Article

Driver Distraction Recognition Using Wearable IMU Sensor Data

Wencai Sun, Yihao Si , Mengzhu Guo and Shiwu Li

School of Transportation, Jilin University, 5988 Renmin Street, Changchun 130022, China; sunwencai@jlu.edu.cn (W.S.); siyh19@mails.jlu.edu.cn (Y.S.); shiwu@jlu.edu.cn (S.L.)

* Correspondence: guomz@jlu.edu.cn; Tel.: +86-137-5693-0850

Abstract: Distracted driving has become a major cause of road traffic accidents. There are generally four different types of distractions: manual, visual, auditory, and cognitive. Manual distractions are the most common. Previous studies have used physiological indicators, vehicle behavior parameters, or machine-vision features to support research. However, these technologies are not suitable for an in-vehicle environment. To address this need, this study examined a non-intrusive method for detecting in-transit manual distractions. Wrist kinematics data from 20 drivers were collected using wearable inertial measurement units (IMU) to detect four common gestures made while driving: dialing a hand-held cellular phone, adjusting the audio or climate controls, reaching for an object in the back seat, and maneuvering the steering wheel to stay in the lane. The study proposed a progressive classification model for gesture recognition, including two major time-based sequencing components and a Hidden Markov Model (HMM). Results show that the accuracy for detecting disturbances was 95.52%. The accuracy associated with recognizing manual distractions reached 96.63%, using the proposed model. The overall model has the advantages of being sensitive to perceptions of motion, effectively solving the problem of a fall-off in recognition performance due to excessive disturbances in motion samples.

Citation: Sun, W.; Si, Y.; Guo, M.; Li, S. Driver Distraction Recognition Using Wearable IMU Sensor Data. *Sustainability* **2021**, *13*, 1342. <https://doi.org/10.3390/su13031342>

Academic Editor: Dongfang Ma
Received: 31 December 2020
Accepted: 24 January 2021
Published: 28 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: traffic safety; manual distraction; Dynamic Time Warping; wearable inertial measurement units; Hidden Markov Model

1. Introduction

Driver distraction and inattention is a common driving behavior that has become a growing public safety hazard. The National Highways Traffic Safety Administration (NHTSA) (2020) [1] found that 2841 people in the United States were killed because of distracted driving in 2018. More significantly, results from the 100-Car Naturalistic Driving Study [2] reported that almost 78% of crashes, and 65% of near-crashes, involved driver inattention. Driver inattention is primarily caused by secondary tasks that compete for mental and physical resources with the primary driving task [3]. The secondary task can take different forms, including visual distractions (e.g., looking away from the roadway), auditory distractions (e.g., listening to music), manual distractions (e.g., adjusting the radio volume), and cognitive distractions (e.g., becoming lost in thought) [4–6]. In recent years, drivers have started to multitask more often, given the extensive availability of in-vehicle intelligent systems, such as smartphones and navigation devices. These tools have made driver distraction and inattention much more common in everyday driving. In most kinds of distractions, a driver lets go of or keeps only one hand on the steering wheel to complete other tasks. This can lead the driver to look away from the roadway or to engage in lapses in attention and judgment. This can cause a decrease in critical information accessibility and processing capabilities required for safe driving, making the driver prone to accidents. Therefore, effectively detecting driver distractions is important for avoiding potentially dangerous situations and reducing the number of traffic accidents. Many studies have

proven that messages of traffic safety can help improve a driver's driving performance, traffic mobility, and transportation efficiency [7–11].

Many distracted driver detection methods have been proposed, by both academia and industry. The dominant information has come from different source categories, including physiological indicators, vehicle behavior parameters from controller area network (CAN), and machine-vision features.

Most driver distraction detection systems have used physiological data, such as skin conductance (Mehler et al. [12], Baldauf et al. [13]), heart rate variability (Miyaji et al. [14]), or an electromyogram (Lawrence et al. [15]). A few studies have also used other machine-vision features to detect driver distractions. For example, Zhang et al. [16] proposed using the Hidden Conditional Random Fields (HCRF) model to detect cellular phone-induced driver distraction; the method assesses the driver's face, mouth, and hand from images captured by a dashboard-mounted color camera. Hoang et al. [17] used a Faster-RCNN (Region Convolutional Neural Networks) approach to identify distracted states by detecting the presence of the driver's hands on or off the steering wheel and detecting cellular phone usage. Huang et al. [18] used a Fisher linear classifier and Back Propagation (BP) neural network to identify distracted states based on the driver's mouth. Harbluk et al. [19] proposed a driver inattention monitoring method based on eye-movements and visual scanning, to analyze the changes in visual behavior. The Saab Driver Attention Warning System [20] uses the driver's eyeball and head movement information to provide an alert when it detects a driver's inattentive state. Kuttila et al. [21] described a machine vision-based detection system, using stereo vision and lane tracking data to monitor driver distraction through rule-based and support vector machine (SVM) classification methods.

Other studies have used vehicle behavior parameters to identify whether a driver is distracted. For example, Yang et al. [22] proposed a Gaussian Mixture Model (GMM) using vehicle information collected using GPS to detect driver distractions. Wollmer et al. [23] used lane departure data and vehicle parameters as inputs to the Long Short-Term Memory (LSTM) model for detecting driver distraction. Jin et al. [24] applied driving performance measures from a CAN as inputs to SVM classifiers to identify a distracted driver. Torkkola et al. [25] constructed a distraction classifier with a Random Forest (RF), based on steering angle, accelerator pedal position, lane boundaries, and the upcoming road curvature. Aksjonov et al. [26] developed a system to detect and evaluate driver distractions, using driver performance information and analyzing the data with an effective fuzzy logic algorithm and machine learning algorithm.

However, these monitoring methods have some inherent limitations and are not suitable for an in-vehicle environment. Physiological-based methods require attaching special devices (e.g., electrodes) to the user's skin or eye. These are intrusive and interfere with drivers. Machine vision-based methods and vehicle behaviors-based methods are non-intrusive, but also have some limitations. Machine vision-based methods are affected by lighting variations in real-world applications and are costly. Vehicle behavior-based methods also have constraints, including the vehicle type and status, driver experiences, and driving conditions. This makes it difficult to implement these monitoring methods in conventional vehicles. As a result, there remains an unmet need to implement distracted driver detection algorithms that can be used for real-time traffic monitoring.

To respond to this need, smart wearable technologies [27], wearable sensor systems (WSS), wearable accelerometers, and inertial measurement units (IMUs) have been developed and are growing in popularity in other domains. For example, Hu et al. [28] used Dynamic Time Warping (DTW) to analyze the body kinematic data obtained from wearable IMUs. The device was able to classify a series of basketball activities, including shooting, passing, dribbling, and lay-ups. Yan et al. [29] developed a warning system for detecting dangerous operational patterns; the approach automatically assesses risky postures using motion data captured by wearable IMUs. Nouredanesh et al. [30] used machine learning-based models to detect a risk of falling, based on data obtained from the IMU and surface electromyography (sEMG) features. Borowska-Terka et al. [31] developed

a SVM model using an IMU around the forehead to distinguish different head gestures (e.g., yaw, pitch, roll, and immobility). Dalmazzo et al. [32] used inertial motion information from the right forearm to automatically classify different violin bow gestures using the Hierarchical Hidden Markov Models (HHMM).

Previous IMU applications indicate they can automatically capture gesture data in a precise and reliable way. A wearable IMU can help assess potential driving hazards using captured gesture data without disrupting driver manual operations because of their portability. Thus, IMU sensors have significant potential for use in the traffic safety domain. This makes them attractive for research about task-based distractions while driving, which is an understudied area [33].

The more commonly used recognition algorithm within the category of motion sensors-based gesture recognition algorithms is the Hidden Markov Model (HMM). The HMM provides a high level of recognition accuracy because it determines the recognition features through machine learning training. However, due to the large number of state transition matrices and observation matrices involved in the HMM, the mathematical calculation is quite complex. In a real-world driving environment, there are too many interference sequences. As such, the increase of gesture complexity would yield an exponential growth in the number of parameters. Further, too many disturbance sequences during actual driving would lead to serious algorithm lag and degradation of recognition performance, increase the computational cost, and weaken the durability of the hardware. However, it would be difficult to recognize gestures correctly if only a single HMM method was directly used.

To address the issues above, a progressive classification model for gesture recognition was developed, including two major time-based sequencing components and a Hidden Markov Model. The two major time-based sequencing components can be called the combined two-class subsequence matching method. The two-class subsequence matching method combined a time-domain analysis with a Dynamic Time Warping (DTW) algorithm to reduce excessive disturbance samples. The most likely probabilistic state transition sequence was calculated using a Baum Welch algorithm, designed to recognize the motion with the highest probability of occurring. This method theoretically accounts for the time-dependent characteristics of IMU sensor data compared to the single HMM method. The approach also reduces the sequence recognition scale of the HMM to maintain or improve the recognition accuracy.

Thus, this study proposes a task-specific detection and type identification method during driving, using the following elements: (1) wrist kinematics data generated by wearable IMUs, (2) a progressive classification model to detect excessive disturbance samples, and (3) establishing the HMM to recognize task-specific activities during effective driving.

2. Experiments

2.1. Participants

Twenty experienced Chinese drivers (14 males, 6 females), ages 21–35 (mean = 25.1, SD = 3.51), were recruited to participate in the study. All the drivers held a valid driving license, with 2 to 15 years of driving experience (mean = 6.2, SD = 3.64). All participants reported having no sleep disorders, and their corrected visual acuity was normal. All were medically evaluated prior to the study, were in good physical condition, and were not on medication.

On the day of the experiment, the participants were asked to not drink alcohol and to restrict tea and caffeine consumption. Prior to participating in the study, each subject signed a written informed consent in accordance with Chinese laws on scientific research using healthy human volunteers. Each participant was paid to be in the study.

2.2. Apparatus

We designed the hardware platform based on a wearable inertial measurement unit (IMU) (BWT901CL Module, Wit Intelligence Technology Co., Shenzhen, CHN). The IMU

includes a triaxial accelerometer, a triaxial gyroscope, and a triaxial magnetometer. Real-time sensor output was transferred to the attached application for data processing, and thirteen types of kinematic measures were generated from the raw data. The output frequency is adjustable from 0.1 Hz to 200 Hz. The platform's output frequency was set to 10 Hz. The platform has a size of $55 \times 40 \times 20 \text{ mm}^3$ and can accurately measure the following dimensions: 3-dimensional acceleration (measuring range: $\pm 16 \text{ g}$, precision: $\pm 0.01 \text{ g}$), 3-dimensional angular velocity (measuring range: $\pm 2000^\circ/\text{s}$, precision: $\pm 0.05^\circ/\text{s}$), 3-dimensional angle (measuring range: $\pm 180^\circ$) and four-dimensional quaternion algebra. Quaternion algebra, developed by Hamilton, is a mathematical number system that can provide an alternative representation of the three-dimensional space. In this paper, we also measured driver gesture data with quaternion format. The output information of the hardware platforms is shown in Table 1.

Table 1. The output information of the hardware platforms.

Abbreviations	Definitions
acc_x, acc_y, acc_z	x-, y-, z-axis accelerations (g)
omg_x, omg_y, omg_z	x-, y-, z-axis angular velocities ($^\circ/\text{s}$)
$ang_\alpha, ang_\beta, ang_\gamma$	Euler angles ($^\circ$)
q_0, q_1, q_2, q_3	quaternion algebra

The actual driving platform is the New Elantra EV, with a cab structure designed to have the driver's seat on the left. The driver's behavior and the surrounding environment were monitored and recorded in real-time using a CMOS camera (ZED2). Figure 1 shows the vehicle and experimental apparatus.

Figure 1. The experimental apparatuses.

2.3. Procedure

2.3.1. Driving Gesture Tasks

In this study, we instructed participants to engage in four common types of gestures that occur during driving. There were three types of manual distraction (dialing a hand-held cellular phone, adjusting the audio or climate controls, and reaching for an object in the back seat) and one regular driving motion (maneuvering the steering wheel to stay in the lane). All these activities are large amplitude wrist-movement tasks. The wrist-movement tasks were prescribed as follows:

Task 1—Dialing a hand-held cellular phone: The participants followed research assistant instructions to dial the hand-held cellular phone during the drive. The contents of these conversations were prepared in advance, consistent with real life.

Task 2—Adjusting audio or climate controls: All participants received information about the audio and climate controls and engaged in trials to become familiar with the controls before starting the actual driving experiments. During the experimental drive, participants followed research assistant instructions to perform a manual function, such as changing the song or adjusting the temperature.

Task 3—Reaching for an object in the back seat: The participants followed research assistant instructions to reach for an object in the back seat. The object was positioned to make it easy for the participant to perform this task only once.

Task 4—Steering wheel maneuvering to stay in the lane: Participants drove through the multi-curve road scenarios without any distractions, performing only steering wheel maneuvering related activities to stay in the lane.

2.3.2. Experimental Design

This driving experiments for this study was conducted at an unoccupied test site, with a size of 200 m × 300 m. Participants wore the IMU-based wearable platform on the right wrist to access wrist movements. After training to become familiar with the car, each participant drove the same designated routes seven times: five for exploratory experiments and two for verification experiments. Each wrist-movement task was performed two or three times by each participant during each experiment, the sequence of four movements was performed randomly. During the five exploratory experiments, research assistants tagged the sample types online and immediately saved the data after each wrist-movement task for later analysis and training. During two verification experiments, data for the complete sequence were acquired in combination with the motion images captured by the CMOS camera for later validation.

3. Methods

3.1. Data Preprocessing

The IMU raw data were processed with a sensor-fusion algorithm using an Extended Kalman Filter (EKF) approach. Thirteen types of kinematic measures were obtained from the raw data, forming the following kinematic parameter matrix:

$$Y = [acc_x, acc_y, acc_z, omg_x, omg_y, omg_z, ang_\alpha, ang_\beta, ang_\gamma, q_0, q_1, q_2, q_3]^T$$

The preprocessing method for the kinematic parameter matrix above was divided into two steps: (1) zero-mean normalization, and (2) baseline subtraction. First, the data used to train the model were normalized to reduce the effect of individual differences. This reduced errors in similarity comparisons between samples. Data entered into the model for training or prediction were also normalized. The standardized value (z-score) of each variable for each participant was calculated using the mean and standard deviation for each variable, as shown in Equation (1):

$$X_z = \frac{X_i - \mu}{\sigma} \quad (1)$$

where X_i is the raw input variable; μ is the mean of the variable; and σ is the standard deviation of the variable.

To eliminate the baseline offset variations in the gesture sequences, a baseline subtraction was performed using the analysis function in ORIGIN (Version 2018, OriginLab Co., Northampton, MA, USA). The connection between the beginning and end of the gesture sequence was selected as the baseline. Figure 2 shows an example of the q_0 time series (adjusting audio or climate controls) before and after the baseline subtraction. This preprocessing method was then applied to all the other kinematic parameters from the same experiment.

Figure 2. (a) An example of the time series (adjusting audio or climate controls) before the baseline subtraction; (b) An example of the time series (adjusting audio or climate controls) after the baseline subtraction.

3.2. Model Establishment

3.2.1. Quaternion-Based Gesture Subsequence Matching

To capture human motion and facilitate the analysis of motion trajectories, quaternions are often used to calculate the rotation angle of three-dimensional objects. Compared to matrices, quaternions are more efficient and take up less storage space. A quaternion consists of both real and imaginary parts, where the imaginary part is usually three-dimensional. As such, the quaternion is essentially an expression vector in a four-dimensional space. A quaternion is defined as:

$$q = q_0 + q_1i + q_2j + q_3k \quad (2)$$

where q_0 is the real part of the quaternion; q_1, q_2, q_3 are the imaginary part of the quaternion; i, j, k is the standard orthogonal basis in three-dimensional space; and $q_1i + q_2j + q_3k$ is a vector representing the axis of rotation in three-dimensional space.

The quaternion can be constructed using the rotation axis of the carrier and the angle of rotation around that axis. This involves establishing a vector in the arbitrary reference system O-XYZ by rotating α angle around rotation axis n . The rotation axis n is:

$$n = \cos(\beta_x)x + \cos(\beta_y)y + \cos(\beta_z)z \quad (3)$$

The quaternion is shown in Equation (4):

$$\begin{cases} q_0 = \cos(\frac{\alpha}{2}) \\ q_1 = \sin(\frac{\alpha}{2}) \cos(\beta_x) \\ q_2 = \sin(\frac{\alpha}{2}) \cos(\beta_y) \\ q_3 = \sin(\frac{\alpha}{2}) \cos(\beta_z) \end{cases} \quad (4)$$

where α is the angle of rotation around the rotation axis; and $\cos(\beta_x), \cos(\beta_y), \cos(\beta_z)$ are the components (direction cosines) that locate the rotation axis in the x, y, z directions.

Due to variability in driver performance, there are differences in the amplitude along the longitudinal axis of the time series. Equation (4) shows that q_1, q_2, q_3 changes as q_0 changes. Therefore, q_0 was selected as the sequence matching parameter for the proposed progressive classification model.

Due to the nature of the quaternion, there are two different ways to represent the arbitrary angular displacement in the three-dimensional space. Therefore, the quaternion q and $-q$ represent the same angular displacement and have the same geometric significance. Therefore, the unit quaternions must be geometrically transformed during data processing

to unify the quaternion symbols of the same group of gestures. Figure 3 shows that q_{01} and q_{02} are realistic sequences, and q_{02} is the transformed sequence.

Figure 3. An example of geometrical transformation for unit quaternions.

3.2.2. Template Creation

The purpose of a motion-based template is to increase the rate of recognition with respect to a specific action. We created different class templates for each action, with the data generated during the training stage. The class templates were constructed based on the selected kinematic features; that is, different numbers of the quaternion (i.e., q_0) for each time series of the training sample for each action. Each template consisted of relevant driving gesture categories, representing the unique characteristics of each driving task. When creating the class template, the data from each experiment were different in duration and timing, due to the intra-individual and inter-individual differences.

To address this problem of individual differences, the kinematic data from the training set were corrected using peak detection and an alignment algorithm. The location index of each peak of the selected kinematic measures was automatically detected using the peak-detection toolbox from MATLAB (Version 2020A, MathWorks Co., Natick, MA, USA). The selected kinematic measures were aligned along their time axes using the detected peak.

Afterwards, the aligned kinematic sequences were trimmed, to ensure that all sequences were of the same data length. For this purpose, the aligned peak in the data was considered to be the center of the sequence. The trimming started at the point where a kinematic sequence had the least number of data points, from the left side of the sequence to its peak. The trimming ended at the point where a kinematic sequence had the least number of data points from the peak to the right side of sequences. Finally, the average length of the trimmed sequence was calculated and served as the class template. The final class template is presented in Figure 4.

3.2.3. Subsequence Primary Matching Method Based on Time Domain Features

The combined two-class subsequence matching method was proposed as a pre-classifier in this paper. This included a subsequence primary matching method based on time domain features and subsequence secondary matching method based on dynamic time warping. Extracting the features related to each driving gesture movement was crucial to implement the first step of the proposed progressive gesture classification model. The disturbance sequences represent sequences that cannot be defined due to unknown wrist movements. A gesture pre-classifier, which distinguishes between task-specific gesture sequences and disturbance sequences in the full sequence set, can effectively reduce the number of disturbance samples and improve the accuracy of gesture recognition. This is done by selecting distinguishing and representative features based on wrist-movement characteristics. The time domain features of each wrist-movement can reflect the differ-

ences between gestures; as such, the subsequence primary matching method was proposed in this paper. We extracted three-time domain features (i.e., number of peaks, number of troughs, and the order of occurrence of peaks and troughs) to design a primary classification covering four classes of wrist-movements. The location and index of each peak of the selected kinematic measures was automatically detected using the peak-detection toolbox from MATLAB (Version 2020A, MathWorks Co., Natick, MA, USA). We then identified a primary subsequence set $\bar{q}_{0i} = [\bar{q}_{01}, \bar{q}_{02}, \dots, \bar{q}_{0n}]$ from the full sequence set $Y_i (1 \leq i \leq N)$ that matched the time domain features of q_0 .

Figure 4. The class template creation of different task-based driving gestures: (a) Reaching for object in the back seat; (b) Dialing hand-held cellular phone; (c) Steering wheel maneuvering to stay in lane; (d) Adjusting audio or climate controls.

3.2.4. Subsequence Secondary Matching Method Based on Dynamic Time Warping

The selected kinematic measures (i.e., q_0) from the testing data were compared with each class template using DTW, which generated a measurement known as an overall “distance”. For each wrist-movement task, the lower bound threshold value ϵ of the DTW distance was statistically determined. The value was then used to determine the validity of the sequence for subsequent identification and to eliminate invalid data. Thus, the DTW was used to calculate the “distance” between the kinematic measures of the “unknown” driving task and the corresponding class templates of the different target driving gestures.

The time complexity of the DTW algorithm is $O(mn)$, which means its calculation requires extensive computational resources, and the data volumes are very large. As such,

significant time is required to simultaneously match them. Therefore, this study statistically determined a DTW distance called D_{best} . The D_{best} was defined as the cumulative distance along the shortest warping path from the beginning of the sequence to the highest peak in the sequence using the DTW algorithm. While computing the lower bound threshold value ϵ , if the cumulative distance between each pair of corresponding datapoints exceeded the D_{best} before reaching the highest peak, then the calculation was stopped. If the full D_{best} needed to be calculated, we then calculated the full DTW by adding D_{best} with the partial DTW accumulation from the highest peak in the sequence to the ending of the sequence. This was then compared to the lower bound threshold value ϵ . After that, we found a secondary subsequence set $\hat{q}_0 = [\hat{q}_{01}, \hat{q}_{02}, \dots, \hat{q}_{0n}]$ on the primary subsequence set $\bar{q}_{0i} = [\bar{q}_{01}, \bar{q}_{02}, \dots, \bar{q}_{0n}]$ to calculate the two lower bounds (i.e., D_{best} and ϵ).

3.2.5. Gesture Recognition Systems Based on HMM

The Hidden Markov Model (HMM) is a popular algorithm for modeling human behavior because it is suitable for sequence data. In our proposed system, wrist-movement kinematic metrics $\hat{Y}$, generated after training by the two matching algorithms above, were used to train classifiers to recognize gestures. In this study, the HMM-based classifier was used for gesture classification, while the Baum-Welch estimation algorithm was used to train the model $\lambda = (\pi, A, B)$. Assume that $Q = \{q_1, q_2, \dots, q_N\}$ is the set of all possible states and $V = \{v_1, v_2, \dots, v_M\}$ is the set of all possible observations for a certain class of gestures. Assume also that N is the number of all possible states and M is the number of possible observations. The expression $I = (i_1, i_2, \dots, i_T)$ is the state sequence of length T ; $O = (o_1, o_2, \dots, o_T)$ is the corresponding observation sequence; and the parameters in model $\lambda = (\pi, A, B)$ are shown in Equation (5):

$$\begin{cases} \pi = \{\pi_i, i = 1, 2, \dots, N\} \\ A = \{a_{ij}, i = 1, 2, \dots, N, j = 1, 2, \dots, N\} \\ B = \{b_{jk}, j = 1, 2, \dots, N, k = 1, 2, \dots, M\} \end{cases} \quad (5)$$

where π_i denotes the initial state probability; a_{ij} denotes the probability that the state at time t is q_i and shifts to state q_j at time $t + 1$; and b_{jk} denotes the probability of generating an observation v_k while in state q_j .

The Baum-Welch algorithm process is divided into three steps (initialization, recursion, and termination) to determine the best model $\lambda = (\pi, A, B)$. In other words, the probability $P(O|\lambda)$ of generating an observed sequence under this model converges to a small value. The Baum-Welch method procedures are as follows:

(1) Initialization:

For $n = 0$, choose $a_{ij}^{(0)}, b_j(k)^{(0)}, \pi_i^{(0)}$, and generate the model $\lambda^{(0)} = (\pi^{(0)}, A^{(0)}, B^{(0)})$.

(2) Induction:

Two sets of probability variables, $\xi_t(i, j)$ and $\gamma_t(i)$, are introduced. The term $\xi_t(i, j)$ is the probability that the state is q_i at time t and the state is q_j at time $t + 1$; and $\gamma_t(i)$ is the probability of being in state q_i at time t . For $n = 1, 2, \dots$, as shown in Equations (6)–(8):

$$a_{ij}^{(n+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (6)$$

$$b_j(k)^{n+1} = \frac{\sum_{t=1, o_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (7)$$

$$\pi_i^{(n+1)} = \gamma_1(i) \quad (8)$$

The values on the right side of the equations above were calculated from observation $O = (o_1, o_2, \dots, o_T)$ and model $\lambda^{(n)} = (\pi^{(n)}, A^{(n)}, B^{(n)})$.

(3) Termination:

The model parameters $\lambda^{(n+1)} = (\pi^{(n+1)}, A^{(n+1)}, B^{(n+1)})$ were obtained once $P(O|\lambda)$ no longer increased significantly.

In this study, we defined an HMM for each gesture class and then trained the models only with the training sequence for the respective class. To apply the HMM for gesture recognition, the input observations used for training or classification needed to be discrete. Thus, the raw data were discretized using a K-Means algorithm. The discrete labels replaced the raw data as the new training compliance sequence.

There were N gesture classes in this study; therefore, there was a set of N HMMs, modeled as $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_N\}$. When a new sequence of gesture observations needed to be identified as $O = (O_1, O_2, O_3, \dots, O_T)$, it was necessary to evaluate the probability that this sequence is generated by a specific HMM λ_i , that is, the $P(O|\lambda_i)$ of all HMMs in the λ . Finally, we determined the HMM generating the maximum likelihood estimation of the observation sequence O , and used the gesture class represented by this HMM to label new gestures, as shown in Equation (9):

$$\text{RecognitionGesture} = \arg_{\text{gesture}} \max(P(O|\lambda_{\text{gesture}})) \quad (9)$$

Thus, the HMM can determine the probability that the model will generate the specific sequence, which can be used to determine the class label for the gesture performed.

Equations (10)–(12) were used to calculate the confidence level defined for each new gesture recognition:

$$P_{\max} = \max(P(O|\lambda_{\text{gesture}})) \quad (10)$$

$$P_{\text{second}} = \max(P(O|\lambda_{\text{gesture except recognized gesture}})) \quad (11)$$

$$\text{confidence} = \frac{1/P_{\text{second}} - 1/P_{\max}}{1/P_{\text{second}}} \quad (12)$$

where $P_{\max}$ is the maximum value of the probability that the new gesture sequence will be identified as a specific HMM λ_i ; and P_{second} is the second maximum value of the probability that the new gesture sequence will be identified as a specific HMM λ_i . The closer the confidence level was to 1, the more confidence there was that the data represented the gesture.

When fitting the model, the initialization elements of an HMM include two key parameters. The parameter N is the number of hidden states in the model, and the parameter M is the number of distinct observation symbols for each state (here, the K-Means cluster labels served as the symbols).

In this study, we selected different ranges of N values, but found that the confidence level did not correspondingly increase with the increase in the number of states. This led to instead mainly determining the M values. This study mitigated the problem of under-generalization and generalization by using 5-fold cross-validation to calculate the variation in the effect of M -values on recognition accuracy. Table 2 shows the relationship between the M value and the corresponding recognition accuracy. In Figure 5, the confidence curve shows an upward and then a downward trend. Therefore, the value of M was selected that generated the maximum mean confidence (i.e., $M = 120$). This determined parameters N and M in the training model.

Table 2. 5-fold cross-validation.

M	Evaluation Indicators	Fold-K				
		K = 1	K = 2	K = 3	K = 4	K = 5
30	Accuracy	78.57%	85.71%	84.62%	92.86%	80.00%
	Mean Confidence	0.6184	0.5981	0.7014	0.6438	0.5650
60	Accuracy	78.57%	71.43%	92.31%	100%	100%
	Mean Confidence	0.6357	0.67113	0.6614	0.7137	0.7191
90	Accuracy	78.57%	71.43%	92.31%	100%	100%
	Mean Confidence	0.6832	0.7189	0.7559	0.7597	0.8007
105	Accuracy	78.57%	71.43%	92.31%	100%	100%
	Mean Confidence	0.7388	0.7426	0.7807	0.7708	0.8140
120	Accuracy	78.57%	71.43%	92.31%	100%	100%
	Mean Confidence	0.8234	0.8236	0.7888	0.8253	0.8566
135	Accuracy	78.57%	78.57%	92.31%	100%	100%
	Mean Confidence	0.74431	0.8094	0.7796	0.8316	0.8482
150	Accuracy	78.57%	78.57%	84.61%	100%	100%
	Mean Confidence	0.6966	0.8007	0.7465	0.8293	0.8931

NOTE: Mean Confidence is the average of the confidence of any of the gesture classes that identify the correct sequence.

Figure 5. Confidence curves for different M values.

4. Results and Discussion

4.1. Time-Domain Analysis for Driving Gestures

As described above, different class templates were created for each wrist-movement. Each class template was constructed and tested based on selected kinematic measures (i.e., q_0). Each class template consisted of different numbers of kinematic time series, and represented the unique features of each driving gesture. Figure 6 shows the analysis for the different class templates for each driving gesture task. The peaks and troughs are shown in red and green, respectively.

4.2. Model Evaluation

4.2.1. Segmentation

This system was applied in a non-controlled environment. As such, it was important to ensure that only the user interacted with the wearable IMU. To intercept the gesture waveform from the continuous waveform, we applied a threshold interception method based on triaxial acceleration. The acceleration signal was smoother in the absence of gestures and changed significantly when gestures begin. Based on this property, we applied a specific threshold, and when the amplitude of the selected kinematic measure exceeded the threshold, the index position was determined to be the starting point of the

gesture. In contrast, when the amplitude of the selected kinematic measure was lower than the threshold, the index position was determined as the end point of the gesture.

This generated the starting and ending points of each gesture, and the gesture waveform was intercepted from the continuous waveform output from the accelerometer. The different values of triaxial accelerations were used to construct thresholds to obtain the starting and ending points of a task-specific gesture sequence in a full-sequence continuous waveform used for validation. The threshold was calculated as the cumulative total of difference values of $N = 10$ points within a 1 s time window of the data amplitude. Equation (13) shows the calculation of the cumulative total of different values:

$$Diff_{accN_{x,y,z}}(i) = \sum_{k=i}^{i+N} (|X_k^{acc} - X_{k-1}^{acc}| + |Y_k^{acc} - Y_{k-1}^{acc}| + |Z_k^{acc} - Z_{k-1}^{acc}|) \quad (13)$$

where $X_k^{acc}, Y_k^{acc}, Z_k^{acc}$ denote the acceleration values in the x-axis, y-axis, and z-axis at time t .

Figure 6. Time-domain analysis for different class templates for driving gesture tasks: (a) Reaching for object in the back seat; (b) Dialing hand-held cellular phone; (c) Steering wheel maneuvering to stay in lane; (d) Adjusting audio or climate controls.

4.2.2. Confusion Matrix

Validation tests confirmed the recognition performance and universality of the proposed algorithm. Specifically, validation datasets were correctly labelled using surveillance

videos of the driver's distracted motions. The validation dataset consists of 647 intercepted sequences, of which 102 were gesture sequences and 545 were disturbance sequences.

A combined two-class subsequence matching method was used to divide the gesture sequence state into two categories: Positive and Negative (i.e., gesture sequences and disturbance sequences). The gesture recognition systems based on HMM were used to divide the gesture sequence states into four categories (i.e., dialing hand-held cellular phone, adjusting audio or climate controls, reaching for object in back seat, steering wheel maneuvering to stay in lane).

A confusion matrix contains information about actual and predicted classifications completed by a classifier. The performance of such a classifier can be evaluated using the data in the matrix. The accuracy, precision, recall (also known as sensitivity), F-measure, and specificity were calculated for each identification stage, and the overall model performance was evaluated, using Equations (14)–(18):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$Specificity = \frac{TN}{TN + FP} \quad (17)$$

$$F_Measure = \frac{2Precision \cdot Recall}{Precision + Recall} \quad (18)$$

where *TP* (true positives) represents the number of correct identifications of gesture sequences; *TN* (true negatives) represents the number of correct identifications of disturbance sequences; *FP* (false positives) represents the number of incorrect classifications of gesture sequences as disturbance sequences; and *FN* (false negatives) is the number of incorrect identifications of disturbance sequences as gesture sequences.

Table 3 shows the confusion matrix for the results of the screening for negatives, using the subsequence matching method. For the first subsequence matching method, the number of true positives (TP) was 88; the number of false positives (FP) was 92; the number of true negatives (TN) was 453; and the number of false negatives (FN) was 14. For the second subsequence matching method, there were 81 true positives (TP); there were 8 false positives (FP); there were 84 true negatives (TN), and there were 7 false negatives (FN). The negative screenings for the first subsequence matching method were better than the second method. The first resulted in 51.11% of the total sample proportion of the second subsequence matching method negative samples and 48.89% of the total sample proportion of the positive samples after screening. The combined two-class subsequence matching method effectively screened the overall disturbance sequences, with the proportion of positive samples increasing from 15.76% to 91.01% and the proportion of interference samples decreasing from 84.23% to 8.98% of the total samples.

Table 3. The confusion matrix for the results of the negative sample screenings.

Classification	Sequence State (Actual Class)	Predicted Class	
		Condition Positive	Condition Negative
The first subsequence matching method	test outcome positive	88	92
	test outcome negative	14	453
The second subsequence matching method	test outcome positive	81	8
	test outcome negative	7	84
The combined two-class subsequence matching method	test outcome positive	81	8
	test outcome negative	21	537

Table 4 shows the subsequence matching method accuracy, precision, recall (also known as sensitivity), F-measure, and specificity. For the first subsequence matching method, the accuracy was 83.62%; the precision was 48.89%; the recall was 86.27%; the F-measure was 62.41%; and the specificity was 83.12%. The second subsequence matching method had a higher accuracy (91.67%) compared to the first, increasing by 9.63%. The second method also had a higher precision (91.01%), which was an increase of 86.16%, and a higher recall (92.05%), which increased by 6.69%. The second method also had a higher F-measure (91.53%), which was an increase of 4.66%, and a higher specificity level (91.30%), which was an increase of 9.85%.

Table 4. The results of the subsequence matching method.

Indicators	The First Subsequence Matching Method	The Second Subsequence Matching Method	The Combined Two-Class Subsequence Matching Method
Accuracy	83.62%	91.67%	95.52%
Precision	48.89%	91.01%	91.01%
Recall	86.27%	92.05%	79.41%
Specificity	83.12%	91.30%	98.53%
F-Measure	62.41%	91.53%	84.82%

The specificity of the combined two-class subsequence matching method increased significantly to 98.53%; the recall was 79.41%; and the precision was 91.01%. These results indicate that the overall model was more sensitive to positive case perception and could screen for disturbance sequences. The F-measure was 84.82%, indicating that the model was robust. Finally, the HMM algorithm in the proposed progressive classification model for gesture recognition had an excellent detection rate (96.63%).

5. Conclusions

This paper proposed a non-intrusive method for detecting in-transit manual distractions, using body kinematic measures collected from wearable inertial measurement units (IMU). The goal was to detect specific gesture tasks and types of gestures during effective driving. Due to the problem of excessive disturbances in real driving scenarios, this study developed a progressive classifier to recognize gestures, which is a key contribution of the study. This study focused on establishing a combined two-class subsequence matching method to screen disturbance sequences, by analyzing gesture kinematic measures. The study also constructed a Hidden Markov Model (HMM) database to recognize gestures using the Baum-Welch algorithm.

First, we annotated an online driving gesture dataset, generated by wearable IMU units around participants' wrists to recognize gestures. Then, we extracted the time-domain features of the quaternions in the Kinematic Parameters and constructed a gesture template. Second, the unknown motion samples were compared with the selected kinematic signatures, based on time-domain analysis and an improved Dynamic Time Warping (DTW) algorithm. Finally, the raw data from the driving gesture dataset under each motion gesture was discretized using K-Means to train the models. By establishing a Hidden Markov Model, the corresponding parameters were defined using a 5-fold cross-validation. The most likely probabilistic state transition sequence was calculated using the Baum Welch algorithm to recognize the motion with the highest possibility.

The class templates of different driving gesture tasks were created based on selected kinematic measures. The features were easily extracted using clear peaks and drops, or patterns in the movement. The accumulated distance (D_{best} and ϵ) between the testing data and the template was considered a cost function, reflecting the similarity between them. Therefore, the model could record complete data about the body movement kinematics during driving, and subtle changes could be relied on to screen the disturbance samples.

The evaluation indicators (accuracy, precision, recall, F-measure, and specificity) from confusion matrices were used to evaluate the performance of the proposed model.

The results show that the specificity of the combined two-class subsequence matching method in the progressive classification model was 98.53%; the recall was 79.41%; and the precision was 91.01%. These results indicate that the model was highly efficient in screening disturbance sequences. The model achieved a 96.63% accuracy level with respect to HMM dynamic gesture recognition. Overall, the progressive classification model demonstrated a high level of accuracy (95.52%) in separating disturbance sequences from gesture sequences during driving. The model also showed good accuracy (96.63%) in distinguishing task-specific wrist-movement categories. The advantage of this method is that the overall model is sensitive to motion perception, and effectively addresses the problem of a fall-off in recognition performance due to excessive disturbance samples. Moreover, the proposed model is easy to implement and provides high recognition accuracy.

The results of this study can be popularized and implemented in the field of human-machine cooperative driving. Intelligent vehicles with assistive driving now represent a mainstream driving pattern and will continue in the long term. The driver remains the main subject, and the real-time monitoring of driving behavior and states is indispensable in shared autonomy scenarios. Therefore, the most important application of this study may be its ability to provide useful guidance for future driving risk assessments. This supports improvements in the safety, efficiency, and sustainability of the transportation system. It may also support the successful development and implementation of distracted driver detection algorithms to advance the sustainability of road systems in a connected vehicle environment. Future research will focus on extracting data from different driving gesture tasks during real driving to validate the model and examine more gesture motions for better generalization. Combined with the future development of Intelligent Connected Vehicle technologies, data processing and transmission capacity will continue to be enhanced, based on real-time monitoring of driver behavior and driving status. This will continue to improve early warning information, and may support future applications, including planning the timing of a driver's takeover of an autonomous vehicle.

Author Contributions: Conceptualization, Y.S. and S.L.; methodology, Y.S.; software, W.S.; validation, M.G.; formal analysis, Y.S.; investigation, W.S.; resources, S.L.; data curation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, W.S. and M.G.; visualization, Y.S.; supervision, M.G.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jilin Province Transportation Science and Technology Project with grant number 2019-1-16, the Jilin Province Science and Technology Development Plan Project with grant number 20180101074JC, the Youth Program of National Natural Science Foundation of China with grant number 52002143, the General Program of National Natural Science Foundation of China with grant number 71971097 and supported by Exploration Foundation of State Key Laboratory of Automotive Simulation and Control.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. National Highway Traffic Safety Administration. Available online: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812926> (accessed on 6 November 2020).
2. Klauer, S.G.; Dingus, T.A.; Neale, V.L.; Sudweeks, J.D.; Ramsey, D.J. The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data. Available online: <https://trid.trb.org/view.aspx?id=786825> (accessed on 18 June 2020).

3. Hurts, K.; Angell, L.S.; Perez, M.A. The Distracted Driver: Mechanisms, Models, and Measurement. *Rev. Hum. Factors Ergon.* **2011**, *7*, 3–57. [\[CrossRef\]](#)
4. Breen, J. Car Telephone Use and Road Safety: Final Report. Available online: <https://trid.trb.org/view/1149617> (accessed on 18 May 2020).
5. Ranney, T.A.; Garrott, W.R.; Goodman, M.J. Nhtsa Driver Distraction Research: Past, Present, and Future. Available online: <https://trid.trb.org/view/711046> (accessed on 20 July 2020).
6. Wang, C.; Li, Z.; Fu, R.; Guo, Y.; Yuan, W. What is the difference in driver's lateral control ability during naturalistic distracted driving and normal driving? A case study on a real highway. *Accid. Anal. Prev.* **2019**, *125*, 98–105. [\[CrossRef\]](#)
7. Bie, Y.; Xiong, X.; Yan, Y.; Qu, X. Dynamic headway control for high-frequency bus line based on speed guidance and intersection signal adjustment. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 4–25. [\[CrossRef\]](#)
8. Wang, S.; Zhang, W.; Bie, Y.; Wang, K.; Diabat, A. Mixed-integer second-order cone programming model for bus route clustering problem. *Transp. Res. Part C-Emerg. Technol.* **2019**, *102*, 351–369. [\[CrossRef\]](#)
9. Ma, D.; Xiao, J.; Ma, X. A decentralized model predictive traffic signal control method with fixed phase sequence for urban networks. *J. Intell. Transp. Syst.* **2020**. [\[CrossRef\]](#)
10. Ma, D.; Xiao, J.; Song, X.; Ma, X.; Jin, S. A Back-Pressure-Based Model With Fixed Phase Sequences for Traffic Signal Optimization Under Oversaturated Networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–12. [\[CrossRef\]](#)
11. Ma, D.; Song, X.; Li, P. Daily Traffic Flow Forecasting Through a Contextual Convolutional Recurrent Neural Network Modeling Inter-and Intra-Day Traffic Patterns. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–10. [\[CrossRef\]](#)
12. Mehler, B.; Reimer, B.; Coughlin, J.F.; Dusek, J.A. Impact of Incremental Increases in Cognitive Workload on Physiological Arousal and Performance in Young Adult Drivers. *Transp. Res. Rec.* **2009**, 6–12. [\[CrossRef\]](#)
13. Baldauf, D.; Burgard, E.; Wittmann, M. Time perception as a workload measure in simulated car driving. *Appl. Ergon.* **2009**, *40*, 929–935. [\[CrossRef\]](#)
14. Miyaji, M.; Kawanaka, H.; Oguri, K.; IEEE. Driver's Cognitive Distraction Detection Using Physiological Features by the AdaBoost. In Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, 4–7 October 2009; pp. 1–6. [\[CrossRef\]](#)
15. Schleifer, L.M.; Spalding, T.W.; Kerick, S.E.; Cram, J.R.; Ley, R.; Hatfield, B.D. Mental stress and trapezius muscle activation under psychomotor challenge: A focus on EMG gaps during computer work. *Psychophysiology* **2008**, *45*, 356–365. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Zhang, X.; Zheng, N.; Wang, F.; He, Y. Visual recognition of driver hand-held cell phone use based on hidden CRF. In Proceedings of the 2011 IEEE International Conference on Vehicular Electronics and Safety, Beijing, China, 10–12 July 2011; pp. 248–251. [\[CrossRef\]](#)
17. Le, T.H.N.; Zheng, Y.; Zhu, C.; Luu, K.; Savvides, M.; IEEE. Multiple Scale Faster-RCNN Approach to Driver's Cell-phone Usage and Hands on Steering Wheel Detection. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 46–53. [\[CrossRef\]](#)
18. Huang, X.; Zhang, B. Research on Method of Driver Distraction State Based on Mouth State. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 301–304. [\[CrossRef\]](#)
19. Harbluk, J.L.; Noy, Y.I.; Trbovich, P.L.; Eizenman, M. An on-road assessment of cognitive distraction: Impacts on drivers' visual behavior and braking performance. *Accid. Anal. Prev.* **2007**, *39*, 372–379. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Saab Driver Attention Warning System. Available online: www.saabnet.com/tsn/press/071102.html (accessed on 26 June 2020).
21. Kuttila, M.; Jokela, M.; Markkula, G.; Rue, M.R. Driver distraction detection with a camera vision system. In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 16–19 September 2007; pp. VI-201–VI-204. [\[CrossRef\]](#)
22. Yang, J.; Chang, T.N.; Hou, E. Driver Distraction Detection for Vehicular Monitoring. In Proceedings of the Iecon 2010: 36th Annual Conference Of the IEEE Industrial Electronics Society, Glendale, AZ, USA, 7–10 November 2010; pp. 108–113. [\[CrossRef\]](#)
23. Woellmer, M.; Blaschke, C.; Schindl, T.; Schuller, B.; Faerber, B.; Mayer, S.; Trefflich, B. Online Driver Distraction Detection Using Long Short-Term Memory. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 574–582. [\[CrossRef\]](#)
24. Jin, L.; Niu, Q.; Hou, H.; Xian, H.; Wang, Y.; Shi, D. Driver Cognitive Distraction Detection Using Driving Performance Measures. *Discret. Dyn. Nat. Soc.* **2012**. [\[CrossRef\]](#)
25. Torkkola, K.; Massey, N.; Wood, C. Driver inattention detection through intelligent analysis of readily available sensors. In Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749), Washington, WA, USA, 3–6 October 2004; pp. 326–331. [\[CrossRef\]](#)
26. Aksjonov, A.; Nedoma, P.; Vodovozov, V.; Petlenkov, E.; Herrmann, M. Detection and Evaluation of Driver Distraction Using Machine Learning and Fuzzy Logic. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2048–2059. [\[CrossRef\]](#)
27. Fang, Y.-M.; Chang, C.-C. Users' psychological perception and perceived readability of wearable devices for elderly people. *Behav. Inf. Technol.* **2016**, *35*, 225–232. [\[CrossRef\]](#)
28. Hu, X.; Mo, S.; Qu, X. Basketball Activity Classification Based on Upper Body Kinematics and Dynamic Time Warping. *Int. J. Sports Med.* **2020**, *41*, 255–263. [\[CrossRef\]](#)
29. Yan, X.; Li, H.; Li, A.R.; Zhang, H. Wearable IMU-based real-time motion warning system for construction workers' musculoskeletal disorders prevention. *Autom. Constr.* **2017**, *74*, 2–11. [\[CrossRef\]](#)

30. Nouredanesh, M.; Tung, J. IMU, sEMG, or their cross-correlation and temporal similarities: Which signal features detect lateral compensatory balance reactions more accurately? *Comput. Methods Programs Biomed.* **2019**, *182*. [[CrossRef](#)]
31. Borowska-Terka, A.; Strumillo, P. Person Independent Recognition of Head Gestures from Parametrised and Raw Signals Recorded from Inertial Measurement Unit. *Appl. Sci.* **2020**, *10*, 4213. [[CrossRef](#)]
32. Dalmazzo, D.; Ramirez, R. Bowing Gestures Classification in Violin Performance: A Machine Learning Approach. *Front. Psychol.* **2019**, *10*. [[CrossRef](#)]
33. Atiquzzaman, M.; Qi, Y.; Fries, R. Real-time detection of drivers' texting and eating behavior based on vehicle dynamics. *Transp. Res. Part F-Traffic Psychol. Behav.* **2018**, *58*, 594–604. [[CrossRef](#)]