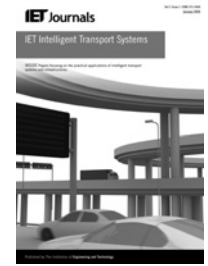


Published in IET Intelligent Transport Systems  
 Received on 15th January 2013  
 Revised on 8th May 2013  
 Accepted on 4th July 2013  
 doi: 10.1049/iet-its.2013.0009



ISSN 1751-956X

# Driving behaviour-based event data recorder

Bing-Fei Wu, Ying-Han Chen, Chung-Hsuan Yeh

Department of Electrical Engineering, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan  
 E-mail: winand@cssp.cn.nctu.edu.tw

**Abstract:** A general event data recorder is a device installed in automobiles to record information related to vehicle crashes or accidents. The data provide a better understanding of how certain crashes come about. This study made a prototype of a driving behaviour-based event data recorder (DBEDR), which provides the information of driving behaviours and a danger level. The authors approach is to recognise the seven behaviours: normal driving, acceleration, deceleration, changing to the left lane or right lane, zigzag driving and approaching the car in front by the hidden Markov models. All data were collected from a real vehicle and evaluated in a real road environment. The experimental results show that the proposed method achieved an average detection ratio of 95% for behaviour recognition. The danger level is inferred by fuzzy rules involved with the above behaviours. DBEDR recorded the recognised driving behaviours and the danger level, and the places were stored with the assistance of a global positioning system receiver. By integrating Google Maps, the locations, the driving behaviour occurrences, the danger level on the travel routes and the recorded images, the proposed DBEDR could be more useful compared with the traditional EDRs.

## 1 Introduction

The study from the National Highway Traffic Safety Administration reported that each year about 56 000 car accidents caused by driver fatigue, of which about 1500 drivers died. In Taiwan, a road accident happens every 2 min on an average, whereas drunk driving and other human errors are the first cause of the accidents in a decade. To protect the rights of pedestrians and drivers, an event data recorders (EDRs), which is a device installed in automobiles to record information related to vehicle crashes or accidents, is widely used among drivers [1]. Most EDRs can record longitudinal, lateral accelerations (ACCs), velocities, headings, headlights, indicator lights, breaks, speakers etc. In recent years, the function of video recording becomes widespread in EDRs, some of them even can access to the Internet to upload the data to a remote server [2].

Many researchers have been developed on the design of EDRs, the integration of them and the analysis from the data of them. Perez *et al.* [3] presented Argos, which is an improved in-vehicle data recorder (IVDR) that allows recording many kinds of alphanumeric data such as the speed (vehicle data), the point of gaze (driver data) or the current distance to lateral road marks (environmental data), to help researchers in the study of car driver behaviour. Jain and Busso [4] employed the UTDrive platform, a car equipped with multiple sensors, including cameras, microphones and controller area network-bus (CAN-bus) information to identify relevant features extracted from a frontal video camera and the car CAN-bus data that can be used to distinguish between the normal and task driving conditions. Toledo *et al.* [5] discussed the potential of IVDR systems to be used in various commercial and

research applications as tools to monitor and provide feedback to drivers on their on-road behaviour.

Moreover, the driving safety is such an important concern, so that a wide range of automotive safety systems have been released, such as driving blind spot detection, a 360° panorama imaging system [6–8]. In addition to the system focused on the detection of the surrounding status, the research of monitoring a driver's performance to prevent potential risks is growing. If the EDR can be integrated with the information of driving behaviour analysis, it can provide more useful information when accidents occur.

The methods to determine the degree of dangerous driving are broadly divided into two categories. One is to use physiological signals and their derivative status of drivers, and the other is to use the behaviours of their own vehicles. The first type considers the danger coming from the driver itself. Driver fatigue, distraction and drunkenness are often to blame. If a system can identify these states, drivers might be awakened in time to correct their courses by the system warning and the possible accidents might be avoided. Various physiological and biobehavioural signals such as heart rate [9, 10], electroencephalogram (EEG) [11–13], electrocardiograph (ECG) [14] and respiration rate [15] are used to distinguish the driver's state. However, to acquire the above signals, one or more intrusive sensors have to be attached on a driver's body, and this may affect the willingness of the drivers to use the system. The other kind is to use cameras to recognise the facial expression using the features such as eyes and a lip [16]. However, the variance of light and shadow during the day affects the visual cues of drivers. It is a difficult challenge by using the computer vision techniques to always obtain accurate and robust results. On the other hand, the second type aims

on the behaviour of vehicles, not that of human. The interested features include the parameters of the host vehicle and the environment such as lateral positions, ACCs/ decelerations (DECs), the distance to the vehicle in front and the distance to the lane markings [17, 18]. These features also indirectly indicate drivers' states and can be extracted by various sensors. Many researchers have studied on this type to develop driving safety systems.

There are several challenges in developing a safety monitoring system by identifying a vehicle's behaviours. For the rule-based approaches, it is difficult to design a comprehensive set of rules to cover all dangerous behaviours. Therefore some researchers use statistical methods to determine the dangerous behaviours by using pattern recognition techniques. However, the pattern of dangerous behaviour is difficult to define, and the boundary of the pattern may not be so clear. Other researchers use regression techniques to identify the danger level. However, it is also arduous to explain the influence of each parameter in the regression model.

In this paper, we propose a driving behaviour-based EDR (DBEDR). The driving behaviour is firstly recognised using the hidden Markov models (HMMs), and then the danger level is inferred by the fuzzy logic. There are seven driving behaviours included in our system: normal driving (ND), ACC, DEC, changing left (CL), changing right (CR), zigzag driving (ZD) and approaching the car in front (AFC). Except the recognised behaviours, the corresponding degree reflects the intensity of each behaviour is also calculated. The parameters for calculating the degree come from a camera, an accelerometer and a global positioning system (GPS) receiver, which provide the information of the lane bias, the distance to the car in front, the frontal/lateral accelerations and the velocity of the host vehicle. This information is used in a fuzzy inference system (FIS) to yield a danger level. The higher the value represents, the worst status of the driver is. We try to treat the process into a clear analysis. The first stage is a classification-based problem and we have to well design the approach to obtain a good detection ratio. In the second stage, an expert system is relied and we expect to offer drivers a reasonable and clear explanation for the meaning of the danger-level indicator.

When the driver is driving on the road, not only the information of behaviours and danger level are stored, but also the position, velocity, accelerations, images and timestamp are also kept. Therefore when the user examines the data offline by our event viewer, he can view his trajectory on Google Maps and the route is marked with different colours according to the danger level. The user can check the route with the higher danger level by a single click, and he can quickly know the status of various sensor values and the behaviours at that moment. For the applications such as fleet management, our system provides extra information (behaviours and danger level) of the driver, which could be beneficial for managers to evaluate the driver's performance. The main contributions of this paper are on providing drivers a DBEDR with the information of danger level and caused behaviours, which is particularly useful for management applications; and on using easily installed sensors without any intrusive interfaces, both for human and vehicles, and it greatly increases willingness for the system usages. The rest of the paper is organised as follows: Section 2 addresses the system architecture of DBEDR. Section 3 describes the proposed methodology from data and knowledge acquisition to infer the danger level. Implementation and experimental results are presented in Section 4. Finally, the conclusions are given in Section 5.

## 2 System architecture

The system architecture and functional schemes of DBEDR are shown in Fig. 1 and described as follows. DBEDR was designed to run on a dual-core embedded platform. Therefore the functional modules were carefully planned to enhance the processing performance. Three sensors including a GPS receiver, an accelerometer and a charge-coupled device camera are used to extract the needed parameters for the system usage. The GPS data are transferred to the platform through the Bluetooth interface, whereas the data from the accelerometer are transferred through a USB to RS232 connector. A Radio Corporation of America connector is used to connect the camera and the video input on the platform, and then the system can access the image buffer by the video for Linux two (V4L2) driver. The digital signal processor (DSP) core has implemented the image processing algorithm of lane departure warning system (LDWS) and forward collision warning system (FCW), which are mainly designed to detect lane markings and vehicles, and the details can be referred in our previous work [19, 20]. In this work, LDWS and FCW are used to obtain the parameters of lane bias and the distance to the car in front. Moreover, the DSP core also serve the function of encode, which compress the raw images into the video in H.264 format for our event viewer in PC. An embedded Linux is run on the advanced reduced instruction set computing (RISC) machine (ARM) core and dealing with the peripheral and graphical user interface. There are four main functional modules on the ARM side: data preprocessing, behaviour recognition, danger-level inference and data logger. All the data from sensors (velocity and acceleration) and DSP (lane bias and the distance to the car in front) are collected and processed in the data preprocessing module. Then, the behaviour recognition module use the collected data as patterns do recognise the behaviours by HMM. After the behaviours are recognised and the corresponding degrees are calculated, the danger-level module uses the obtained information to infer the danger level by an FIS. All information is displayed on the screen of the platform, and the data logger module records the information such as positions, velocities, accelerations, timestamps, current behaviours and danger level. For our event viewer, these data can be loaded to provide a convenient interface to examine the drivers' status.

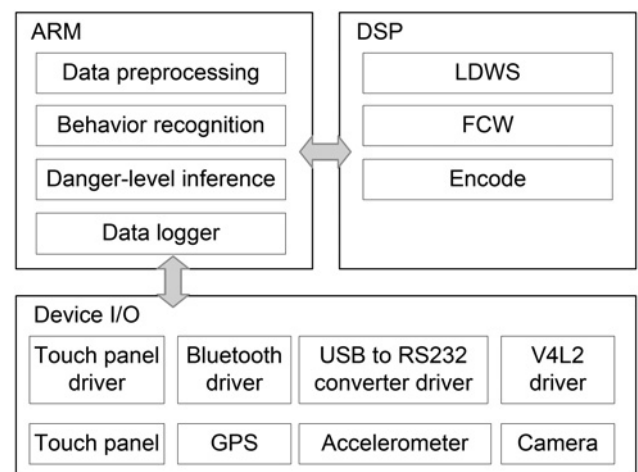


Fig. 1 System architecture and functional schemes of DBEDR

### 3 Design of functional modules

There are four functional modules in the system. First, data are acquired from the camera, the accelerometer and the GPS receiver, and then the interested features are extracted. Second, the features are applied to recognise the most likely driving behaviour using HMMs. The corresponding parameters of the recognised behaviours, which will affect the dangerous degree, are also calculated. Third, these parameters are used to deduce the danger level by a FIS. Finally, the information of recognised behaviours and inferred danger level will be recorded for users to examine the driving performance offline.

#### 3.1 Data acquisition and preprocessing

The data collected from sensors include the lane bias, the distance to the car in front, the frontal/lateral accelerations and the velocity of the host vehicle. The lane bias and the distance to the car in front are acquired from the captured images using the image processing technologies, containing the image/world coordinate transformation, the lane model regression and the morphology detection. On the other hand, the acceleration and the velocity are directly obtained from the accelerometer and the GPS receiver. However, the data have to be preprocessed before entering the next stage. First, a second-order Butterworth lowpass filter with a 2 Hz cutoff frequency is applied to reduce the influence of noise. Since a vehicle is of a relatively large mass, the behaviours of concern in our system generally belong to the low-frequency part. Second, the input data with high sampling rate have to be downsampled to reduce the computational load. Both the data from the accelerometer and the camera are downsampled to 5 Hz. Taking the accelerometer as an example, it provides the three-axis acceleration at a sampling rate of 50 Hz. The mean is calculated every ten samples as a new value. It implies the original sampling rate of 50 Hz is downsampled to a rate of 5 Hz. Third, the downsampled data are gathered through a sliding window. We assume that the normal behaviour of a vehicle is presented within 5 s. That means there are 250 samples of raw data and 25 samples after downsampling. Therefore the length of the sliding window is set to 25 to acquire sufficient data. Fourth normalisation used to standardise the range of independent variables or features of data is applied. Since the range of values of raw data varies widely, objective functions do not work properly without normalisation in many machine learning algorithms. The selected HMM method is no exception. The minimum and the maximum are selected from the 25 samples in the window, and these two values are used to scale the data range into  $[0, 1]$ . Finally, the vector quantisation is introduced to generate distinct observation symbols for the HMM usage. Vector quantisation works by encoding values from a multidimensional vector space to a finite set of values from a discrete subspace of a lower dimension. The dimension of quantised data represents the number of symbols used in an HMM, and it affects the HMM complexity. The quantised data with a low dimension may eliminate the variation of the data, whereas data with a high dimension increases the computational complexity. By observing the recording data, we chose a dimension of ten to preserve the trend of the data. It means that the normalised value in  $[0, 1]$  is quantised with an integer in  $[1, 10]$ . To achieve this purpose,  $k$ -means, a classic clustering algorithm, is used to divide the original data into

**Table 1** Driving behaviour categories

Longitudinal behaviours	Lateral behaviours	Car-following behaviours
ND	ND	ND
ACC	CL	AFC
DEC	CR	
	ZD	

one of ten groups, and each of which is represented by its centroid point.

#### 3.2 Behaviour recognition

To examine the driving behaviours clearly and to study further information, driving behaviours are divided into three categories, longitudinal behaviours, lateral behaviours and car-following behaviours. Each category contains one normal situation and other situations. Details are listed in Table 1. With these behaviour categories, it is easier to describe one driving section, and further analyse the danger level. Therefore the HMM is applied to recognise the driving behaviours.

HMM is a probabilistic tool for time series data recognition. Owing to its stochastic nature, HMM successfully has a wide range of applications in area of pattern recognition, especially in speech recognition. Theories of HMMs were introduced by Baum and Petrie in the late 1960 s [21]. In this section, only basic concepts are introduced. Detailed tutorials on HMM could be found in [22].

An HMM is characterised by the following elements:

- $N$ , the number of states in the HMM model.
- $M$ , the number of distinct observation symbols per state.
- The state transition probability distribution  $A = \{a_{ij}\}$ , where  $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$ ,  $1 \leq i$  and  $j \leq N$ .
- The observation symbol probability distribution in state  $j$ ,  $B = \{b_j(k)\}$ , where  $b_j(k) = P[v_k \text{ at } t | q_t = S_j]$ ,  $1 \leq j \leq N$  and  $1 \leq k \leq M$ .
- The initial probability distribution  $\pi = \{\pi_i\}$ , where  $\pi_i = P[q_1 = S_i]$ ,  $1 \leq i \leq N$ .

Therefore an HMM  $\lambda$  could be specified as  $\lambda = (N, M, A, B, \pi)$ . The observation probability of the sequence  $O$  is  $P(O|\lambda)$ .

Moreover, there are different types of HMMs according to the limitation on the state probability matrix  $A$ . The architecture of HMMs adopted in the algorithm is the so-called left-to-right model. The left-to-right model always starts from the first state, and the transitions are only allowed towards right state or the same state. The left-to-right model is better than the general model at performing dynamic pattern recognition, like speech recognition, gesture recognition and signature verification because it emphasises more on the context relationship between states.

For each driving behaviour, an HMM is constructed. As mentioned above, the left-to-right HMM was adopted. The training phase is done by the Baum–Welch reestimation method. The larger number of symbols would reduce the quantisation error, but in the meanwhile reduces the recognition rate and increases the computation complexity. On the other hand, an HMM with a large number of states leads to a larger error rate. Therefore the numbers of states and symbols of each model are tested in order to find the preferable result. In all the cases, except

for an ND, the number of states is five; for an ND, the number of states is two. The number of all symbols is ten. The number of symbols is decided according to the design in the quantification step. As to the number of states, we made preliminary tests with various numbers of states. The number was chosen from two to five to train each model. Then, observation probabilities were found for each observation sequence from the same set. The number that had the smallest variance in output probabilities was then selected as the parameter of each HMM, because the results show that it effectively minimised false rejections.

The behaviour recognition procedure is shown in Fig. 2. Given an observation sequence, the observation probabilities of each HMM model are calculated by the forward algorithm. Then, the model with the highest probability is selected as the recognised manoeuvre. In other words, for one observation sequence set, one of the longitudinal behaviours is output as the result, like DEC. Similarly, the lateral behaviours and the car-following behaviours.

### 3.3 Danger-level inference

The major task in the second stage is to combine the effect of each driving event in order to infer the danger level using an FIS. The degree of each driving event is estimated as a quantifiable indicator that represents a more explicit description. In addition, a hierarchical decision strategy is also presented to improve the efficiency of the fuzzy rule selection by using the recognised results obtained from the first stage as conditions. The danger level provides drivers an intuitive indicator of current driving status. Three basic states, normal, attention and danger, are designed as three different danger levels. However, it is a challenge to exactly define what values belong to which danger level. We know driving in high speed is dangerous, but the boundary of the speed to represent danger or not is ambiguous. To handle this kind of problem, an effective method, fuzzy logic, was adopted to model the human decision making to achieve our objective. The basic concept underlying fuzzy logic is the linguistic variable, which is a variable whose values are words rather than numbers. A general fuzzy logic model contains four components: fuzzification, fuzzy inference engine, defuzzification and a fuzzy rule base.

In the first stage, the recognised behaviours with the corresponding parameters are obtained. In this stage, these results are used to design the fuzzy IF-THEN rules to infer the desired danger level. Except ND, recall that there are six recognised behaviours including ACC, DEC, CL, CR, ZD and AFC. If each behaviour is designed for three degrees, such as HIGH, MEDIUM and LOW, the rules of combinations have more than 700. Not to mention the other factor like the velocity of the host vehicle has not been

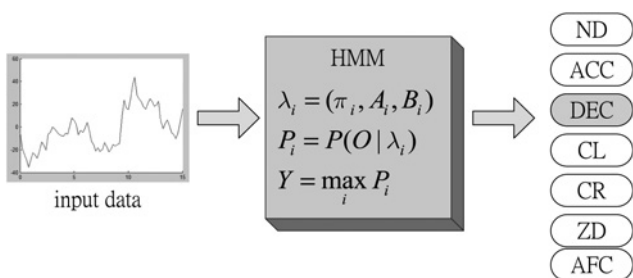


Fig. 2 HMM recogniser

considered yet. Too many rules increase the computational load, and it also lacks a clear understanding of the relationship between the behaviours and the danger level.

To overcome the above shortcomings, a hierarchical process flow is designed to decrease the number of fuzzy rules. There are three ideas involved. First, we can know that some behaviours are mutually exclusive, that is, they cannot simultaneously occur. For example, when the vehicle is zigzagging in a lane, the behaviour of lane changing cannot happen at the same time. Therefore if we know the current behaviour is the ZD, the parameters corresponding to the lane changing can be ignored. Thus, the number of combinations is reduced. Second, if the behaviour itself does not exist, its effect has not to be considered. For example, the behaviour of the host vehicle AFC of it can only be observed when there actually is a car in front. Third, if the parameters of corresponding behaviours are the same, they can be merged by using the same membership function. For example, the parameters of ACC and DEC are both the variation of longitudinal accelerations. According to the sign of the values, we know it represents the influence of accelerations or DECs. Therefore we can create a new linguistic variable acceleration and deceleration (AD) to denote the effect of ACC and DEC. In a similar sense, we also use another linguistic variable, LC (lane changing), to indicate the degrees of CL and CR. According to these three conditions, the fuzzy rules can be reduced by the hierarchical process flow as shown in Fig. 3.

### 3.4 Data logger and event viewer

The data logger module is responsible to record the important information in a secure digital (SD) card, whereas the event viewer is a program running in PC to load the data in the SD card. There are two types of files which will be loaded together. One is a text file, and the other is a video file. Both of them are indexed with the date. The text file stores the data of positions, velocities, accelerations, timestamps, current behaviours and danger level, whereas the video is the compressed H.264 video at the same date. Remember that Google Maps is utilised in the event viewer to provide a friendly interface of maps. To show the customised information on the map, all the data in the text file have to be extracted and converted to keyhole markup language format. The user can find that there are many markers with

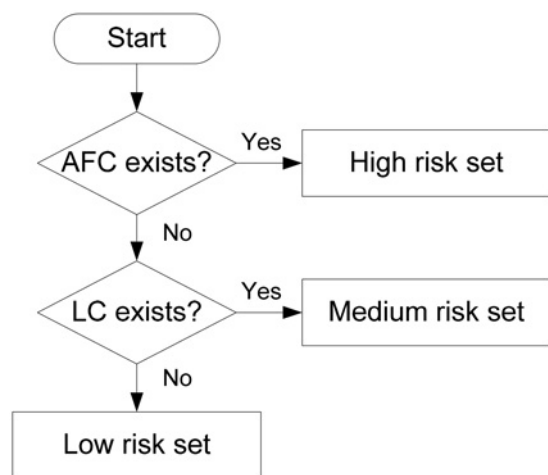


Fig. 3 Hierarchical process flow

three different colours on the map. The colour is set according to the danger level. Therefore the user can easily identify the zone with the various driving performance by browsing these markers. Our event viewer provides two different ways for users to efficiently review the recorded videos. The first way is 'select by the list'. All information is listed in a table and ordered according to the timestamp. This mode is suitable for users to quickly examine the video at a certain moment. The second way is 'select by the map'. The user just click the marker on the map and the timeline of the video will immediately jump to the time of the event occurring. This mode is particularly suitable for users who want to examine their driving status at certain locations.

## 4 Experimental results

This section addresses the implementation details and the experimental results. All functionalities have been successfully implemented and installed in our experimental car. The camera was mounted in the front windshield, and the accelerometer at a sampling rate of 50 Hz and the Bluetooth GPS receiver were attached on the top of the dashboard. The system prototype was implemented on a platform based on a TI DM3730 processor, which has a 1 GHz ARM core and a 600 MHz DSP core as shown in Fig. 4. The captured image size is  $320 \times 240$  (QVGA), and the processing performance achieved 30 frames per second for lane marking and vehicle detection. All data for training and testing were recorded under the real environment with the experimental car. The experimental results of each function are described below.

### 4.1 Recognition results

All data for training and testing are recorded with the experimental car under the environment of the freeway, highway and routes in suburban in Taiwan. Freeway and highway are used for recording the data at high speed, whereas the routes in suburban are for the one at low speed. Three test drivers are requested to execute all driving events under the safe conditions. The data were all collected in sunny and cloudy conditions at daytime. The recorded data were manually selected for training and testing the HMMs of all driving events. The  $x$ -axis acceleration is used for longitudinal behaviours, the lane bias is used for lateral behaviours, and the distance to the front car is for car-following behaviours. Table 2 contains the detection ratio of the experimental results. Detection ratio is defined

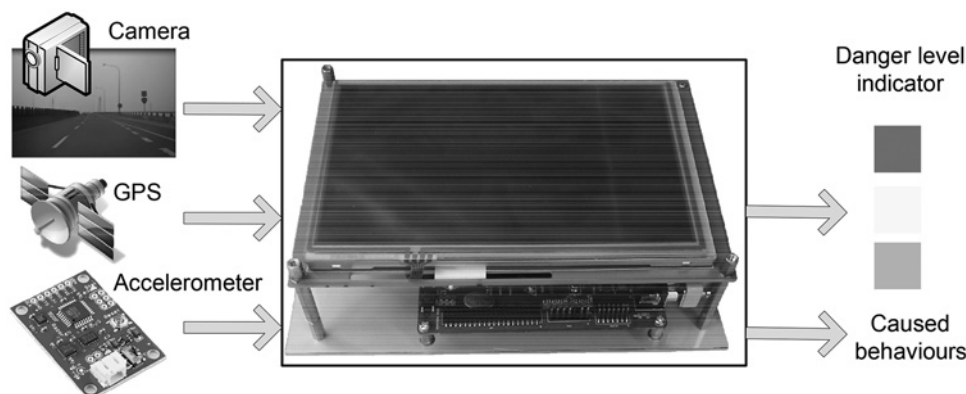
**Table 2** Detection ratio of behaviour recognition

	False negative/true positive	Detection ratio, %
ND	0/193	100
ACC	2/113	98.2
DEC	0/110	100
CL	1/161	99.3
CR	2/158	98.7
ZD	2/152	98.7
AFC	0/102	100

as true positive/(true positive + false negative), where true positive means the number of evaluated sequences for which the driving behaviours are correctly recognised, and false negative represents the number of evaluated sequences for which the driving behaviours are misrecognised. Of about 30% of the data of each evaluated sequence were used to train each HMM, whereas the remainder was used to evaluate the detection ratio of recognition. It can be noted that ND, DEC and AFC are perfectly discriminated by the classifier. Although the behaviours of acceleration and DEC are similar, ACC may be misrecognised as ND because of the unobvious variation of acceleration compared with DEC. This can be understood from the different delay effect of the throttle and brakes. ZD is the one with worst recognition ratio, because the trendy is very similar to ND, the data of ZD with a smaller variation are confused with that of ND.

### 4.2 Event viewer

The event viewer loads the recorded data of the DBEDR to let drivers examine their routes and driving status. Fig. 5 shows the list mode, which is suitable for users to quickly examine the video at a certain moment. The left side of the event viewer is a video player used to display the video file, whereas the right side is a list table, which shows the information of the danger level and the timestamp. Furthermore, the timestamp in the list and the timeline of the video is synchronised. When the user selects the certain row of the list, the video will jump to the time of the event occurring. On the other hand, Fig. 6 shows the map mode, which is suitable for users to examine their driving status at certain locations. The left side is still a video player, whereas the right side shows Google Maps. The map displays the trajectory and the markers. The user can easily drag the map to examine his trajectory. The markers



**Fig. 4** System prototype



Fig. 5 List mode of the event viewer



Fig. 6 Map mode of the event viewer

indicate the places of the events happening, and the danger level is represented with red/yellow/green. The red marker means an event with high risk, whereas the green one is low risk. If the user clicks the marker, a popup dialog with the information of timestamp and recognised driving behaviours will appear. If the recognised driving behaviour is not an ND, it will be displayed in red. In the meantime, the video also jumps to the time of the event occurring.

## 5 Conclusions

We have presented a driving DBEDR for providing the information of driving behaviours and the danger level. The seven driving behaviours including ND, acceleration, DEC, changing to the left lane or right lane, ZD and AFC, are recognised by HMMs, and the danger level is inferred by an FIS through a hierarchical process flow. Except the information of driving events, a single index to evaluate a driver's performance is very useful, because it is convenient to know drivers' status on their routes. People could more easily clarify the cause of an accident or know the stability of a driver by examining the occurring driving events according to the danger levels. The event viewer provides two modes for users to efficiently examine the event data

by list mode and map mode, which mainly prevent users from spending time by searching certain interesting moments of the video. The proposed DBEDR offers users the additional information of driving risk compared with the traditional EDR. The developed functions were fully implemented on an embedded platform and tested in the real environment. Experimental evaluation shows that all HMM training models for seven behaviours achieved an average detection ratio at more than 95%. Currently the proposed DBEDR is implemented on an embedded platform. Owing to the prototype of DBEDR, the installation of the system is a little inconvenient. Three sensors have to be properly set first in the vehicle. This is a disadvantage for people who want to try our system. Therefore to ease the installment of DBEDR is another goal of development. With the growing of powerful smartphones, the porting of DBEDR is possible [23]. Since the sensors used in DBEDR, that is, a camera, an accelerometer and a GPS receiver, most smartphones in the market also have. However, the performance has to be optimised because of the computing load, especially in image processing. This direction is an interesting field because we believe that if DBEDR can be deployed as application software (APP) for smartphones, it would

encourage people to use this system. Moreover, we plan in the future to add more sensors to collect more data of the vehicles for identifying more driving behaviours. The use of a hierarchical HMM structure for recognising more general driving behaviours patterns also belongs to our future plan.

## 6 Acknowledgment

This work was supported by the National Science Council, Taiwan under grant no. NSC 100-2221-E-009-041-.

## 7 References

- Gabler, H.C., Gabauer, D.J., Newell, H.L., O'Neill, M.E.: 'Use of event data recorder (EDR) technology for highway crash data analysis' (Transportation Research Board of the National Academies, 2005)
- Hua, Y., Gang, D.: 'A digital vehicle monitoring system based on 3G for public security'. Proc. Computer and Information Application (ICCIA), Tianjin, December 2010, pp. 146–148
- Perez, A., Garcia, M.I., Nieto, M., Pedraza, J.L., Rodriguez, S., Zamorano, J.: 'Argos: an advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation', *IEEE Trans. Intell. Transp. Syst.*, 2010, **11**, (2), pp. 463–473
- Jain, J.J., Busso, C.: 'Analysis of driver behaviors during common tasks using frontal video camera and CAN-bus information'. Proc. Multimedia and Expo (ICME), Barcelona, Spain, July 2011, pp. 1–6
- Toledo, T., Musicant, O., Lotan, T.: 'In-vehicle data recorders for monitoring and feedback on drivers' behavior', *Transp. Res. C, Emerg. Technol.*, 2008, **16**, (3), pp. 320–331
- Ehlgen, T., Pajdla, T., Ammon, D.: 'Eliminating blind spots for assisted driving', *IEEE Trans. Intell. Transp. Syst.*, 2008, **9**, (4), pp. 657–665
- Chen, Y.Y., Tu, Y.Y., Chiu, C.H., Chen, Y.S.: 'An embedded system for vehicle surrounding monitoring'. Proc. Power Electronics and Intelligent Transportation System (PEITS), Shenzhen, December 2009, pp. 92–95
- Gandhi, T., Trivedi, M.M.: 'Vehicle surround capture: survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps', *IEEE Trans. Intell. Transp. Syst.*, 2006, **7**, (3), pp. 293–308
- Atsumi, B.: 'Evaluation of mental condition on drivers by analysis of heart rate variability: measurement of mental stress and drowsiness by indexes of autonomic nervous system', *JSAE Rev.*, 1995, **16**, (1), pp. 110–110
- Jiao, K., Li, Z., Chen, M., Wang, C., Qi, S.: 'Effect of different vibration frequencies on heart rate variability and driving fatigue in healthy drivers', *Int. Arch. Occup. Environ. Health*, 2004, **77**, (3), pp. 205–212
- Jap, B.T., Lal, S., Fischer, P., Bekiaris, E.: 'Using EEG spectral components to assess algorithms for detecting fatigue', *Expert Syst. Appl.*, 2009, **36**, (2, Part 1), pp. 2352–2359
- Yeo, M.V.M., Li, X., Shen, K., Wilder-Smith, E.P.V.: 'Can SVM be used for automatic EEG detection of drowsiness during car driving?', *Saf. Sci.*, 2009, **47**, (1), pp. 115–124
- Shen, K.Q., Li, X.P., Ong, C.J., Shao, S.Y., Wilder-Smith, E.P.V.: 'EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimate', *Clin. Neurophysiol.*, 2008, **119**, (7), pp. 1524–1533
- Chua, C.P., McDarby, G., Heneghan, C.: 'Combined electrocardiogram and photoplethysmogram measurements as an indicator of objective sleepiness', *Physiol. Meas.*, 2008, **29**, pp. 857–868
- Lal, S.K.L., Craig, A.: 'A critical review of the psychophysiology of driver fatigue', *Biol. Psychol.*, 2001, **55**, (3), pp. 173–194
- D'Orazio, T., Leo, M., Guaragnella, C., Distante, A.: 'A visual approach for driver inattention detection', *Pattern Recogn.*, 2007, **40**, (8), pp. 2341–2355
- Zehang, S., Bebis, G., Miller, R.: 'On-road vehicle detection: a review', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (5), pp. 694–711
- McCall, J.C., Trivedi, M.M.: 'Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation', *IEEE Trans. Intell. Transp. Syst.*, 2006, **7**, (1), pp. 20–37
- Chen, C.J., Peng, H.Y., Wu, B.F., Chen, Y.H.: 'A real-time driving assistance and surveillance system', *J. Inf. Sci. Eng.*, 2009, **25**, (5), pp. 1501–1523
- Wu, B.F., Chen, Y.H., Kao, C.C., Li, Y.F., Chen, C.J.: 'A vision-based collision warning system by surrounding vehicles detection', *KSII Trans. Internet Inf. Syst.*, 2012, **6**, pp. 1203–1222
- Baum, L.E., Petrie, T.: 'Statistical inference for probabilistic functions of finite state Markov chains', *Ann. Math. Stat.*, 1966, **37**, (6), pp. 1554–1563
- Rabiner, L.R.: 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proc. IEEE*, 1989, **77**, (2), pp. 257–286
- Paefgen, J., Kehr, F., Zhai, Y., Michahelles, F.: 'Driving behavior analysis with smartphones: insights from a controlled field study'. Proc. Mobile and Ubiquitous Multimedia, Ulm, Germany, 2012, pp. 1–8