Driver Behavior Recognition using Recurrent Neural Network in Multiple Depth Cameras Environment

Ying-Wei Chuang¹, Chien-Hao Kuo¹, Shih-Wei Sun², and Pao-Chi Chang¹
¹Department of Communication Engineering, National Central University, Taoyuan, Taiwan
²Department of New Media Art, Taipei National University of the Arts, Taipei, Taiwan

Abstract
To improve the driving safety triggered by driver's behavior recognition in an in-car environment, we propose to use depth cameras mounted in a car to generate behavior models generated by a deep learning algorithm for a driver's behavior classification. The contribution of this paper is trifold: 1) The proposed multi-view driver behavior recognition system can handle the occlusion problem happened in one of the cameras; 2) Using the recurrent neural network can effectively recognize the continuous time behavior; 3) the average recognition accuracy of proposed systems can achieve 83% and 88%, respectively.

Introduction
A driver's behavior plays an important role to affect the traffic safety. For example, answering a phone, watching a video, or chatting with the people with a head turning behavior often lead the following car accidents. To increase a driving safety, a driver's behavior is analyzed, understood, and recognized [1, 2] to assist a driver to behave in a proper manner in a car. For example, Jain et al. [2] proposed to utilize cameras to understand a driver's behavior in an in-vehicle environment. However, it is challenging to use an in-car camera for behavior recognition due to the light changing, occlusion, and the clutter issues. Furthermore, in a limited in-car space, as shown in Fig. 1 (a), mounting positions of a camera to capture a driver's behavior is also very limited. Based on a limited mounting position, the captured content of a frame leads severe self-occlusion issue, as shown in Fig. 1 (b).

Related Work
To recognize human behavior using a Kinect depth camera, Hussein et al. [4] proposed a 3D joint covariance descriptor which employs the angular relationships among joint vectors with the linear support vector machine (SVM) classifier for recognizing actions. By adopting MoCap and a Kinect depth camera, Wang et al. [5] extracted 3D joint features and used local occupancy patterns to generate spatial histograms for behavior recognition, using SVMs to train the classifiers. Yang and Tian [6] proposed the EigenJoints method based on a principal component analysis for behavior recognition. In addition to using a depth camera, human action recognition approaches [7, 8] adopt deep learning algorithms with a color camera.

On the other hand, to achieve behavior recognition, researchers paid attention to utilize multiple cameras to compensate the the occluded areas and out of observation rage issues in a single camera environment. Azis et al. [9] proposed a weighted averaging fusion algorithm for generating a multiview skeleton with extracted 3D joint features to train the behavior classifiers. Kuo et al. [11] proposed a time-variant skeleton vector projection scheme using multiple infrared-based depth cameras. The proposed occlusion-based weighting element generation can be employed to train SVM classifiers to recognize behaviors in a multiple view environment.

In the in-vehicle environment, to recognize a driver’s behavior becomes a challenging research issue in human action recognition due to the limited viewing angle and the clutter environment in a car. To name a few, Xing et al. [12] used a Kinect camera to match the FFNN network to identify driving and non-driving actions. On the other hand, Chuang et al. [13] used the relative position in the space of the skeleton to recognize the driving behavior.

Therefore, in this paper, we will focus on the driver behavior recognition in a single depth camera and a multiple depth cameras environments. In addition, deep learning algorithms will be adopted for training a proper model for classification. Furthermore, the computational complexity for adopting different architectures of deep learning algorithms will be compared and discussed.
Proposed Driver Behavior Recognition System

In this paper, a driver behavior recognition system is proposed, as shown in Fig. 2. Basically, the proposed system is separated as a training stage and a testing stage. In the training stage, the training data needed to be pre-processed and used for training the recurrent neural network (RNN) model. In the testing stage, when the testing data is pre-processed, the RNN model generated in the training stage is used for classification, which is applied for driver behavior recognition in the proposed system. The details will be described in the following subsections.

Recurrent Neural Network

To generate a model for behavior recognition, a conventional recurrent neural network (RNN) is adopted in this paper. As shown by the left part of Fig. 3, an RNN uses a sequential data (the green circle) with a memory state (the orange circle) and the generated hidden layers (blue circles) to decide the output (yellow circle). By extending from the conceptual architecture in the left part of Fig. 3, the sequential step-by-step flow chart is shown in the right part of Fig. 3. For example, at time instance $t_2$, the input vector is $x_2$ (the green circle in the bottom-middle part), the hidden layer $s_2$ (the central blue circle) is influenced by the memory state value $c_1$ (the orange circle, a copy from $s_1$) in the previous time instance at $t_1$. Meanwhile, the current hidden layer $s_2$ is made a copy to $c_2$ to be the input of $s_3$ in the next time instance $t_3$.

Skeleton Based Driver Behavior Recognition System

In the proposed driver behavior recognition system, as shown by the scenario in Fig. 1, Kinect cameras is mounted at the left and right of a driver to capture the skeleton data, according to the official Microsoft Kinect SDK 2.0 [3] with 25 skeletal joints, as shown in Fig. 4. As shown in Fig. 1 (b), because the lower body of the driver is occluded by an instrument panel, only the skeletal joints of the upper body of the driver is used as the input for generating the RNN model for behavior recognition. The pre-processing step in the proposed skeleton-based approach is to remove the skeletal joints not belonging to the upper body to be the input $X$ of an RNN.

Multiple Views Point Cloud Based Driver Behavior Recognition System

In the proposed driver behavior recognition, in stead of using a single camera, it is possible to mount a second Kinect camera in a car to compensate the occlusion issues and the out of the observation range issues (in a field of view). As shown in Fig. 6...
Recurrent Neural Network for Skeleton Based Driver Behavior Classification

(a), the 3D point cloud can be captured from two Kinect cameras. According to an operation by a homography matrix [16, 17], as shown in Fig. 6 (b), the point clouds from multiple views can be fused as a more complete 3D point cloud. Next, as shown in Fig. 6 (c), the background point cloud of a driver can be removed by setting a 3D region-of-interest. Finally, to achieve a reasonable RNN model generation target, the 3D points are uniformly downsampled to 10,000 points, as shown in Fig. 6 (d).

After the pre-processing step, 10,000 points of the point cloud are reserved for one frame. The 3D position values in x, y, and z axis are obtained for each point, according to the Kinect SDK. Therefore, $10,000 \times 3 = 30,000$ values are utilized as the input nodes of the LSTM neural network for a time instance. For example, to time instance $t_1$, 30,000 nodes are collected at frame $t_1$ of a Kinect camera, as by the left part shown in Fig. 7. In addition, for recognizing a driver’s behavior, total 30 frames are used for behavior observation. Furthermore, in the three layers LSTM neural network for generating an RNN model, the number of the nodes for hidden layers are set as 2048, 512, and 128 nodes, respectively.

Eventually, either the proposed skeleton-based approach or the multiple views point cloud based approach, the generated RNN models are utilized for driver’s behavior recognition, using the conventional LSTM [14] classification.

Experimental Results

In the experimental results, two Kinect v2 depth cameras with the resolution $512 \times 424$ to obtain the 3D point cloud to capture a driver’s behavior with the official Kinect SDK [3], and the depth data is served as the raw data. To simulate the in-car environment, as shown by the right Kinect camera in Fig. 8, it was positioned 1.1m away from the driver to capture the right view. On the other hand, the other Kinect camera was mounted 0.8m away to capture the left view. The preprocessing tasks include: skeleton obtaining, multi-camera point cloud calibration, background removal, and downsampling. In addition, Tensorflow 1.8.0 [18] is used to build the RNN model.

VAP Multiple Views Driver Behavior Dataset

In order to evaluate the proposed method, we generate a “VAP Multiple Views Driver Behavior Dataset” for evaluation. As shown in Fig. 9, ten volunteer users were invited to perform ten different behaviors for three times. As a result, $10 \times 10 \times 3 = 300$ video clips were generated, with a manually time-synchronization process. For example, Fig. 10 shows the consecutive skeleton and point cloud of a waving behavior after performing the preprocessing steps. In the evaluation, a leave-one-out cross-validation (LOOCV) is adopted. In our test, the data from the nine of the ten drivers are used for model training/validation, and the remained one driver data is used for testing the classification performance, with average classification accuracy displayed as follows.

![Figure 5. Recurrent Neural Network for Skeleton Based Driver Behavior Classification](image1)

![Figure 6. Preprocessing stage: (a) point cloud collected from different Kinect cameras. (b) calibration of point cloud from two views and merging point cloud. (c) removing background and extracting point cloud of body part, (d) downsampling the driver body point cloud.](image2)

![Figure 7. Recurrent Neural Network for Multiple Views Point Cloud Driver Behavior Classification](image3)

![Figure 8. Testing environment of the experimental results](image4)

![Figure 9. VAP Multiple Views Driver Behavior Dataset](image5)
**Single View Skeleton Based Driver Behavior Recognition**

At first, the joints of the skeleton data captured from a single Kinect camera is use for evaluation. As shown left bottom green circle in Fig. 3, the RNN input $X$ is set as 57 nodes, and the $x_1, x_2, x_3 \cdots$ is observed until $x_{60}$ to represent that a user’s behavior is observed during 60 frames. The learning rate and drop out is set to 0.0001 and 0.5 respectively.

By 10,000 times iterations for obtaining the RNN model, the average accuracy rate can achieve 0.83, ranging from 0.67 to 0.90, which is shown in Table 1. It is obvious that the “right side” behavior recognition result has higher accuracy than “left side” in Table 1, due to the left-driving setting has fewer self-occlusion issues with proper Kinect camera observation distance, about 1.0m falls into the rage of valid depth observation 0.5m – 3.5m from the infra-red based depth sensor. In other words, the skeletons observed from the “left side” camera is relatively noisy due to the too short distance, smaller than 0.5m with almost out of the valid observation range from the depth sensor.

Furthermore, as shown by the confusion matrix for different behaviors in Fig. 11, the behaviors “Turning right” and “Adjusting mirror” can be successfully classified with the accuracy as 0.97, but the behavior “Watching video” can be classified with a relatively lower accuracy as 0.70. The false classification result in “Watching video” is caused by the similar geometric skeleton distribution in “Look up” and “Waving left” from a single camera, due to certain self-occlusion issues and out of observation range issue from a single camera environment.

**Multiple Views Point Cloud Based Driver Behavior Recognition**

To compensate the limitation from a single view camera environment, according our proposed method, 3D point cloud captured from multiple views with Kinect cameras is used for performance evaluation. After the preprocessing stage, the RNN input $X$ is set as 30,000 nodes, and the $x_1, x_2, x_3 \cdots$ is observed until $x_{30}$ to represent that a user’s behavior is observed during 30 frames. The learning rate and drop out is set to 0.00001 and 0.5 respectively. As shown in Table 2, the average behavior recognition accuracy is proportional to the number of epochs. For example, when the epochs is set to 2,000, the average accuracy is 0.88.

Moreover, the confusion matrix of point cloud based multiple views setting is shown in Fig. 12. By comparing with the single view skeleton-based approach in Fig. 11, the accuracy in most of the behaviors are achieved near 1.00, but the “Horn” behavior was incorrectly classified as “Look up” behavior, because of the hand motion of honking horn is occluded by the steering wheel. In addition, the accuracy of “Turning right” is reduced to from 0.97 to 0.83, because of the too short distance (less than 0.5m) from the Kinect camera to the driver. As a result, by combining the depth information from the two views of Kinect cameras, some of the missing parts or occluded parts in one view can be compensated from the other view, and the behavior recognition accuracy can be improved.

<table>
<thead>
<tr>
<th>Driver</th>
<th>Camera position</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>right side</td>
</tr>
<tr>
<td>Person 1</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 2</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 3</td>
<td>0.73</td>
</tr>
<tr>
<td>Person 4</td>
<td>0.76</td>
</tr>
<tr>
<td>Person 5</td>
<td>0.87</td>
</tr>
<tr>
<td>Person 6</td>
<td>0.67</td>
</tr>
<tr>
<td>Person 7</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 8</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 9</td>
<td>0.83</td>
</tr>
<tr>
<td>Person 10</td>
<td>0.83</td>
</tr>
<tr>
<td>Average</td>
<td>0.83</td>
</tr>
</tbody>
</table>

**Figure 9.** VAP multiple view driver behavior dataset contain 10 persons and 10 behaviors: turning right, turning left, looking up, horn, texting, watching video, phone, waving right, waving left, and adjusting mirror.

**Figure 10.** Consecutive skeleton and point cloud of a waving behavior after preprocessing stage.
Table 2: Recognition Results Using RNN

<table>
<thead>
<tr>
<th>Driver</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person 1</td>
<td>0.80</td>
<td>0.63</td>
<td>0.77</td>
<td>0.83</td>
</tr>
<tr>
<td>Person 2</td>
<td>0.67</td>
<td>0.90</td>
<td>0.90</td>
<td>0.80</td>
</tr>
<tr>
<td>Person 3</td>
<td>0.90</td>
<td>0.90</td>
<td>0.09</td>
<td>0.97</td>
</tr>
<tr>
<td>Person 4</td>
<td>0.97</td>
<td>0.83</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 5</td>
<td>0.93</td>
<td>0.90</td>
<td>0.87</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 6</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 7</td>
<td>0.87</td>
<td>0.90</td>
<td>0.87</td>
<td>0.80</td>
</tr>
<tr>
<td>Person 8</td>
<td>0.83</td>
<td>0.90</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 9</td>
<td>0.87</td>
<td>0.90</td>
<td>0.87</td>
<td>0.90</td>
</tr>
<tr>
<td>Person 10</td>
<td>0.80</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Average</td>
<td>0.85</td>
<td>0.86</td>
<td>0.87</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Complexity Comparison

The proposed methods were executed on a computer, with an Intel 3.20-GHz CPU (Core i7), GTX 1080 Ti GPU, and 64Gb of RAM. The total computational time for the skeleton-based and the point-cloud-based approaches are shown in Table 3. By comparing the results in the first row and the second row, it is apparent that the skeleton-based approach spent much less time than the point-cloud-based approach. The main reason is that RNN input $X$ of the point-cloud-based approach needs 30,000 nodes, but the skeleton-based approach only needs 57 nodes. In addition, the computational cost for the total training time and the training time per 100 epochs from the point-cloud-based approach to the skeleton-based approach is about 100 times. However, the testing time is about 3 times, and the GPU memory usage is about 35 times, from the point-cloud-based approach to the skeleton-based approach. Therefore, in order to obtain the higher accuracy with the point cloud compensating property, the computational cost and the memory usage increasing is needed.

Table 3: Time complexity

<table>
<thead>
<tr>
<th>Feature</th>
<th>Total Training time</th>
<th>Training Time per 100 epochs</th>
<th>Testing time</th>
<th>GPU memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skeleton</td>
<td>111.55s</td>
<td>1.12s</td>
<td>0.02s</td>
<td>247MiB</td>
</tr>
<tr>
<td>Point cloud</td>
<td>2392.59s</td>
<td>119.63s</td>
<td>0.06s</td>
<td>8401MiB</td>
</tr>
</tbody>
</table>

Conclusion

In conclusion, we proposed two approaches for a driver behavior recognition: a skeleton-based approach and a multiple views point cloud based approach, based on Kinect depth cameras. The recurrent neural network models based on LSTM algorithm is adopted for training the behavior models in the proposed approaches. In the experimental results, the driver behavior recognition accuracy can achieve 83% and 88%, respectively. In the future, the proposed driver behavior recognition scheme can be applied in an in-vehicle environment. Furthermore, wearable sensors on a driver and the sensors mounted on cars can be also utilized for driver behavior recognition. In the future, the proposed driver behavior recognition is possible to be adopted in the development of advanced driver assistance systems (ADAS).

References


Author Biography

Ying-Wei Chuang received B.S. and M.S. degree in communication engineering from National Central University (NCU), Taiwan, in 2016 and 2018 respectively. His research interests include Multi-view video/image processing.

Chien-Hao Kuo received B.S. and PhD degree in communication engineering from National Central University (NCU), Taiwan, in 2009 and 2018 respectively. His research interests include video/image processing, video compression.

Shih-Wei Sun received the B.S. degree from Yuan-Ze University and Ph.D. degree from National Central University, Taiwan, in 2001 and 2007, respectively, both in Electrical Engineering. From 2007 to 2011, he was a post doctoral research fellow at the institute of information science, Academia Sinica. Since 2012, he is an assistant professor at the Department of New Media Art, Taipei National University of the Arts, Taiwan, where he currently leads the Ultra-Communication Vision Laboratory (ucVision Lab). His research interest includes: visual content analysis, computer vision and the application for interactive technologies, 3D signal processing for depth cameras, multi-camera scene analysis and synthesis, and the security for network and multimedia. He published more than 30 international journal papers (SCI) and conference papers. He serves as the reviewers and technical program committee (TPC) members for many international SCI journals and academic conferences.

Pao-Chi Chang received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, and the Ph. D. degree from Stanford University, California, 1986, all in electrical engineering. From 1986 to 1993, he was a research staff at IBM T.J. Watson Research Center, New York. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, and multimedia retrieval.
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!

JOIN US AT THE NEXT EI!

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org