matrix is constructed with time series X and Y on the top and left sides of the grid respectively as shown in Fig. 3 . Each element $(i, j)$ of the matrix contains the Euclidean distance between the points $x_i$ and $y_j$ on the two corresponding time series, where

$$d(x_i, y_j) = (x_i - y_j)^2. \tag{1}$$

**Table 2** Thresholding constraints

| Driving Events | Monitored Signals | Constraints |
|---|---|---|
| Right/Left Turn | Heading (degree) | $\delta Heading$ = 20-30 deg/s |
| Aggressive Right/Left Turn | Heading (degree) | $\delta Heading$ > 30 deg/s |
| Braking/Acceleration | Acceleration $(m/s^2)$ | $\lvert Acceleration \rvert$ < 0.3G |
| Sudden Braking/Acceleration | Acceleration $(m/s^2)$ | $\lvert Acceleration \rvert$ 0.3G |

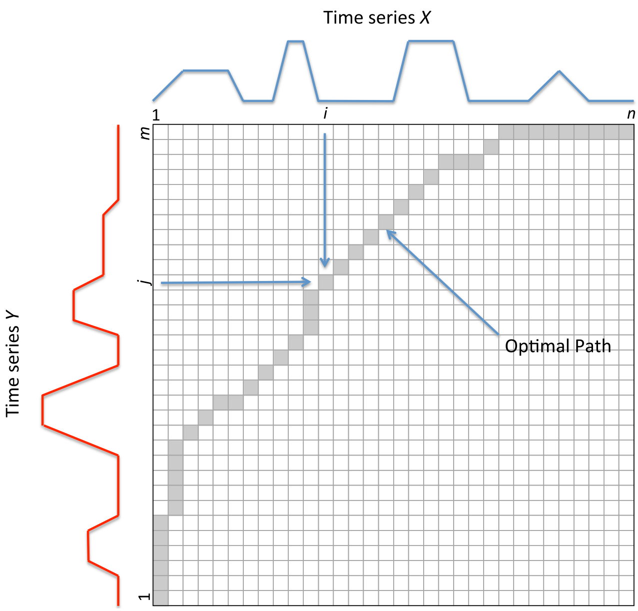



**Fig. 3** $n \times m$ grid for DTW calculation

The lower the value of $d$ the closer the two points are to each other. Essentially, DTW tries to find an optimal alignment of the two time series. This idea is applied in this work where the time series of the pre-recorded template is aligned with the raw data. The next step of DTW is to identify a warping path $W$ which consists of the minimum distances between the two points on the time series where the $k^{(th)}$ element of W is denoted by $w_k = (i, j)_k$ [24]. The next stage is to sum these minimum distances along the warping path $W$ in order to obtain the cost function $C$ as described in Eq. 2.

$$C(X, Y) = \sum_{k=1}^{K} w_k(x_{nk}, y_{mk}). \tag{2}$$

Finally, the reference signal with the lowest total cost $C$ is the best match to the given incoming signal.

Figure 4 illustrates the proposed pattern matching algorithm to detect driving events based on the use of DTW algorithm. The algorithm is divided into three main stages.

1) *Pre-Processing:*
Data pre-processing is an initial stage of the pattern matching algorithm. Raw data collected from the accelerometer sensor is pre-processed in order to smooth out the effect of unwanted noise in the signals. In this paper a simple moving average is utilised to achieve that goal.

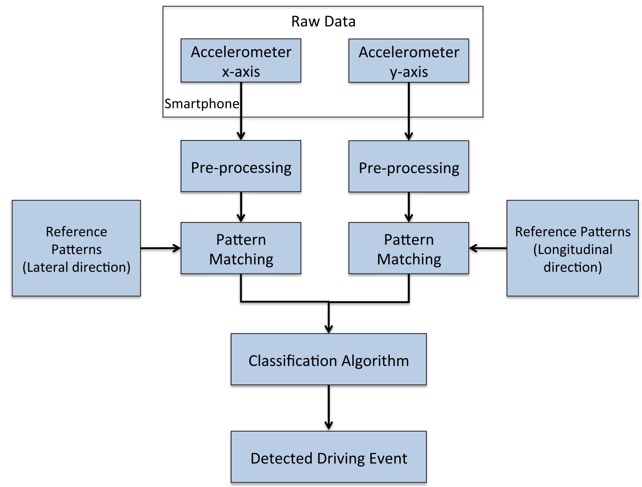
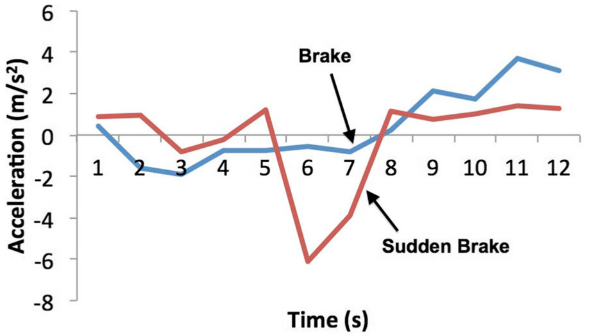


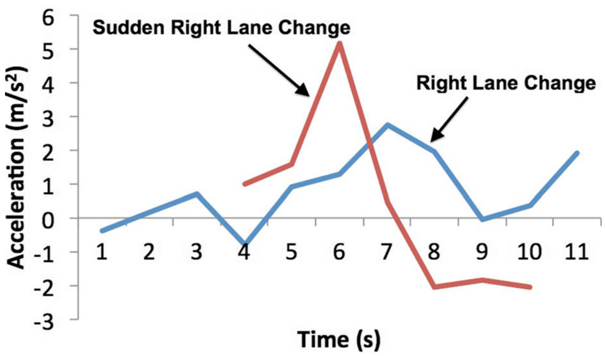

**Fig. 4** Pattern matching algorithm

2) *Pattern Matching:*
This is the stage where DTW algorithm is deployed to find the best match using a given reference pattern for all driving events in Table 1. Some of the reference driving patterns are illustrated in Fig. 5. It can be seen that there is an apparent dissimilarity in the shape of the waveforms of an ordinary driving event, in a green line, and an aggressive driving event, in a red line. Aggressive or sudden driving events tend to possess a large change in acceleration values. The essential goal in this stage of the algorithm is to use the reference driving patterns to find the best match for the incoming driving data.
In order to generate appropriate reference patterns for each driving event, a training data set is obtained through a real-world experiment for our pattern matching algorithm. These reference patterns are then used as a template to match the incoming signals from accelerometer sensor in the test data set. In this paper, 70 % of data samples are utilised as training data, resulting in 30 % to be used as test data set. At the completion of the algorithm, a total cost of the alignment path $C$ is obtained for all of the templates selected to cover all 12 events.

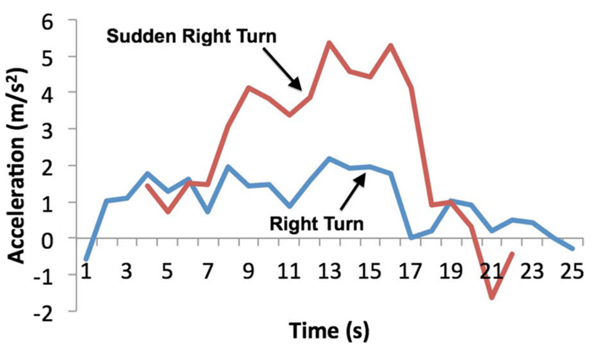
3) *Classification Algorithm:*
At this point the algorithm will decide which of the reference patterns is the best match to the incoming signal. Specific constraints are set in accordance to each of the 13 different driving events. One of the constraints is the similarity score produced by the DTW algorithm, where a score of zero would indicate two patterns being exactly alike. The fact that driving events are categorised into




(a) Acceleration data from y-axis of the accelerometer inferring longitudinal movements, braking and sudden braking.




(b) Acceleration data from x-axis of the accelerometer inferring lateral movements, right lane change and sudden right lane change.




(c) Acceleration data from x-axis of the accelerometer inferring lateral movements, right turn and sudden right turn

**Fig. 5** Reference patterns for driving event detection. **a** Acceleration data from y-axis of the accelerometer inferring longitudinal movements, braking and sudden braking. **b** Acceleration data from x-axis of the accelerometer inferring lateral movements, right lane change and sudden right lane change. **c** Acceleration data from x-axis of the accelerometer inferring lateral movements, right turn and sudden right turn

two domains, lateral and longitudinal movements as described in Table 1 means that there are two pools of events to select depending on the source of incoming signals.

### 4.3 Self Triggered Pattern Matching Algorithm

An extended version of the Pattern Matching Algorithm is proposed for a more robust and efficient execution of the algorithm. The algorithm in section *B* is executed for every new input data points picked up by the accelerometer. This can lead to high computational power required which is very crucial for smartphone implementation. However, the Self Triggered version does not operate in that manner. Here, the standard deviation of the current window of data points to be processed, denoted by $\sigma_w$, is examined. If $\sigma_w$ is greater than a threshold, denoted by $\sigma_{th}$ then the Self Triggered Pattern Matching Algorithm is automatically executed. A physical interpretation of this change in standard deviation of the acceleration data is that the Self Triggered Pattern Matching Algorithm will only be executed when the spread of the acceleration data is sufficiently high which is reflected by higher value of $\sigma_w$. This would indicate an occurrence of a driving event in comparison to a vehicle travelling in a steady state. On the other hand, the Self Triggered Pattern Matching Algorithm will remain idle if $\sigma_w$ is lower than $\sigma_{th}$. Thus, the sensitivity level of the proposed algorithm is fully customisable. This feature of our algorithm enables potential deployment on various application domain as the algorithm can be fully customised to cater for different requirements such as different road types and different time of day. Moreover, this additional feature will reduce computational power and, in turn, lower power consumption when implemented on a smartphone.

It is important to note that the threshold standard deviation, $\sigma_{th}$, is not constant and is data dependent as every new data point acquired is taken into account. A time series of accelerometer data with length $n$ is represented by $A = \{a_1, a_2, a_3, ..., a_i, ..., a_n\}$. For this time series $A$, the threshold standard deviation $\sigma_{th}$ is computed as a rolling standard deviation from the start of the data set, $a_1$, to the current data point, $a_i$. Therefore, when a new data point is acquired the value of $\sigma_{th}$ is recomputed using (3).

$$\sigma_{th_i} = \sqrt{\frac{(i-2)\sigma^2_{th_{i-1}} + (i-1)(\mu_{i-1} - \mu_i)^2 + (a_i - \mu_i)^2}{i-1}}, \tag{3}$$

where $i$ denotes the current data point acquired from the accelerometer sensor, $\sigma_{th_i}$ denotes the standard deviation of the accelerometer data from the start of the dataset up to the $i^{th}$ point, $\mu_i$ denotes the mean of the accelerometer data from the start of the dataset up to the $i^{th}$ point and $a_i$ denotes the current value of the accelerometer sensor. $i-1$ refers to the previous data point.

Figure 6 illustrates a time series of acceleration data. From visual inspection, it can be seen that some parts of the data fluctuate more than the other. This fluctuation is likely to be caused by one of the 13 driving events. From the plot in Fig. 6, the algorithm is triggered on three occasions as indicated in the blue boxes as $\sigma_w$ is greater than $\sigma_{th}$.

The process of evaluating standard deviation of the data is performed before pattern matching stage in order to ensure that the number of false alarm is minimised while maximising the detection rate. During the execution, the process of this algorithm is the same as that of Pattern Matching Algorithm. With this technique, it is expected that the algorithm will be required to execute less frequently than the algorithm in section $B$ for the same dataset.

## 5 Experimental Setup

In this experiment raw data was collected using a single driver in one vehicle, 2010 Toyota Vigo pick-up truck. The reason that this was selected as a test vehicle is that pick-up trucks are the second most accident prone on the road in Thailand behind motorcycles. This makes it the most risky amongst vehicles with four wheels. Overall, approximately 120 driving events in urban and rural road environments were recorded. The route chosen was approximately 40km long, from central Bangkok to the outskirt on the north west of the city. This particular route consists of freeway section as well as arterial road section with different characteristics and features that are suited for our experiments. There are a number of turns and intersections to perform driving events of interest such as sudden turn and sudden lane changes. Data was collected using Android-based smartphone with a tailored made application. During a journey, the test driver is instructed to drive normally and also performing dangerous and sudden movements alternately. Driving events that occur during a journey are manually marked in real-time by team of traffic engineering researchers with corresponding timestamps being noted in order to label our ground truth data.

Two metrics will be used in order to assess the effect of varying the threshold of standard deviation on the performance of driving event detection. These are the detection rate (DR) and the false alarm rate (FAR) which are defined in Eq. 4 and 5 respectively.

$$DR = \frac{Number\ of\ Driving\ Events\ Detected}{Total\ Number\ of\ Driving\ Events} \times 100, \quad (4)$$

where the numerator is the number of driving events detected by the algorithm and the denominator is the total number of driving events indicated by a team of traffic engineering researchers.

$$FAR = \frac{Number\ of\ Alarms\ not\ in\ Specified\ Duration}{Total\ Number\ of\ Alarms} \times 100, \quad (5)$$

where FAR is expressed as false alarm rate per second in %. At the time of writing, no other work in the literature has reported FAR for the detection of driving events using sensory data from smartphone.

## 6 Analysis of Results

### 6.1 Comparison of Rule-based and Pattern Matching Algorithms

Figures 7 and 8 show the confusion matrices of the proposed rule-based and pattern matching algorithms for driving events detection. It can be seen clearly from the figures that the Pattern Matching Algorithm performs significantly better than the Rule-based Algorithm as most of the detections

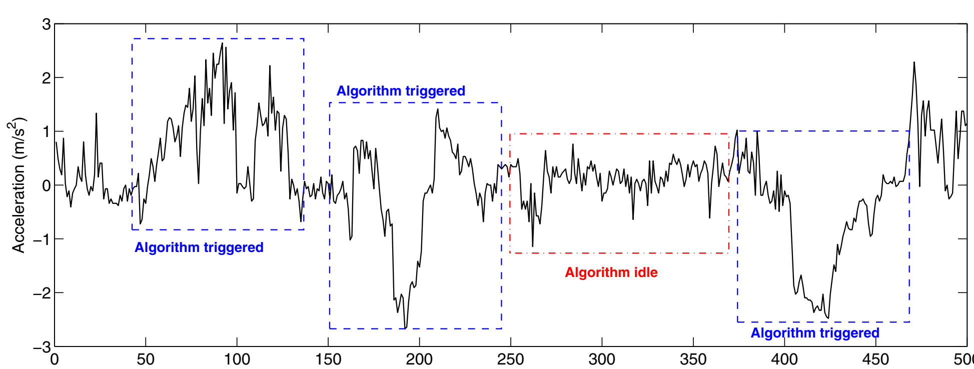



**Fig. 6** Thresholding of standard deviation

| | Driving Events Detected by Rule-based Algorithm | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Event | B | SB | A | SA | L | SL | R | SR | Missed Detection |
| B | 8 (44.44) | | 1 (5.56) | 1 (5.56) | | | | 1 (5.56) | 7 (38.89) |
| SB | | 0 (0.00) | 5 (62.50) | 1 (12.50) | | | | | 2 (25.00) |
| A | 4 (14.29) | | 6 (21.43) | 3 (10.71) | | | | | 15 (53.57) |
| SA | 1 (50.00) | | | 0 (50.00) | | | | | 1 (50.00) |
| L | | | 1 (25.00) | | 0 (0.00) | | | | 3 (75.00) |
| SL | 1 (20.00) | | 1 (20.00) | | 1 (20.00) | 1 (20.00) | | | 1 (20.00) |
| R | | | | | | | 0 (0.00) | | 1 (100.00) |
| SR | | | | | 1 (25.00) | | | 1 (25.00) | 2 (50.00) |

Ground Truth

**Fig. 7** Confusion matrix for rule-based algorithm. 12 types of driving events have been observed. These are brake (B), sudden brake (SB), accelerate (A), sudden accelerate (SA), left turn (L), sudden left turn (SL), right turn (R), sudden right turn (SR), lane change left (CL), sudden lane change left (SCL), lane change right (CR) and sudden lane change right (SCR). The *diagonal* entries highlighted in *green* are the number of events correctly detected by the algorithms according to ground truth data. The numbers in each entry in the table denotes the number of occurrence of each event detected while the number inside the parenthesis denotes the detection rate (DR). The cells highlighted in *red* denote the events which have been incorrectly detected as they are from completely different domains

from the algorithm are in the green diagonal elements. 12 types of driving events have been observed. These are brake (B), sudden brake (SB), accelerate (A), sudden accelerate (SA), left turn (L), sudden left turn (SL), right turn (R), sudden right turn (SR), lane change left (CL), sudden lane change left (SCL), lane change right (CR) and sudden lane change right (SCR). In total there are approximately 83 usable driving events noted by the team of traffic engineering researchers.

From the figures, the diagonal entries highlighted in green are the number of events that have been correctly detected by the algorithms according to ground truth data. For a perfect detection by the algorithm, all the elements in the matrix will be blank except the diagonal elements where all of the detections will occur. The numbers in each entry in the table denotes the number of occurrence of each event detected while the number inside the parenthesis denotes the detection rate (DR). The cells highlighted in red denote

| | Driving Events Detected by Pattern Matcing Algorithm | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Event | B | SB | A | SA | L | SL | R | SR | CL | SCL | CR | SCR | Missed Detection |
| B | 17 (94.44) | 1 (5.56) | | | | | | | | | | | 0 |
| SB | 4 (50.00) | 3 (37.50) | 1 (12.50) | | | | | | | | | | 0 |
| A | | | 28 (100) | | | | | | | | | | 0 |
| SA | | | 1 (50.00) | 1 (50.00) | | | | | | | | | 0 |
| L | | | | | 4 (100.00) | | | | | | | | 0 |
| SL | | | | | | 4 (80.00) | | | | 1 (20.00) | | | 0 |
| R | | | | | | | 1 (100.00) | | | | | | 0 |
| SR | | | | | | | | 3 (75.00) | | | | 1 (25.00) | 0 |
| CL | | | | | | | | | 5 (100.00) | | | | 0 |
| SCL | | | | | | | | | 1 (50.00) | 1 (50.00) | | | 0 |
| CR | | | | | | | 2 (50.00) | | | | 2 (50.00) | | 0 |
| SCR | | | | | | | | | | | 1 (50.00) | 1 (50.00) | 0 |

Ground Truth

**Fig. 8** Confusion Matrix for Pattern Matching Algorithm. 12 types of driving events have been observed. These are brake (B), sudden brake (SB), accelerate (A), sudden accelerate (SA), left turn (L), sudden left turn (SL), right turn (R), sudden right turn (SR), lane change left (CL), sudden lane change left (SCL), lane change right (CR) and sudden lane change right (SCR). The *diagonal* entries highlighted in *green* are the number of events correctly detected by the algorithms according to ground truth data. The numbers in each entry in the table denotes the number of occurrence of each event detected while the number inside the parenthesis denotes the detection rate (DR). The cells highlighted in *red* denote the events which have been incorrectly detected as they are from completely different domains the events which have been incorrectly detected as they are from completely different domains. For example, from Fig. 7 a brake event has been detected 4 times instead of an acceleration event. The parts that are highlighted in dark grey represent a scenario which should not occur theoretically as the algorithms should not detect longitudinal events when the vehicle is moving in the lateral direction and vice versa. For example, a right turn event was detected by the algorithm when the actual event was a sudden acceleration.

These are initial results from our research and it can clearly be seen that the rule-based algorithm based on raw data from GPS receiver does not do very well in terms of driving event detection in the lateral directions due to very low detection rates while a slightly better detection rates were reported in the longitudinal direction. The misdetection could be caused by inconsistency in the GPS signal as incorrect positioning as well as incorrect speed and heading values are likely to occur. More importantly, the rule-based algorithm is only able to detect 8 types of driving events in comparison to 12 types of events being detected by the pattern matching algorithm. The 4 types of events that are missing are the 4 lane change events, CL, CR, SCL and SCR. A possible reason for this failure to detect the 4 lane change events is that the sampling rate of GPS receiver on the smartphone of 1Hz might be insufficient to pick up the positional changes during the occurrence of the events. Lane change events occur in a very short space of time where a sampling rate higher than 1Hz is required to detect positional changes.

On the other hand the pattern matching algorithm using accelerometer data performs significantly better than the rule-based algorithm with high percentages of detection rate of events being correctly identified across all driving events in both lateral and longitudinal axes. The detection rate of the pattern matching algorithm range from 37.5 % upto 100 % with 11 out of 12 types of driving events achieving above 50 % detection rate. The only exception is the sudden brake event with 37.5 % detection rate reported. From Fig. 8 it can be seen that some of these sudden brake events have been detected as a brake event. One possible reason for this misdetection is the fact that the change in acceleration value might not be high enough for the algorithm to detect it as a sudden brake event. This could be improved by re-training the algorithm to recognise more sudden brake event patterns. In addition, only one detection has been identified in the incorrect driving domain indicated in a cell highlighted in red and no lateral-longitudinal cross detection have been reported. For all driving event types, a small number of events have been incorrectly detected. However, these detections appear within the same driving event domain. For example a sudden lane change left detected instead of a normal lane change left or a lane change right detected instead of a right turn.

An equivalent version of the confusion matrix for the Self-Triggered Pattern Matching Algorithm is discussed in the next section where results are obtained for different threshold values $\sigma_{th}$.

### 6.2 Effect of Self-Triggering Technique

In order to evaluate the effect of the self triggering technique the value of $\sigma_{th}$ is varied for a range of weightings. Low values of $\sigma_{th}$ signifies high sensitivity level for the algorithm as slight changes in the acceleration data picked up by the accelerometer sensor will trigger the algorithm to execute. On the other hand, high values of $\sigma_{th}$ signifies low sensitivity level for the algorithm as acceleration data would be required to be diverse and spread out to trigger the algorithm. More importantly, this illustrates one of the key features of the algorithm as the sensitivity level can be fine tuned to appropriately match any given application domain. In this experiment, an execution time is recorded for each parameter settings. In this context, the execution time is defined as the time taken for the algorithm to process a time series of driving dataset containing approximately 16,000 data points. Note that this is the same dataset which is used in all experiments in this paper.

Figure 9 is an illustration of the execution time of the algorithms for our dataset with approximately 16,000 data points. It can be seen from the plot that as the value of $\sigma_{th}$ increases the execution time of the algorithm decreases linearly. This demonstrates that the execution time of the algorithm can be controlled by selecting the required weighting for $\sigma_{th}$ depending on the application. As far as smartphone implementation is concerned, execution time infers resource usage on smartphone which includes CPU, memory and

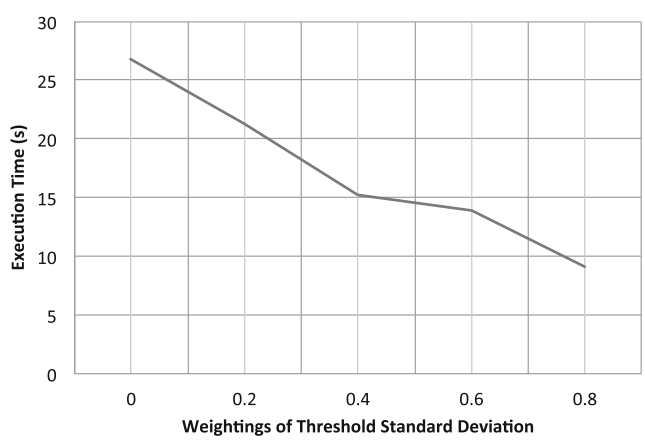



**Fig. 9** Algorithm execution time over a range of $\sigma_{th}$ weightings

| | | $0^*\sigma_{th}$ | $0.2^*\sigma_{th}$ | $0.4^*\sigma_{th}$ | $0.6^*\sigma_{th}$ | $0.8^*\sigma_{th}$ |
|---|---|---|---|---|---|---|
| B | DR (%) | 94.44 | 88.89 | 38.89 | 27.78 | 16.67 |
| | FAR (% per second) | 26.94 | 21.91 | 18.05 | 13.70 | 9.53 |
| SB | DR (%) | 37.50 | 37.50 | 0.00 | 0.00 | 0.00 |
| | FAR (% per second) | 2.21 | 1.58 | 1.46 | 1.39 | 1.09 |
| A | DR (%) | 100.00 | 75.00 | 53.57 | 25.00 | 25.00 |
| | FAR (% per second) | 27.17 | 22.44 | 20.30 | 17.11 | 11.71 |
| SA | DR (%) | 50.00 | 0.00 | 0.00 | 0.00 | 50.00 |
| | FAR (% per second) | 0.56 | 0.49 | 0.45 | 0.41 | 0.34 |
| L | DR (%) | 100.00 | 50.00 | 25.00 | 25.00 | 0.00 |
| | FAR (% per second) | 12.76 | 3.68 | 2.40 | 1.88 | 1.61 |
| SL | DR (%) | 80.00 | 75.00 | 50.00 | 0.00 | 0.00 |
| | FAR (% per second) | 13.06 | 0.04 | 0.04 | 0.04 | 0.00 |
| R | DR (%) | 100.00 | 100.00 | 100.00 | 0.00 | 0.00 |
| | FAR (% per second) | 12.05 | 3.86 | 3.19 | 2.66 | 1.76 |
| SR | DR (%) | 75.00 | 75.00 | 50.00 | 0.00 | 0.00 |
| | FAR (% per second) | 0.56 | 0.30 | 0.41 | 0.41 | 0.53 |
| CL | DR (%) | 100.00 | 100.00 | 100.00 | 80.00 | 80.00 |
| | FAR (% per second) | 13.13 | 9.42 | 8.82 | 7.50 | 5.89 |
| SCL | DR (%) | 50.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | FAR (% per second) | 1.39 | 0.41 | 0.26 | 0.26 | 0.26 |
| CR | DR (%) | 50.00 | 25.00 | 0.00 | 0.00 | 0.00 |
| | FAR (% per second) | 6.12 | 3.00 | 2.59 | 2.10 | 1.28 |
| SCR | DR (%) | 50.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | FAR (% per second) | 0.71 | 0.30 | 0.23 | 0.19 | 0.11 |

**Fig. 10** Detection rate and false alarm rate under different algorithm settings

battery. From the plot, the execution time can be reduced from approximately 21s to 9s by changing the weighting of $\sigma_{th}$ from 0.2 to 0.8. This subsequently indicates that the resource usage on smartphone can be reduced by approximately 57 %.

Figure 10 illustrates the table of detection rate and false alarm rate for each of the driving event over a range of different weight settings for $\sigma_{th}$. The definitions of the terms detection rate (DR) and false alarm rate (FAR) are given in equations (4) and (5) respectively in section V of this paper. From the analysis of raw data, false alarms occur when a portion of accelerometer data experiences high volatility and sudden rise and/or fall of data points created unintentional patterns which match the reference patterns in our database. The table in Fig. 10 reveals that, as expected, the FAR decreases as $\sigma_{th}$ increases. In other words, unintentional patterns are filtered out by higher values of $\sigma_{th}$ resulting in low FAR. On the other hand, increasing the value of $\sigma_{th}$ also decreases the detection rate. This is caused by some parts of the raw accelerometer collected during the occurrence of a driving event were filtered out due to insufficient volatility in the actual data points. Consequently, the algorithm was not triggered resulting in a misdetection.

The most important finding from the table in Fig. 10 is that there is a trade-off between the detection rate (DR) and the false alarm rate (FAR) meaning that improving one would worsen the other. Therefore it is up to the system designer to specify the appropriate values of DR and FAR depending on the application domains of interest. Example use cases are for drivers who do not wish to be disturbed often while driving for the detection of aggressive driving events may have a high $\sigma_{th}$ for a less sensitive detection algorithm. Another example is for a low $\sigma_{th}$ to be deployed in a scenario where a fleet vehicle is carrying fragile goods to make the detection algorithm very sensitive to slight vehicle movement.

## 7 Conclusion

This paper proposes three algorithms which detect driving events using data collected from smartphone sensors such as GPS receiver and accelerometer sensor conveniently without the aid of external sensors or hardware. More importantly, the proposed algorithms classify whether or not these driving events are aggressive. All three algorithms has different requirements making them suitable for different types of smartphone implementation. The first algorithm deploys a thresholding technique to detect and classify driving events from data collected from GPS receiver. Due to the minimum requirement of just a GPS receiver, this can be deployed on low end entry level smartphones that do not contain movement sensors. On the other hand, the second algorithm implements a pattern matching algorithm analysing time series data from accelerometer sensor. The third algorithm adds further enhancement to the second algorithm by using the volatility in the acceleration data as a self triggering technique. The second and third algorithms provide the ability to detect more types of driving events but they can only be deployed on higher end smartphones which are equipped with accelerometer sensor. One of the outstanding features of the third algorithm is the ability to fine tune and adjust its sensitivity level. The main benefit of this feature is that it enables the third algorithm to be fully customisable and can be modified to suit different requirements for various application domains.

Initial experimental results reveal that the pattern matching algorithm outperforms the rule-based algorithm for driving events in both lateral and longitudinal movements as high detection rates ranging from 50 to 100 % have been reported for 11 out of 12 types of driving events. In addition, an experiment has been conducted to demonstrate a trade-off between detection rate and false alarm rate using the proposed self-triggered pattern matching algorithm. It has been found that by adjusting the algorithm settings, the sensitivity of the algorithm can be controlled. This will be beneficial for applications with different requirements and settings.

As a future work, a more generic patterns to represent each of the 12 types of driving events can be formed to be utilised in the pattern matching algorithm which can potentially lead to an unsupervised pattern matching algorithm.

## References


1. Hickman, J.S., Geller, E.S.: Self-management to increase safe driving among short-haul truck drivers. J. Organ. Behav. Manag. (2005)
2. DriveCam, The Driver Science Company, http://www.drivecam.com, accessed December 2012
3. Amin, S., Andrews, S., Apte, S., Arnold, J., Ban, J., Benko, M., Bayen, R.M., Chiou, B., Claudel, C., Claudel, C., Dodson, T., Elhamshary, O., Flens-Batina, C., Gruteser, M., Herrera, J.-C., Herring, R., Hoh, B., Jacobson, Q., Iwuchukwu, T., Lew, J., Litrico, X., Luddington, L., Margulici, J., Mortazavi, A., Pan, X., Rabbani, T., Racine, T., Sherlock-Thomas, E., Sutter, D., Tinka, A.: Mobile century using GPS mobile phones as traffic sensors: A field experiment. In: Proceedings 15th World Congress Intelligent Transport Systems, New York (2008)
4. Chan, C.-Y.: On the detection of vehicular crashes-system characteristics and architecture. IEEE Trans. Veh. Technol. **51**(1), 180–193 (2002)
5. Needham, P.: Collision prevention: The role of an accident data recorder (ADR). In: Proceedings International ADAS Conference, pp. 48–51 (2001)
6. Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M., González, M.C.: Safe Driving Using Mobile Phones. IEEE Trans. Intell. Transp. Syst. **13**(3), 1462–1468 (2012)
7. Saiprasert, C., Pholprasit, T., Pattara-atikom, W.: Detecting driving events using smartphone. In: Proceedings of the $20^{th}$ ITS World Congress (2013)
8. Lotan, T., Toledo, T.: An in-vehicle data recorder for evaluation of driving behavior and safety. In: Proceedings of the Transportation Research Board 85th Annual Meeting, pp. 112–119 (2006)
9. Musicant, O., Lotan, T., Toledo, T.: Safety correlation and implications of in-vehicle data recorder on driver behavior. In: Proceedings of the Transportation Research Board 86th Annual Meeting (2007)
10. Ueyama, M., Ogawa, S., Chikasue, H., Muramatu, K.: Relationship between driving behavior and traffic accidents - accidents data recorder and driving monitor recorder. In: Proceedings of the 16th International Technical Conference on Experimental Safety Vehicles, Paper No. 98-s1-o-06, Windsor (1998)
11. Arai, Y., Nishimoto, T., Ezaka, Y., Yoshimoto, K.: Accidents and near-misses analysis by using video drive-recorders in a fleet test. In: Proceedings of the 17th International Technical Conference on the Enhanced Safety of Vehicles Conference, Paper No. 01-s4-o-255, Amsterdam (2001)
12. Green, E.R., Agent, K.R., Pigman, J.G.: Evaluation of Auto Incident Recording System, Kentucky Transportation Center, University of Kentucky, Report No. KTC-05-09/SPR277-04-1F (2005)
13. Han, I., Yang, K.S.: Characteristic analysis for cognition of dangerous driving using automobile black boxes. International Journal of Automotive Technology **10**(5), 597–605 (2009)
14. Johnson, D.A., Trevidi, M.M.: Driving style recognition using a smartphone as a sensor platform. In: Proceedings 14th International IEEE Conference on Intelligent Transportation Systems, pp. 1609–1615 (2011)
15. Lan, M., Rofouei, M., Soatto, S., Sarrafzadeh, M.: SmartLDWS: A robust and scalable lane departure warning system for the smartphones. In: Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (2009)
16. Mohan, P., Padmanabhan, V.N., Ramjee, R.: Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM conference on Embedded network sensor systems, pp. 323–336 (2008)
17. Thompson, C., White, J., Dougherty, B., Albright, A., Schmidt, D.C.: Using smartphones to detect car accidents and provide situational awareness to emergency responders. In: Proceedings of the 3rd mobile wireless middleware, operating systems, and applications conference, pp. 29–42 (2010)
18. Hong, I.K., Ryu, J.B., Cho, J.H., Lee, K.H., Lee, W.S.: Developmentof a driving simulator for virtual experience and training of drunk driving. In: Proceedings of the 3rd International Conference on Road Safety and Simulation (2011)
19. Mortimer, R.G., Segel, L., Dugoff, H., Campbell, J.D., Jorgeson, C.M., Murphy, R.W.: Brake Force Requirement Study: Driver-Vehicle Braking Performance as a Function of Brake System Design Variables, National Highway Safety Bureau (1970)
20. Anderson, R.W.G., Baldock, M.R.J.: Vehicle Improvements to Reduce the Number and Severity of Rear End Crashes, CASR Report Series, University of Adelaide (2008)
21. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition, vol. 26 (1978)
22. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: AAA1-94 Workshop on Knowledge Discovery in Databases, pp. 359–370 (1994)
23. Hiri-O-Tappa, K., Pan-Ngum, S., Sorawit Narupiti, S., Pattara-Atikom, W.: A novel approach of dynamic time warping for short-term traffic congestion prediction. In: Transportation Research Board 90th Annual Meeting. No. 11-3402 (2011)
24. Keogh, E., Pazzani, M.: Derivative dynamic time warping. In: Proceedings 1st SIAM International Conference on Data Mining (2001)