Journal of Knowledge Information Technology and Systems(JKITS), Vol. 9, No. 5, pp. 617~628, October 2014

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

An Efficient Recognition Method using 2 Layer Hidden Markov Models for Human Driving Behavior

Eun-Jae Shin, In-Ho Yun, Dinh Xuan Hao, Sang-Youn Kim, Goo-Cheol Jeong*

Interaction Lab., ATRC, Korea University of Technology and Education, Korea

ABSTRACT

This paper proposes a stochastic model for human driving behavior using a double layer Hidden Markov Model (HMM) with continuous observations. In the proposed model, gas pedal's position, steering wheel's angle, velocity and angular velocity of the vehicle is used and recorded in every 100 msec for recognizing driving task. Data acquisition is done during a simulated driving task, after that data is segmented and clustered into 9 different cases. The lower-layer with one-dimensional continuous HMM is used for recognizing translational acceleration of a vehicle. The upper-layer with one dimensional continuous HMM is used for recognizing angular velocity of a vehicle. For recognizing a user's behavior, we used sliding windows with size of 10 samples and length of sequence with size of 30 samples. We apply a Kalman filter to reduce noise. After the filtering, the data was processed by sorting into three groups for a pedal, a steering wheel, and speed. We used two main features, which are angular velocity and the translational acceleration, in order to present driving behavior. We constructed a driving simulator based on Logitech G27 Racing platform to evaluate the proposed method. Using the developed driving simulator, some experiments were conducted for comparing the accuracy rate of the proposed method with that of the conventional method. Futhermore, we compared the average learning time of the proposed method with that of the conventional method because Learning time becomes also an important factor for investigating the performance of stochastic models. The experimental results confirm the accuracy of the proposed approach by revealing recognition accuracy around 96%-97%. Furthermore, the proposed method decreases learning time around 40%.

KEYWORDS: Hidden Markov Models, Driving Patterns, Driving behaviors, Driving models

ARTICLE INFO: Received 17 September 2014, Revised 14 October 2014, Accepted 14 October 2014.

*Corresponding author is with the School of Computer Science and Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro Byeongcheon-myeon Dongnam-gu Cheonan 330-744, KOREA. *E-mail address*: jeong@jkut.ac.kr

1. Introduction

Nowadays, driving already has played an increasingly central role in our daily life. Whenever drivers drive their cars, they face highly complex tasks involving the execution of multiple critical subtasks. Generally, most of driving behaviors have been focused on operating a steering wheel, an accelerator and a brake pedal. Therefore, to investigate a driver’s behavior, many researchers have developed a variety of models [1,2,3,4,5,6,7,8,9,10,11] which are the representation of the driving signal obtained from a steering wheel, an accelerator, and a brake pedal. For obtaining the driving signals, it is necessary to recognize the driver’s behavior. Correct recognition of driver’s behavior can be useful for understanding automatic or auxiliary driving which secures drivers against driver's deviant status such as fatigue and drunken driving.

The driving models can be classified into two categories: a computational model and a stochastic model. The computational model, which is used to compute, to simulate, and to predict various driver behaviors, is a powerful tool for studying the theoretics of driver’s behavior and form the basis of the practical development of intelligent vehicle systems.

Stochastic models have been widely studied in signal processing and pattern recognition fields in order to effectively analyze the different features of time series data. The reason is that the stochastic model makes a target system stable, although there are some losses in dataset. An HMM (Hidden Markov Model) [14], which is one of the most famous stochastic models, is a representation for easily decoding and abstracting the human skill for both non-observable mental states and observable task states.

Driving signals, which are obtained from a gas pedal, a brake pedal, and a steering wheel, can be represented by the time series data. Since the stochastic model is a powerful tool for analyzing, recognizing the time series data, many researchers have applied a stochastic model to develop their driving models [1,2,3,4,5,6,7]. An intelligent vehicle security system for handling vehicle theft problems was proposed based on the modeling framework for dynamic human driving behaviors [1]. Wataru Takano et al. proposed an approach to stochastically and intelligently recognize the state of automobiles by using HMM [2]. A simple and reliable method for recognizing driving events using HMM was proposed [3]. The method assumed that useful navigation history, which is obtained by learning driving behavior patterns, was provided for drivers. Holger Berndt et al. proposed a method for investigating early driver intention inference with HMM [4]. A driver intension recognition system, which awares drivers’ lane change, was studied based on a continuous HMM [5]. Pinar Boyraz et al. proposed a hybrid driver model, which combines with a stochastic model for velocity control and a continuous control theoretic model for lateral control [6]. The main goal of [6] is to obtain a comprehensive driver models including all available knowledge and state-of-the-art methods for the development of human centric active safety. Amardeep Sathyanarayna et al. proposed a hierarchical framework to recognize driving and to detect a driver's distraction [7]. However, above research works has some limitations such as low accuracy rate [1,4,5,7], recognition of only steering wheel operation [3,4,5], control of only velocity [6], or prediction result of steering wheel angle with large errors [2].

There have been other research works (like Gaussian Mixture Models [8] or Neural Network [9]) for recognizing driving behavior. Chiyomi Miyajima et al. modeled driving behaviors with car-following and pedal operation patterns [8]. In those research works, the driver models were evaluated through driver identification experiments where driving signals were collected from a driving simulator and in a real vehicle. Rizal Othman et al. proposed another method for modeling and recognizing driving behaviors using linear prediction analysis and auto associative neural network [9]. However, the accuracy rate of their methods was just 76.8% [8], 81.7% [9], respectively. Moreover, only two signals obtained from a brake pedal and a gas pedal were considered for analysis [8,9].

Driving behavior are composed of multiple interaction processes, thus it has two or more variables for representing driving signals. With a single state variable, conventional HMM is illsuited to these problems. In order to model these interactions, a new architecture of HMM is needed. Some researchers proposed driving models using a double layer HMM for recognizing driving behavior [10,11]. Christian Laugier et al proposed a conceptual framework to analyze and to interpret dynamic traffic scenes by means of sensor fusion technology with the Bayesian Occupancy Filter [10]. A Smart Car test platform framework for modeling and for recognizing driving operations was proposed [11]. The system used coupled HMM for recognizing seven different driving operations such as passing, changing lanes to the left and right, turning right and left, starting and stopping. Their models were essential to build realistic automated virtual cars in driving simulators and to develop the human machine interface in driver assistance systems [11,12,13]. However, their systems have a problem in that their learning time increases in comparison with conventional HMM. Another problem is that the accuracy rate of turning left/right is not over 86% [10,11]. Therefore, in order to overcome these drawbacks, we propose a new type of a double layer HMM consisting of a lower layer and a upper layer for recognizing driving behavior. The lower-layer is used for recognizing translational acceleration corresponding to gas pedal operation event, in the upper-layer is used for recognizing angular velocity corresponding to steering wheel operation event. It is the fact that the learning time of the one-layer HMM is faster than that of the multiple-layer HMM. The learning time of the proposed system decreases around 40% in comparison with using one-layer HMM. So, we can conclude that this works can be useful for supporting driving systems.

2. System overview

2.1 Architecture of the proposed system

<Figure 1> shows the architecture of the proposed system. As we mentioned before, the proposed system consists of two layers: a lower layer and an upper layer. In the lower layer, three HMMs were applied for recognizing translational acceleration corresponding to the state of velocity. If a driver pushes a gas pedal, the speed is increased.

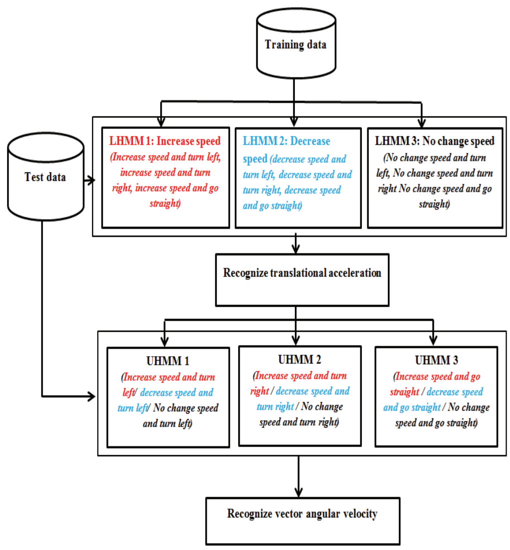


그림 1. 제안하는 시스템의 구성도

Figure 1. Architecture of the proposed system

In the contract, if a driver releases a gas pedal, the speed is decreased. Translational acceleration is used as an input for the lower layer consisting of three HMMs. The only one HMM's result of three with the highest likelihood in the lower layer was selected for understanding current acceleration on gas pedal. The obtained acceleration was transferred to the upper layer for recognizing angular velocity (turn left, turn right, and go straight). The upper layer is composed of three HMMs like the lower layer. In this layer, angular velocity is used as an input of each HMM. When a driver operates the steering wheel of a car, angular velocity is changed. The only one HMM's result of three with the highest likelihood in the upper layer like the lower layer was selected for recognizing angular velocity on steering wheel.

2.2 Experiment setup

A driving simulator was constructed based on Logitech G27 Racing Wheel force feedback device <Figure 2>. A graphic program was developed using VC++ and Ogre 3D.

그림 2. Logitech G27 모델 기반의 운전 시뮬레이터

Figure 2. Driving simulator is based on Logitech G27.

Ten subjects were participated in the experiments. The subjects are required to drive the vehicle along the driving track repeatedly for at least 5 times with limited velocity from 10m/s to 20m/s (36km/h to 72 km/h). The subjects controlled a gas pedal to increase/decrease the velocity and to operate steering wheel. During the experiment, the required driving signals such as gas pedal's position, steering wheel's angle, velocity and angular velocity of the vehicle were recorded in every 100 msec and the numbers in the parentheses are the number of bits.

2.3 Segmentation and clustering

Before recognition of driving behavior, we should prepare driving signal according to the following steps shown in <Figure 3>.

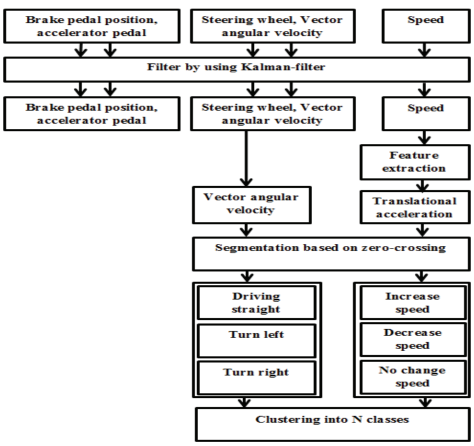


그림 3. 데이터 준비과정

Figure 3. Process of data preparation.

A Kalman-filter was used to eliminate noise [16]. After the filtering, the data was processed by sorting into three groups for a pedal, a steering wheel, and speed. We used two main features, which are angular velocity and the translational acceleration, in order to present driving behavior. The angular velocity, which related to rotate of vehicle, was segmented into three cases: "Driving straight, turn left, and turn right". The translational acceleration was segmented into three cases: "increase speed, decrease speed, and no change speed".

<Figure 4> shows the result of the translational acceleration. When a driver just pushes a gas pedal to increase velocity, the translational acceleration is increased. Whereas, when the driver releases the gas pedal, the translational acceleration is decreased. When the driver keeps a gas pedal to maintain velocity, the translational acceleration becomes near zero. Therefore, by analyzing the translational acceleration, we can easily understand the driver's behavior through a gas pedal operation.

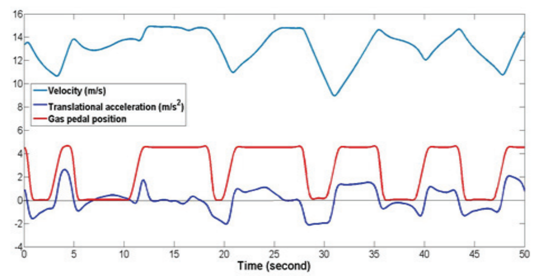


그림 4. 측정된 가속도

Figure 4. Translational acceleration

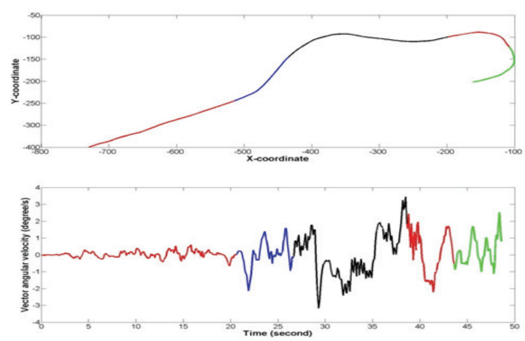
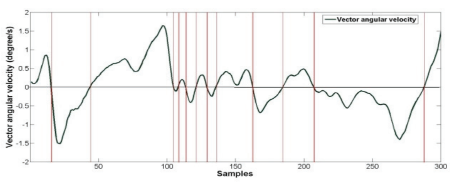


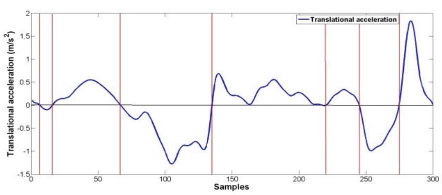
그림 5. 측정된 각속도

Figure 5. Angular velocity.

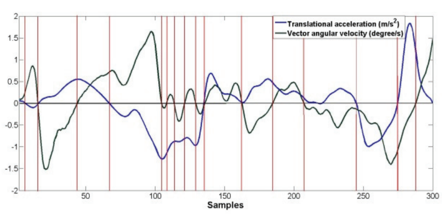
Let us remember that angular velocity is changed according to the control of a steering wheel. <Figure 5> shows the result of angular velocity. First a user drives his/her car to go straight. After that, he/she turn right after turning left. Again, the driver turns his/her car right and left. Finally, the driver turns his/her car right. In <Figure 5>, as we mentioned before, since the first red line shows the situation where a car goes straight, angular velocity becomes almost zero. When the car is turned right, the angular velocity becomes positive, whereas angular velocity is negative when the car is turned left. So if we investigate the obtained angular velocity, we can understand driving behavior on a steering wheel operation.



(a)



(b)



(c)

그림 6. 각속도의 segmentation(a), 가속도의 segmentation(b), 가속도와 각속도의 segmentation(c)

Figure 6. Segmentation of angular velocity (a), segmentation of translational acceleration (b), segmentation of translational acceleration and angular velocity (c).

The driving time series data (<Figure 4> and <Figure 5>) was segmented based on a criterion inspired by driving signal. Let us consider the case where a user releases gas pedal after pushing the pedal in order to increase and decrease the translational acceleration of a car. So, we selected the point where the translational acceleration becomes zero to segment. It means that whenever the translational accelerations cross the zero axes, a segment was begun. Similarly, when a driver rotates a steering wheel to turn left or turn right, angular velocity becomes negative or positive. Thus, we selected the point where the angular velocity becomes zero to segment. Translational acceleration and angular velocity were segmented as shown in <Figure 6>.

After the segmentation, we need to decide a criterion to categorize similar segmented data into a several clusters. Before categorizing, we define an upper threshold and a lower threshold as values for finding the similar segmented driving signals. By finding upper and lower threshold values based on histogram of each driving signal (translational acceleration and angular velocity), each the driving signal was approximated and discretized into three cases: 1) greater than the upper threshold, 2) between upper and lower thresholds, and 3) smaller than lower threshold. With the translational acceleration was categorized into three cases. If the magnitude of the translational acceleration is greater than the upper threshold (the first case), the velocity of a car increases. When the magnitude of the acceleration is in between upper threshold and lower threshold (the second case), the velocity is maintained.

When the acceleration is smaller than the lower threshold (the third case), the velocity decreases. Like the translational acceleration, the angular velocity was categorized into three cases. The first is the case when the magnitude of the angular velocity is greater than upper threshold. In the first case, a car operated by a driver turns right. When the magnitude of the angular velocity is in between upper threshold and lower threshold (the second case), the car goes straight. If the magnitude of the angular velocity is smaller than the lower threshold, the car turns left.

표 1. 9가지 경우의 표

Table 1. Label of 9 cases.

	Turn left	Go straight	Turn right
Increase speed	1	2	3
No change speed	4	5	6
Decrease speed	7	8	9

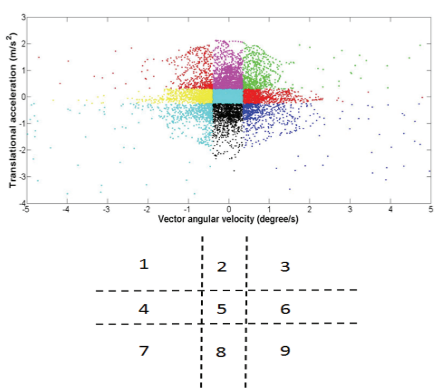


그림 7. 가속도와 각속도를 이용한 9가지 경우의 운전신호 역치 군집도

Figure 7. Threshold clustering driving signal into 9 cases using translational acceleration and angular velocity.

So, we categorized the segmented driving signal data as 9 cases using threshold clustering method [15] as shown in <Figure 7>. In <Figure 7>, the marked number's meaning is tabulated as <Table 1>.

3. Results and Evaluation

We constructed a driving behavior recognition system consists of two phases. The first phase is to require driving signal from the developed driving simulator. The driving simulator was constructed based on G27 Logitech steering wheel device as we mentioned before. After acquiring driving signal, the driving signals were analyzed, segmented, and clustered. The second one is a recognition phase for recognizing 9 cases of driving data. The recognition phase was developed using Kevin Murphy's HMM toolbox. Before training HMMs, we should make a decision what kind of HMM is useful for our experiment. We conducted a simple experiment for recognizing the case 1 of the 9 cases with different number of hidden states in order to find a useful number of hidden states in our experiment.

표 2. 은닉상태의 운전 정보에 대한 정확도

Table 2. Accuracy rate of case 1 of driving data with different number of hidden states.

Number Hidden state	2	3	4	5
Avg	93.45	94.12	88.24	91.18
Number Hidden state	6	9	12	15
Avg	88.06	82.35	91.18	88.24

표 3. 3가지 은닉 상태의 혼합 변량 가우시안 밀도함수의 다른 번호에 대한 운전정보의 정확도

Table 3. Accuracy rate of case 1 of driving data with different number of mixture univariate Gaussian density function in case three hidden states.

Number Mixture	1	2	3	4
Avg	94.12	51.06	51.06	65.96
Number Mixture	5	6	7	8
Avg	59.57	65.96	57.45	65.6

A computer with 4 Gbytes of RAM, core 2 due E8400 3.0 GHz of CPU was used during our experiment. We applied a mixed univariate Gaussian density function to this experiment. For the experiment, we assumed that length of sequence is 30 samples, and sliding windows size is 10 samples. The result of accuracy rate of the simple experiment is shown in <Table 2>.

As you observed the accuracy rate in <Table 2>, the accuracy rate of three hidden states is highest. Let us remember that HMM with left-right topology is the best case for recognizing the time series data. So, we chose continuous HMMs with left-to-right topology and three hidden states for recognizing driving data in our experiment. Similarly, we conducted a simple experiment in order to find a suitable number of mixture univariate Gaussian density function. We found that one mixed univariate Gaussian density function is suitable for our experiment (<Table 3>).

We conducted some experiments in order to compare the accuracy rate of the proposed method with that of the conventional method. As we mentioned before, translational acceleration was categorized into three cases. The translational acceleration data of each case was used as an input for training upper layer. The trained upper layer was used to recognize for translational acceleration of 9 cases. Angular velocity was categorized into three cases, the angular velocity data of each case was used as an input for training lower layer. The trained lower layer was used to recognize for angular velocity of 9 cases. In the recognition phase, sliding windows with size of 10 samples and length of sequence with size of 30 samples were used for training the upper layer and lower layer. The sequence with maximum likelihood value was chosen as a result of recognition. <Table 4> is the classification rate regarded to each case. As we observed the result, the average accuracy rate of the proposed method was about 97.47%. The average accuracy rate of one layer HMM was about 94.95%.

표 4. 단층 HMM과 가속도와 각속도를 이용하여 제안된 방법의 분류율 비교

Table 4. Comparison classification rate between one layer HMM, and proposed method using translational acceleration and angular velocity.

	Case 1	Case 2	Case 3	Case 4	Case 5
Hmm	81.82	100	95.45	100	100
Proposed	95.45	100	100	100	100
	Case 6	Case 7	Case 8	Case 9	Avg
Hmm	95.45	81.82	100	100	94.95
Proposed	95.45	86.36	100	100	97.47

According to the threshold-clustering method, we clustered translational acceleration and steering wheel angle into 9 cases as shown in <Figure 8>.

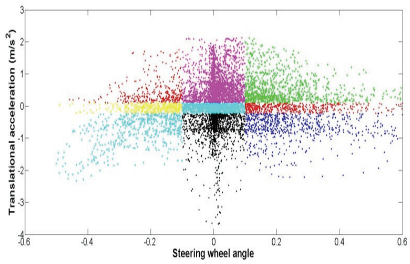


그림 8. 가속도와 핸들의 각도를 이용한 9가지 경우의 운전 신호 역치 군집도

Figure 8. Threshold clustering driving signal into 9 classes using translational acceleration and steering wheel angle.

By applying the proposed method to the experiment, classification results were obtained (<Table 5>). As we observed the recognition result, overall average correct classification ratio of the proposed method is about 95.98% and that of one layer HMM is about 92.43%. Let us compare <Table 4> with <Table 5>, we can easily observe that the classification rate from angular velocity is better than the rate from steering wheel angle.

표 5. 단층 HMM과 가속도와 핸들 각도를 이용하여 제안된 방법의 분류율 비교

Table 5. Comparison classification rate between one layer HMM, and proposed method using translational acceleration and steering wheel angle.

	Case 1	Case 2	Case 3	Case 4	Case 5
Hmm	95.74	95.74	85.11	91.49	89.36
Proposed	100	97.87	91.49	97.89	93.62
	Case 6	Case 7	Case 8	Case 9	Avg
Hmm	89.36	100	97.87	87.23	92.43
Proposed	91.49	100	97.87	93.62	95.98

Learning time becomes also an important factor for investigating the performance of stochastic models. So, we compared the average learning time of the proposed method with that of the conventional method. For computing the learning time, we applied the proposed method to the recognition phase for recognizing 9 cases of driving data, and we also applied the conventional method to the recognition phase for recognizing 9 cases of driving data. A function of the recognition phase was used to calculate total learning time of each recognition method. The computed average learning time for recognizing one case of driving data with different samples was shown in <Figure 9>. As we observed, the learning time of proposed method decreases about 40% in comparison with learning time of conventional method. The reason is that the proposed method used total 6 HMMs for recognizing 9 cases of driving data and conventional HMM used 9 HMMs for recognizing 9 cases of driving data.

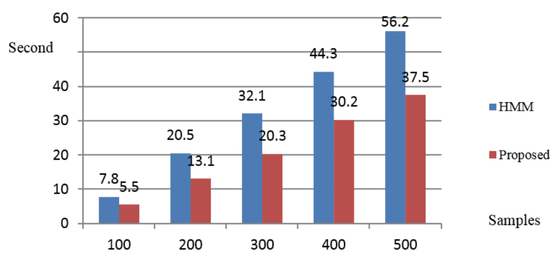


그림 9. 일반적인 HMM과 제안된 방법간의 학습시간 비교

Figure 9. Comparison learning time between conventional HMM and proposed method for recognizing one case of driving data.

4. Conclusions

In this paper, we proposed a double layer-HMM consisting of an upper layer and a lower layer for recognizing driving behavior using translational acceleration and angular velocity signals. The lower-layer was used for recognizing translational acceleration corresponding to gas pedal position state. The upper-layer HMM was used for recognizing angular velocity. Currently, we just assumed that each translational acceleration and angular velocity signals are discretized into three levels (increase/decrease/keep for translation acceleration, left/right/neutral position for steering wheel). Current work was focused on recognizing short-term driving such as increasing/decreasing/keep velocity or turning left/right/go straight. In future, we will expand our research for recognizing long-term driving by training another set of HMMs on short-term HMMs like combining with a brake pedal, a gearbox, and etc.

References

[1] Xiaoning Meng, Ka Keung Lee, and Yangsheng Xu, *Human driving behavior recognition based on hidden markov models*, Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, pp. 274-279, 2006.

[2] Wataru Takano, Akihiro Matsushita, Keijiro Iwao, and Yoshihiko Nakamura, *Recognition of human driving behaviors based on stochastic symbolization of time series signal*, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems Acropolis Convention Center Nice, pp. 167-172, 2008.

[3] Dejan Mitrovi´c, *Reliable method for driving events recognition*, IEEE Transactions on Intelligent Transportation Systems, Vol. 6 , No. 2, pp. 198 – 205, 2005.

[4] Holger Berndt, Jorg Emmert, and Klaus Dietmayer, *Continuous driver intention recognition with hidden markov models*, Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, pp. 1189-1194, 2008.

[5] Lisheng Jin , Haijing Hou and Yuying Jiang, *Driver intention recognition based on continuous hidden markov model*, 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), pp. 739 – 742, 2011.

[6] Pinar Boyraz, Amardeep Sathyanarayana, John H.L. Hansen, and Erik Jonsson, *Driver behavior modeling using hybrid dynamic systems for 'Driver-aware' active vehicle safety*, International Technical Conference on the Enhanced Safety of Vehicles (ESV)

[7] Amardeep Sathyanarayana, Pınar Boyraz, and John H.L. Hansen, *Driver behavior analysis and route recognition by hidden markov models*, Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety Columbus, pp. 276-281, 2008.

[8] Chiyomi Miyajima, Yoshihiro Nishiwaki, Koji Ozawa, Toshihiro Wakita, Katsunobu Itou, Kazuya Takeda, and Fumitada Itakura, *Driver modeling based on driving behavior and its evaluation in driver identification*, Proceedings of the IEEE, Vol. 95, No. 2, pp. 427-437, 2007.

[9] MD Rizal Othman, Zhong Zhang, Takashi Imamura and Tetsuo Miyake, *Modeling driver operation behavior by linear prediction analysis and auto associative neural network*, Proceedings of the 2009 IEEE International Conference on Systems, pp. 649-653, 2009.

[10] Christian Laugier, Igor E. Paromtchik, Mathias Perrollaz, and Mao Yong, John-David Yoder, Christopher Tay, Kamel Mekhnacha, and Amaury Nègre, *Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving*

safety, IEEE Intelligent Transportation Systems Magazine, Vol. 3, No. 4, pp. 4-19, 2011.

[11] Nuria Oliver and Alex P. Pentland, *Driver behavior recognition and prediction in a SmartCar*, CiteSeerX, 2000.

[12] Ryunosuke Hamada, Takatomi Kubo, Kazushi Ikeda, and Zujie Zhang, *A comparative study of time series modeling for driving behavior towards prediction*, Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-4, 2013.

[13] Wenshuo Wang, Junqiang Xi, and Huiyan Chen, *Modeling and recognizing driver behavior based on driving data: A survey*, Mathematical Problems in Engineering, Vol. 2014, pp. 1-20, 2014.

[14] L. R. Rabiner, *A tutorial on hidden markov models and selected application in speech recognition*, proceedings of the IEEE, Vol. 77, pp. 257-286, 1989.

[15] Anthony Danalis, Collin McCurdy, and Jeffrey S. Vetter, *Efficient quality threshold clustering for parallel architectures*, 2012 IEEE 26th International Parallel and Distributed Processing Symposium, pp. 1068-1079, 2012.

[16] G. Welch, G. Bishop, *An introduction to the Kalman Filter*, UNC-Chapel Hill, TR95-041, November 2000.

복층 은닉 마르코프 모델을 이용한 사람의 운전 습관의 효율적인 인식 방법

신은재, 윤인호, 딘 수안 하오, 김상연, 정구철
인터랙션 연구실, 첨단기술연구센터, 한국기술교육대학교

요 약

본 연구에서는 복 층의 은닉 마르코프 모델(Hidden Markov Model, HMM)을 이용하여 사람의 운전습관을 파악하기 위한 확률론적 모델 (stochastic model)을 제안한다. 제안하는 모델에서 운전페달 (gas pedal)의 위치, 자동차 스티어링 휠의 각도, 선속도와 각속도가 운전 태스크를 인식하기 위한 신호로 사용되며 매 100msec 마다 기록된다. 데이터의 획득은 모의 운전 태스크 동안 얻어진다. 얻어진 데이터는 9개의 다른 경우들로 세그먼트 (segment)되고 분류(cluster)된다. 복층 은닉 마르코프 모델에서 1자유도의 연속적인 HMM의 하위 층은 대상 자동차의 선 가속도를 인식하기 위해 사용된다. 또한, 제안하는 모델에서 상위 층은 대상 자동차의 각속도를 인식하기 위해서 사용된다. 사용자의 운전 습관을 인식하기 위해 본 연구에서는 10개의 샘플을 가진 슬라이딩 윈도우(sliding window)를 이용하였으며 노이즈를 감소시키기 위해 칼만 필터를 적용한다. 필터링 과정을 수행한 후, 데이터는 페달, 스티어링 휠, 스피드 의 세 그룹으로 정렬하여 처리한다. 또한 본 연구에서는 사용자의 운전 습관의 파악을 위해 각속도와 선 가속도의 특징을 적용한다. 본 연구의 결과를 평가하기 위해 로지텍 레이싱 플랫폼 (Logitech G27 Racing platform)을 이용하여 운전 시뮬레이터를 구성한다. 제안하는 방법의 정확도와 기존의 방법의 정확도를 파악하기 위해 개발한 운전 시뮬레이터를 이용하여 몇 개의 실험을 수행한다. 일반적으로 HMM과 같은 확률론적 모델(stochastic model) 에서는 학습시간이 중요한 요소가 되므로 본 연구에서는 기존의 방법과 제안하는 방법의 평균 학습시간을 비교한다. 제안하는 방법을 이용하였을 경우 실험결과는 약 96~97%정도의 인식률로 높은 효율성을 가지는 것을 알수 있다. 또한 제안하는 방법은 학습시간을 약 40% 단축시켜줌을 확인한다.

Eun-Jae Shin received a B.S. (2014) from the Korea University of Technology and Education. She is undertaking a master course as a member of the Interaction at Korea University of Technology and Education. Her current research interests include Haptics, Vibrotactile Actuators, and Dielectric polymer actuators.

E-mail address: ejshin@koreatech.ac.kr

In-Ho Yoon received a B.S. (2012) from the Korea University of Technology and Education. And now he is undertaking a master and doctoral course as a member of the Interaction lab at Korea University of Technology and Education. His current research interests include Haptics, Kinesthetic Actuators, Torque feedback actuators.

E-mail address: dogooly@koreatech.ac.kr

Dinh Xuan Hao received the B.S. in Department of Information technology from Hanoi University of Science and Technology in 2007. And now he is undertaking a master course as a member of the Interaction lab at Korea University of Technology and Education. His research interests include Virtual Reality and Pattern Recognition.

E-mail address: xuanhaobk@yahoo.co.kr

Sang-Youn Kim received a B.S. (1994) from the Korea University, Korea and an M.S.E. (1996) and a Ph.D. (2004) in mechanical engineering at Korea Advanced institute of Science and Technology (KAIST). From 2004 to 2005, he was a researcher at Human Welfare Robot System Research Center. In 2005, he was a research staff at Samsung Advanced Institute of Technology (SAIT). He is currently a professor of internet-media engineering at Korea University of Technology and Education. His current research interests include Human Computer Interaction, Virtual Reality, and Haptics.

E-mail address: sykim.kut.ac.kr

Goo-Cheol Jeong received the bachelor's degree in the Department of Electronic Engineering from the ChungAng University in 1979. He received the M.S. degree and the Ph.D. degree in the Department of Electronic Engineering from ChungAng University in 1981 and 1988, respectively. From 1982 to 1984, he was a researcher at Kia Motor Research Institute. He was a professor in the Department of Electronic Engineering at Dongseoul College from 1991 to 1993. He has been a professor in the School of Computer Science and Engineering, at Korea University of Technology and Education since 1993. His current research interests include Web based courseware, Numerical analysis of Antenna etc. He is a member of the KKITS.

E-mail address: jeong@kut.ac.kr