Accident Analysis and Prevention 96 (2016) 162–168

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Identification of safety-critical events using kinematic vehicle data and the discrete fourier transform

Robert Kluger[a,*], Brian L. Smith[b], Hyungjun Park[b], Daniel J. Dailey[c]

[a] University of Virginia, Department of Civil and Environmental Engineering, 351 McCormick Road, Charlottesville, VA, 22903, United States
[b] University of Virginia, United States
[c] University of Washington, United States

ARTICLE INFO

Article history:
Received 10 March 2016
Received in revised form 25 May 2016
Accepted 4 August 2016
Available online 17 August 2016

Keywords:
Crashes
Near-crashes
Safety-critical event
Naturalistic driving study
Discrete fourier transform

ABSTRACT

Recent technological advances have made it both feasible and practical to identify unsafe driving behaviors using second-by-second trajectory data. Presented in this paper is a unique approach to detecting safety-critical events using vehicles' longitudinal accelerations. A Discrete Fourier Transform is used in combination with K-means clustering to flag patterns in the vehicles' accelerations in time-series that are likely to be crashes or near-crashes. The algorithm was able to detect roughly 78% of crasjavascript:void(0)hes and near-crashes (71 out of 91 validated events in the Naturalistic Driving Study data used), while generating about 1 false positive every 2.7 h. In addition to presenting the promising results, an implementation strategy is discussed and further research topics that can improve this method are suggested in the paper.

1. Introduction

High resolution, kinematic vehicle data (second-by-second speed, acceleration, yaw, etc.) is becoming more available than ever in the transportation community. With this influx of data, there are a considerable number of potential benefits to a wide range of safety applications, including monitoring driver performance, identifying unsafe locations on the road (hot spots), or even providing real-time emergency response. However, before these benefits can be realized, there is a need to be able to identify unsafe driving activities, like crashes and near-crashes, amongst a vast amount of regular driving.

The goal this paper to is to develop ways to identify safety-critical events (SCEs), defined in this context as crashes, near-crashes, and other unsafe driving behaviors using kinematic data from single vehicles. Creating an algorithm that can detect SCEs using only the trajectories of single vehicles could have a variety of applications including:

- Allowing infrastructure providers to identify SCEs in connected vehicle environments and evaluate road network safety
- Allowing taxis and shared ride service providers to monitor their drivers and ensure they provide safe rides to customers
- Allowing insurance companies to monitor their customers' driving tendencies and better evaluate risk
- Allowing agencies to monitor fleet vehicles (e.g. buses, snow plows, etc.) for both driver performance and tort liability claims
- Allowing transportation management agencies to monitor traffic and provide emergency response when necessary
- Providing real-time alerts to emergency response services in connected vehicle environments
- Identifying events in large-scale naturalistic driving studies

Each of these applications is slightly different and will likely require varying inputs to a method or algorithm when identifying SCEs but there is a clear benefit to a variety of stakeholders by having the ability to identify them.

Many established methods for identifying unsafe driving, whether it be SCEs, or a specific subset of SCEs, rely on information to be available describing how one or more vehicles are interacting. One example of such information is Time-to-Collision (TTC), which is an estimate of how much time a vehicle has on its current trajectory before it would collide with a lead vehicle. This typically requires access to radar data, which can be expensive to equip on large fleets of vehicles. As a result the analysis was restricted to kinematic data that is native to connected vehicle standards (SAE

* Corresponding author.
E-mail address: rmk7su@virginia.edu (R. Kluger).

http://dx.doi.org/10.1016/j.aap.2016.08.006
0001-4575/

International, 2009) and can be collected from smart phones or aftermarket devices.

For this study, crash and near-crash data was acquired from the SHRP2 Naturalistic Driving Study (NDS) (Virgina Tech Transportation Institute, 2013). The methodology outlined performs subsequence-matching techniques on longitudinal accelerations observed in vehicles during a set of crashes and near-crashes. A Discrete Fourier Transform (DFT) is used to transform subsequences of the observed time-series and a K-means clustering algorithm is then used to classify those subsequences as events or baseline driving.

2. Research goals

The primary goal of this study is to develop a methodology for identifying safety critical events when given a high-resolution time series of kinematic vehicle data, specifically longitudinal acceleration. With recent advancements in vehicle and roadside technology, learning how to identify unsafe driving behavior using high-resolution data streams has become a practical endeavor that can provide benefits in a variety of applications.

Time series data was acquired from the SHRP2 NDS during crash and near-crash events. The goal of the algorithm developed was to identify time series where a crash or near-crash occurred without flagging time series that did not contain any SCEs. Before proceeding, it is necessary to provide definitions relevant to this study.

- *Crash: "Any contact with an object, either moving or fixed, at any speed in which kinetic energy is measurably transferred or dissipated. Includes other vehicles, roadside barriers, objects on or off the roadway, pedestrians, cyclists, or animals."*
- *Near-Crash: "Any circumstance that requires a rapid evasive maneuver by the participant vehicle or any other vehicle, pedestrian, cyclist, or animal, to avoid a crash. A rapid evasive maneuver is defined as steering, braking, accelerating, or any combination of control inputs that approaches the limits of the vehicle capabilities."*
- *Baseline: Any time series without a crash or near-crash.*
- *Safety-Critical Event (will be used synonymously with the term "Event"): Any crash or near-crash event.*

The crash, near-crash and driving definitions were those used by VTTI for their naturalistic driving studies (Guo et al., 2010), since that is the source of the data. The authors defined a safety-critical event as any crash or near-crash, though a case can certainly be made to include other situations and will also be discussed further at a later point.

The proposed algorithm takes the following steps:

- Break time-series into small subsequences or "windows" to examine specific sections in time
- Perform Discrete Fourier Transform to identify the strength of different frequencies present in each window
- Execute K-means clustering to group each window by the strength of different frequencies.

Relevant literature is examined, including additional context for the research motivation as well as some information on previous approaches to this problem. Then a description of the methods used and why they were applied is provided. While the range of applications is diverse, the specific inputs the presented methodology addresses is light vehicle crashes and near-crashes. The discussion section addresses how this algorithm may change based on specifics of each application.

3. Literature review

Discussed in this section will be background information on two key topics relevant to this paper. The first will outline a few studies that collect high-resolution kinematic data on a large-scale. Second, there will be a review of literature that uses this type of data to classify events, or other patterns and behaviors, with a description of the methods being used.

Two studies that have successfully recorded kinematic data during crashes are the 100-Car Naturalistic Driving Study (Dingus et al., 2006) and the larger follow up, SHRP2 Naturalistic Driving Study (Virgina Tech Transportation Institute, 2013). In both of these studies, subjects were recruited to equip their vehicles with cameras, radar, and a Data Acquisition System (DAS) designed by Virginia Tech Transportation Institute (VTTI). High-resolution trip-level data was then generated for subjects over the life of each study. Other studies such as the Safety Pilot Model Deployment in Ann Arbor, Michigan (Harding et al., 2014), the NGSim study in California (Halkias and Colyar, 2006), and Integrated Vehicle-Based System Safety Field Operation Test in Ann Arbor, Michigan (Sayer et al., 2008) Also collected similar data on different scales and in different contexts. Since all of these studies are naturalistic, subjects could, and on occasion did, get into crashes and near-crashes.

In particular, the SHRP2 NDS is unique due to the scale of the study in terms of both network coverage and number of participating subjects, the presence of a system for documenting events, and the presence of a suite of cameras equipped to vehicles for establishing ground truth. While some of the other listed studies also had some of those qualities, they were unable to accomplish all of those at the level of the SHRP2 NDS.

In the SHRP2 NDS, VTTI and the field teams at each site were responsible for identifying when and where their subjects got into crashes. Their approach was to use a collection of criteria to flag potential events in the trip data collected. Those flags include a longitudinal acceleration threshold, a lateral acceleration threshold, some time-to-collision (TTC) thresholds, a yaw rate trigger, and an event button that subjects could press to signal a collision. Individual thresholds alone (e.g. 0.6 g's of longitudinal acceleration) tended to have low recall (true positives/total events) and many of them also had poor precision (true positives/test positive) (Dingus et al., 2006). The SHRP2 Study has adjusted the criteria to flag events by removing most of the radar-based triggers, adding a time-component to the deceleration, adjusting the acceleration thresholds to 0.75 g's of lateral acceleration and 0.65 g's of longitudinal acceleration, and adding some vehicle-safety system activation triggers. The individual triggers often had recall in the single digits, with the best individual flag had around 20% recall. While VTTI was successful in locating crashes despite the low individual identification rates of individual flags, they were able to include some data elements that were not native to the BSM and they have video to verify if an event did or did not occur for trips that were flagged.

Vehicle trajectories from the SHRP2 NDS and similar studies have been analyzed to classify certain occurrences on the road in terms of kinematic data elements. Engström and Victor developed and patented at method using neural networks to classify driving patterns and demonstrated the method on vehicle trajectories in different roadway setting (Engstrom and Victor, 2005). McDonald et. al used a computationally efficient SAX-VOX method to transform time series data into character strings and perform natural language processing to identify commonly observed action and patterns (Mcdonald et al., 2013).

In terms of specifically detecting safety-critical events, Wu and Jovanis proposed a novel algorithm to classify crash types using the maximum differences over time in both lateral and longitudinal accelerations during crashes and near-crashes. They also outlined

the sensitivity and specificities they were able to achieve for a variety of thresholds in those calculated kinematic elements, which were an improvement on the NDS event flags (Wu and Jovanis, 2013a; Wu and Jovanis, 2013b). Kluger and Smith used Euclidean distance to classify crashes with longitudinal acceleration time series data, assigning known patterns to a pre-defined action and flagging any subsequence that did not fit into one of those patterns. This analysis was performed on a limited sample size with promising results (Kluger and Smith, 2014).

A range of additional studies in crash and near-crash dynamics has occurred using TTC or other lead-vehicle-following-vehicle information as a metric. Wu and Thor proposed the idea of a safety frontier calculated by temporal headway and the difference in speeds between the lead and following vehicles. They showed that if the safety frontier was violated, a rear-end crash was likely to occur (Wu and Thor, 2015). Talebpour et al. developed an algorithm that specifically identifies near-crashes in connected vehicle environments using drivers' accelerations and behavior during car-following situations to identify near-crashes and specifically highlights the differences between drivers (Talebpour et al., 2014).

The last type of work in event detection methodologies relates to the concept of traffic conflicts, which has frequently been proposed as a surrogate event for crashes. A traffic conflict is defined as "an observable situation in which two or more road users approach each other in space and time to such an extent that there is risk of collision if their movements remain unchanged" (Amundsen and Hyden, 1977). This traffic conflict technique is used frequently in both simulation and application, however the prevailing concern with this method is that traffic conflicts can often be subjective. Additionally, identifying traffic conflicts in a large network requires an enormous amount of video data reduction (Chin and Quek, 1997). Computer vision techniques that identify vehicles, calculate frame-by-frame trajectories, and if the TTC is below a certain acceptable amount, the event is identified as a conflict. While this is going to consistently call certain types of actions conflicts, it requires a widespread deployment of cameras and software capable of performing this on a large scale in order to capture these (Saunier et al., 2010). Our proposed method does not require additional equipment such as cameras, and is utilizing technology that has already been deployed in many fleets and will continue to be deployed as additional technologies are developed.

4. Data

Data for this study was acquired from the aforementioned SHRP2 Naturalistic Driving Study. The NDS data set contains the same trajectory data elements, collected at the same frequency, 10 Hz. Furthermore, with safety-critical events identified in the data set through analysis by VTTI staff using video data, the NDS provides validated events that are critical to the research.

The event data received consisted of 91 unique incidents, 49 near-crashes and 42 crashes. From here on, the unique incidents will be referred to as "safety-critical events", or sometimes just "events". The events followed a specific, predefined distribution in order to try and encompass most situations that could occur on the road. Table 1 shows the breakdown of the events received by type and speed at the time of event occurrence. The crashes had varying degrees of severity with some being police-reportable and others incurring little to no damage. Obviously, the specific crash type had to be observed within the study, which did limit the sample size for some of the more severe crash types, however the authors are confident that those tend to be the easier crashes to detect as they experience the exceedingly high accelerations relative to what is normally observed. The "other" rows were requested to be filled out with the less commonly seen types in the NDS. For the near-crashes other referenced accelerating and no reaction near-crashes, while for crashes they included animal strikes and any other type of collision not listed in the predefined distribution. It should be noted that in some scenarios the subject driver was hit while in other cases the subject driver struck another vehicle.

Each measured event was a 30-s time series of kinematic data collected by the Data Acquisition System. The time series received consisted of about 300 observations collected around the time of the event. The series was made up of three portions, one pre-event, one during the event, and one after the event.

The additional test data included the same data elements for 35 h of driving during which no known safety-critical events occurred. This data will be referred to as the normal, or baseline, driving set and was used for false positive testing as a way to validate the proposed methodology. Video data was not acquired for either data set but analysts at VTTI watched it and verified there were no safety-critical events before providing it to the authors.

5. The discrete fourier transform

Every time series, $x = \{x_1, x_2 \ldots x_t\}$, can be expressed as a combination of unique circular patterns of varying frequencies, amplitudes, and phases by using a Fourier Transformation. This will result in a function, $X = \{X_1, X_2 \cdots X_f\}$, which is dependent on frequency (f) values instead of time (t). In the case of discrete data, such as time series data sampled at a specific frequency, the Discrete Fourier Transform (DFT) is used to estimate the amplitude, phase, and frequencies. For a time series of length n, the DFT is Equation (1) where $j = \sqrt{-1}$.

$$X_f \stackrel{\text{def}}{=} t \underset{n-1}{\overset{\Sigma}{=}} 0 x_t e^{-j2\pi ft/n} \; for f = 0, 1, 2, \ldots, n-1 \tag{1}$$

The "$\stackrel{\text{def}}{=}$" in Equation (1) stand for "equals, by definition." The time series can also be reconstructed using Equation (2), the inverse DFT.

$$x_t = \frac{1}{N} \sum_{t=0}^{n=1} X_f e^{-j2\pi ft/n} \; for f = 0, 1, 2, \ldots, n-1 \tag{2}$$

By considering Euler's formula in Equation (2), Equation (1) can be represented as a sum of real and imaginary waves of different frequencies, amplitudes, and phases in Equation (3) (Smith, 2007).

$$e^{j\theta} = \cos\ \theta + j\ \sin\ \theta \tag{3}$$

$$X_f \stackrel{\text{def}}{=} \left(\cos\left(-\frac{2\pi ft}{n}\right) + j\ \sin\left(-\frac{2\pi ft}{n}\right)\right) for f = 0, 1, \ldots, n-1 \sum_{t=0}^{n-1} x_t \tag{4}$$

The transform was used to obtain relationship between amplitude and frequency in subsequences of the time series. For reference, the time domain is used to describe the series of accelerations observed over time while the frequency domain will be used to describe the relationship after the DFT has been executed.

The software, R, was used to perform the DFT using built in functions. The functions use an expanded version of the Cooley-Tukey algorithm (Cooley and Tukey, 1965) presented by Singleton (Singleton, 1969) to quickly estimate the amplitude and phase at each frequency.

Agrawal et al. suggested a method to group similar subsequences by some observed characteristics of the DFT of those subsequences (Agrawal et al., 1993). A similar approach was used in the present application, but our interest is in the subsequences that do not match with what is normally expected from drivers. In essence, subsequences are being compared to baseline driving and the ones that do not fit will be flagged as possible events.

Table 1
Distribution of Crash and Near-Crash Events.

Speed		<20 mph	20–29 mph	30–39 mph	40–49 mph	>50 mph
Evasive Maneuver	Braking	3	5	4	3	4
	Steering	4	4	4	4	4
	Other	2	1	3	2	2
Crash Type	Rear End	3	2	1	0	1
	Sideswipe	1	2	2	0	1
	Angle	2	4	2	2	0
	Run-off the Road	1	2	0	0	1
	Curb Strike	1	2	3	2	1
	Other	0	0	4	1	0

6. Methodology

To begin with, the longitudinal acceleration time series were broken into 2.5 s subsequences or "windows". This was done to examine local areas of the time series so the exact region of the time series that was a crash could be identified. Additionally, actions that happen far enough apart are unlikely to be related to the current action. The implications of selecting the specific value of 2.5 s for the window length will be addressed in the discussion section. If a data point was missing in the time series, it was interpolated linearly, as long multiple consecutive observations were not missing. If there were multiple observations missing, the window was removed from the analysis.

Each window of acceleration time series data was treated as a single observation and placed in a data frame. The DFT was then performed on each window, and the amplitudes at each frequency were then recorded and placed in a new data frame. For windows with no event, the transform generally had amplitudes very close to zero for all frequencies past the first ($f = 0$), but transforms on windows with crashes tended to have one or more frequencies with relatively high amplitudes. Fig. 1a shows a selection of windows of longitudinal accelerations in the time domain and Fig. 1b shows the same windows in the frequency domain after undergoing a DFT. The red windows are windows with crashes and the black windows are baseline driving. The transformed windows appear to have a higher amount of separation in the frequency domain compared to the time domain.

The final step in data preparation before classification was to characterize the location of observed peaks. To achieve this, the transform was segmented into P equal sections and the area under the windows in the frequency domain was calculated for each of those sections. The trapezoidal area was used to estimate the area using Equation (5). Note that the first value of the transform X, was omitted because amplitudes observed at frequencies equal to 0 are essentially noise. Additionally the total area under the amplitude for frequencies greater than 0, was recorded for each window.

$$AUC_{p,p+1} = \frac{X_p + X_{p+1}}{2} * \frac{10}{P} for\ p > 0 \quad (5)$$

K-means clustering was then performed in order to group the transformed windows by how similar they are, in the hopes that there would be one or more clusters of only events that emerge and another set of separate clusters with no events would also be present. The areas under each section and the total area under curve were used as variables in the K-means algorithm. The K-means clustering algorithm (Leisch, 2008) is an unsupervised learning technique (i.e. the method is grouping the inputs without knowing their classification), performed in the following manner:

1. Define number of clusters (k) and randomly assign each window (i) to one of the k clusters. Calculate the centroid (c_k) of each cluster.

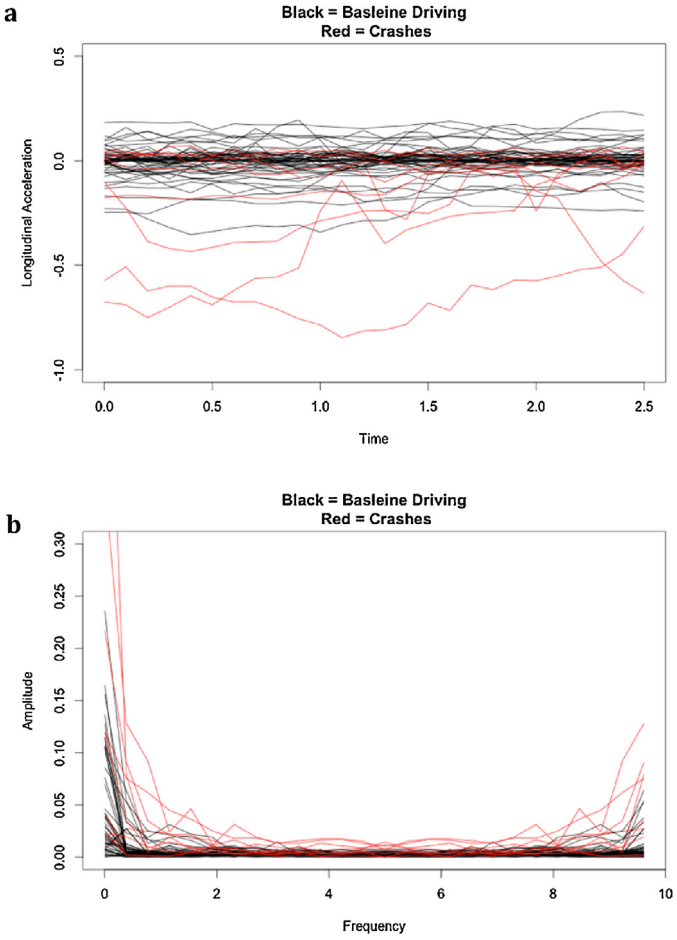

Fig. 1. Time series windows of accelerations during both baseline driving and during crashes. b Windows from 1a converted to the frequency domain with the DFT.

2. Calculate the Euclidean distance (d) between each window and each cluster centroid. Each window is made up of a set of N variables. In this application, the N variables are the areas under each section of the transformed time series in the amplitude-frequency relationship, as calculated by Eq. (5), as well as the total AUC for all sections.

$$d_{i-c} = \sqrt{\sum_{n=1}^{N} (X_n - X_c)^2} \quad (6)$$

- Assign windows to cluster with the minimum calculated Euclidean distance.
- Recalculate center of each cluster to include the newly classified windows.

- Repeat steps 2–5 until the algorithm converges and assignments stop changing.

$$c_{k-n} = \frac{1}{i_{c-k}} \sum_{i} x_{i-n} \tag{7}$$

$$c_k : \left\{ c_{k-1}, c_{k-2}, \ldots, c_{k-N} \right\} \tag{8}$$

What is essentially happening in the clustering algorithm is the normal baseline driving with no events is consistently being classified into one group while anything else that does not fall into this category is being classified into another cluster. In some cases, this method can be sensitive to the arbitrary starting location so the K-means algorithm was rerun multiple times to test how sensitive the results were to the random starting location.

Clusters were trained on a sample of the baseline driving set (50 randomly selected baselines) and all of the event data. The remaining baseline driving consisted of about 35 h and those time series were broken into windows and assigned to clusters without re-centering the clusters. This was done to ascertain if specifically defined cluster centers could be used in an application of this methodology without needing to perform the clustering algorithm in an ad-hoc manner, especially since an iterative process like K-means clustering is unlikely to respond well to scaling at the level envisioned, given current computing capabilities.

The time series was broken up into windows as a way to look at smaller happenings within a trip. Since each window had a constant length, breaks between windows were arbitrary and realistically could, and in some cases did, occur midway through a crash event. A successful algorithm should be able to detect at least one window within the duration of a crash event, but it does not matter if every window during the crash was detected. By flagging a single window, the entire crash can be detected, and it does not matter what the classification says is happening in the surrounding windows. It is entirely plausible that an event spanning multiple windows may only have a single window where the impact occurred classified as an event. While technically the windows during the event that go undetected are labeled false negatives, there is no loss of information by the algorithm's failure to indicate that window was positive if a neighboring window is classified as positive. So the only events that were considered false negatives were the crashes where no window spanning the length of the crash event was assigned to the event clusters. Otherwise, if one or more nearby windows indicated an event, it was considered a success.

Similarly, if consecutive windows indicate an event occurred, that is treated as a single event. So if the algorithm indicates a false positive, if consecutive or nearby windows have all been assigned to one of the event clusters, they are treated as a single false positive. How close two windows need to be in order to be considered the same event is unclear as in all cases of false positives spanning multiple windows, the false positives were next to each other. For the reasons stated, the performance will be evaluated on the percent of crash events detected (the recall), and a false positive rate (number of false positives per hour of driving time).

7. Results

The area under the amplitude-frequency relationship was calculated using Eq. (5) and the following breaks, presented in Table 2. Additionally, total AUC was calculated. Each subsequence had the twelve corresponding observations on Table 2 and the total AUC used in the clustering algorithm.

For the K-means clustering, the "flexclust" (Leisch and Dimitriadou, 2015) package and R version 3.2.4 was used. In K-means, the number of clusters needs to be defined before the algorithm can run. The starting point was varied between 2 and 6 clusters. 4 clusters were selected to represent distinct patterns in the data, as this had the best balance of classification rates in the training set and low false alarm rates in the baseline test set. The resulting cluster centroids are shown in Fig. 2. Fig. 3 shows a neighborhood plot of the cluster centers and the data points assigned to each cluster, projected to the first two principal components. The reason 4 clusters worked best is the decision boundary between clusters 2 and 3 was better than other values for cluster starting points.

Table 2
Breaks for Area Calculations.

Variable	Start Frequency (p)	End Frequency (p + 1)
X1	0.3846154	1.1538462
X2	1.1538462	1.9230769
X3	1.9230769	2.3076923
X4	2.3076923	3.0769231
X5	3.0769231	3.8461538
X6	3.8461538	4.6153846
X7	4.6153846	5.3846154
X8	5.3846154	6.1538462
X9	6.1538462	6.9230769
X10	6.9230769	7.6923077
X11	7.6923077	8.4615385
X12	8.4615385	9.6153846

Table 3 shows a summary of what was assigned to each cluster. The "Subsequences" column shows the total number of subsequences assigned to each cluster. The other four columns show the number of unique events represented in each cluster. This Table helps illustrate the trade-off between classification rates and false alarms. As detection rates increase, the number of false alarms also increase.

Using Table 3, clusters 3 & 4 were selected to be the "event clusters". The events were considered flagged if there was at least one subsequence spanning a portion of the event was placed in one of the two event clusters. An event was considered missed (false negative) if none of the subsequences spanning the event were placed in either of the event clusters. A false positive was any baseline subsequence, or group of consecutive baseline sequences, placed into the event cluster. The additional recall gained by including cluster two does not outweigh the addition of 9 false alarms, especially considering baseline driving is generated at a considerably higher rate than SCEs. This will be discussed further in section 8 Table 4.

The centroids of these clusters were recorded and an additional 35 h of baseline driving was used to get a realistic sense of this algorithm's false alarm rate in application. During the baseline driving,

Table 3
Clustering Classifications.

Cluster	Subsequences	SCEs	Crash	Near Crash	Baselines
1	1049	91	42	49	50
2	202	76	41	41	9
3	90	69	40	29	0
4	7	7	7	0	0
3&4	97	71	31	40	0
2&3&4	299	90	41	49	9

Table 4
Event Cluster Breakdown.

Clusters	Number of Data Sets	Number of Unique Events	Flagged by Algorithm
Event Total	91 Epochs	91	71
Crash	42 Epochs	42	31
Near-Crash	49 Epochs	49	40
Baseline	50 Epochs	0	0
Baseline Test	35 h	0	13

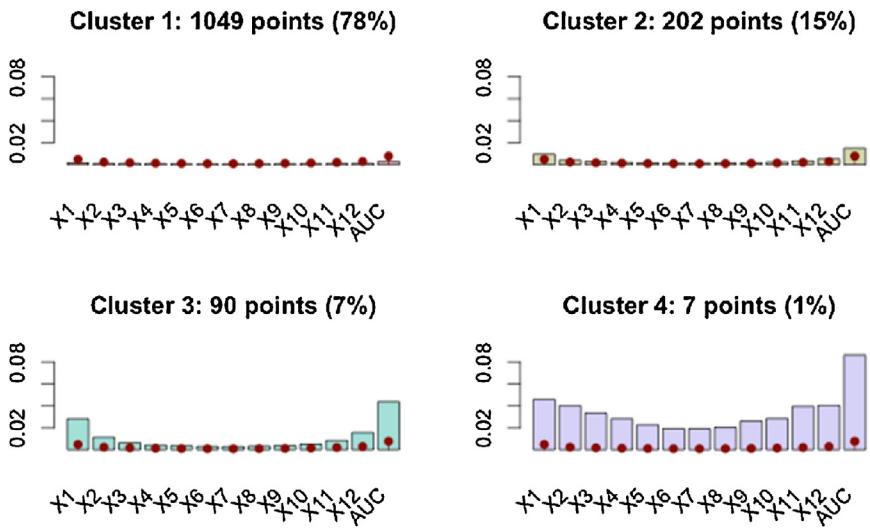

Fig. 2. Cluster Centroids.

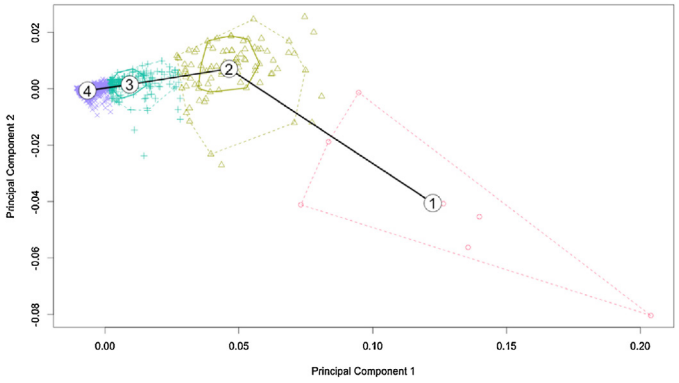

Fig. 3. Neighborhood Plot of K-means Analysis.

a total of 13 false alarms were identified. This translates to about 1 false positive per 2.7 h.

False positives can be explained by a variety of possibilities including:

1. Certain drivers behave more aggressively and the algorithm needs to take that into account when making a prediction.
2. The equipment recording certain drivers' actions may have been calibrated differently, located poorly within the vehicle, or malfunctioning.
3. Different vehicles are prone to different dynamics based on various factors such as the vehicle's weight, tire quality, tire pressure, road conditions, etc.
4. Site characteristics like poor pavement quality could be responsible for the unusual action
5. Clusters may not be using a set of ideal baseline samples, though they were randomly sampled for each run and the results only changed minimally.

There is also a possibility that some of the false positives are actually near-crashes that were not categorized by the NDS study, and since the authors do not have access to video to review what was happening, that will remain unknown. All of the false alarms were assigned to the same event cluster, but the majority of events also were classified in that cluster as well.

The code was run 100 times with different random seeds to determine how sensitive the methodology was to the random starting point. The size of each cluster was the same for all 100 runs showing that the method is not sensitive to random starting point.

8. Discussion

The results using this proposed algorithm to detect safety-critical events were markedly better than anything found in the literature review while using a reasonably complete sample of 91 total crash and near-crash events. Findings indicate that roughly 78% of all safety-critical events can be identified with a high frequency time series of longitudinal acceleration data alone.

As discussed in the introduction, the authors envision a variety of possible applications for a methodology like the one presented. Those applications include:

1. Allowing infrastructure providers to identify SCEs in connected vehicle environments and evaluate road network safety
2. Allowing taxis and shared ride service providers to monitor their drivers and ensure they provide safe rides to customers
3. Allowing insurance companies to monitor their customers' driving tendencies and better evaluate risk
4. Allowing agencies to monitor fleet vehicles (e.g. buses, snow plows, etc.) for both driver performance and tort liability claims
5. Allowing transportation management agencies to monitor traffic and provide emergency response when necessary
6. Providing real-time alerts to emergency response services in connected vehicle environments
7. Identifying events in large-scale naturalistic driving studies

Each potential application will likely have slightly different inputs depending on the purpose of the application, sensor quality, allowable false positive rate, vehicle type, and numerous other factors. This work was done with high quality sensors and for light vehicles driven by the public and is applicable as presented in connected vehicle environments according to current standards as well as large-scale naturalistic driving studies.

There are a variety of ways that longitudinal accelerations can be collected in vehicles, including dedicated sensors like accelerometers or through cellular technology and GPS data. The vehicles used in this study were equipped with high-quality sensors using an accelerometer, but it is unclear how well this carries over using accelerations derived from GPS technology in cell phones. It is possible that the cluster centers may look different and there may be more false positives as errors in GPS readings can lead to larger

accelerations since they are derived from positions and that positional error will carry over.

Allowable false positive rate and definition of an SCE is also going to determine how usable this algorithm is in application and what some of the inputs need to be for it to be successful. While the false positive rate is low, baseline driving gets generated at a much higher rate than SCEs. As a result, there are still likely to be some false positives due to exposure despite the small percentage of baselines that get flagged, although their significance is application dependent. While false positives may be completely unacceptable in some situations, they also likely still indicate an unexpected driving action such as hitting a pot-hole or approaching the end of a queue in congestion, which could be useful to document.

The exact methodology presented in this paper is only applicable for finding SCEs in light vehicles with accelerations collected at a frequency of 10 Hz. The exact inputs, like the 2.5 s window length and number of pre-defined clusters were selected because, of the values tested, the results were the best on a consistent basis when varying other variables.

The general process using a DFT and K-means clustering could work, but likely requires different window lengths, data processing needs, numbers of clusters, resulting cluster centers, and error rates and therefore will need to be altered accordingly. Thus experimentation was performed on the number of predefined clusters and the window length used in order to assess sensitivity. Of the ones testes, the values that worked best in this application with the stated goals in mind were selected. Specifying crash type and severity while considering location is a clear next step to this work, as is examining other kinematic data like speed and lateral acceleration.

9. Conclusion

Presented in this paper was a unique approach to detecting safety-critical events using vehicles' longitudinal acceleration. It used the Discrete Fourier Transform in combination with K-means clustering to flag windows that were likely to be crashes or near-crashes. The algorithm was able to detect almost 78% of crashes and near-crashes that were acquired for this study, while generating about 1 false positive every 2.7 h. This algorithm had excellent performance in comparison to what is currently being used in application, and can also be easily expanded upon as other advances are made and other data types are collected.

Further points of emphasis in future studies on this subject will be to include additional variables frequently collected, such as lateral acceleration to improve the recall and the false positive rate. The most glaring issue with this methodology is the inability to differentiate between crashes and near-crashes as well as the inability to determine crash type for events that were crashes. One additional check to differentiate between crashes and near-crashes could be a heuristic to check if the driver continues to drive. Other than that, including different variables, and placing additional emphasis on the relationship between location of peaks in the frequency domain and the type or severity of event is a logical next step to improve this methodology's performance. Some examples of additional sensors that could provide more information include noise sensors and multi-directional radar. Additional focus on driver-specific or site-specific trajectories could also improve classification rates and reduce false positives.

Funding

This work was funded through the Connected Vehicle/Infrastructure University Transportation Center [DTRT12-G-UTC20]; Federal Highway Administration Office of Operations Research and Development [DTFH6414G00180]; and the FHWA Office of Safety R&D. The research project data set was purchased from Virginia Tech Transportation Institute (VTTI) using the FHWA SHRP2 Safety Implementation program budget.

References

Agrawal, R., Faloutsos, C., Swami, A., 1993. Efficient similarity search in sequence databases. Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO '93), 69–84 (8958546) http://doi.org/10.1007/3-540-57301-1_5.

Amundsen, F., Hyden, C., 1977. Proceeding of first workshop on traffic conflicts. In: First Workshop on Traffic Conflicts, Oslo, Norway.

Chin, H.C., Quek, S.T., 1997. Measurement of traffic conflicts. Saf. Sci. 26 (3), 169–185, http://dx.doi.org/10.1016/S0925-7535(97)00041-6.

Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine computation of the complex fourier series. Math. Comput. 19, 297, http://dx.doi.org/10.2307/2003354.

T. Dingus, S.G. Klauer, V.L. Neale, A.D. Petersen, S.E. Lee, J. Sudweeks, R.R. Knipling, 2006. The 100-Car Naturalistic Driving Study Phase II –Results of the 100-Car Field Experiment.

J. Engstrom, T. Victor, 2005. System and Method for real-time recognition of driving patterns United States.

F. Guo, S.G. Klauer, M.T. McGill, T.A. Dingus, 2010. Evaluating the Relationship Between Near-Crashes and Crashes: Can Near-Crashes Serve as a Surrogate Safety Metric for Crashes? Contract No. DOT HS (Vol. 811).

J. Halkias, J. Colyar, 2006. Next Generation SIMulation Factsheet Retrieved May 18 2016 from www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm.

J., Harding, G., Powel, R., Yoon, J., Fikentscher, C., Doyle, D., Sade, J., Wang, 2014. Vehicle-to-vehicle communications: Readiness of V2 V technology for application (Report No. DOT HS 812 014). Washington.

Kluger, R., Smith, B.L., 2014. Pattern matching longitudinal time series data to identify crashes in naturalistic driving data. In: ITS World Congress, Detroit, Michigan, pp. 3329–3339.

F. Leisch, E. Dimitriadou, 2015. Package flexclust.

F. Leisch A Toolbox for K-Centroids Cluster Analysis (1999) 2008 1–22.

A.D. Mcdonald, J.D. Lee, N.S. Aksan, J.D. Dawson, J. Tippin, M. Rizzo, 2013. The Language of Driving: Advantages and Applications of Symbolic Data Reduction for Analysis of Naturalistic Driving Data Transportation Research Record, 2392 22-30. http://doi.org/10.3141/2392-03.

SAE International, 2009 Surface Vehicle Standard: J2735 Dedicated Short Range Communication Message Set Dictionary.

Saunier, N., Sayed, T., Ismail, K., 2010. Large-scale automated analysis of vehicle interactions and collisions. Trans. Res. Rec.: J. Trans. Res. Board 2147, 42–50, http://dx.doi.org/10.3141/2147-06.

Sayer, J., LeBlanc, D., Bogard, S., Hagan, M., Sardar, H., Buonarosa, M.L., Barnes, M., 2008. Integrated vehicle-based safety systems field operational test plan. Ann. Arbor. (Retrieved from) http://deepblue.lib.umich.edu/bitstream/2027.42/62108/1/102281.

Singleton, R., 1969. An algorithm for computing the mixed radix fast Fourier transform. IEEE Trans. Audio Electroacoust. 17 (2), 93–103, http://dx.doi.org/10.1109/TAU.1969.1162042.

J.O. Smith, 2007. Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second ed. Retrieved from https://ccrma.stanford.edu/~jos/mdft/.

Talebpour, A., Mahmassani, H., Mete, F., Hamdar, S., 2014. Near-crash identification in a connected vehicle environment. Trans. Res. Rec.: J. Trans. Res. Board 2424, 20–28, http://dx.doi.org/10.3141/2424-03.

Virgina Tech Transportation Institute 2013 The SHRP2 Naturalistic Driving Study.

Wu, K.F., Jovanis, P.P., 2013a. Defining and screening crash surrogate events using naturalistic driving data. Accid. Analysis Prev. 61, 10–22, http://dx.doi.org/10.1016/j.aap.2012.10.004.

Wu, K.-F., Jovanis, P.P., 2013b. Screening naturalistic driving study data for safety-critical events. Trans. Res. Rec.: J. Trans. Res. Board 2386, 137–146, http://dx.doi.org/10.3141/2386-16.

Wu, K., Thor, C.P., 2015. Method for using naturalistic driving study data to analyze rear-end crash sequences. Trans. Res. Rec.: J. Trans. Res. Board 2518, http://dx.doi.org/10.3141/2518-04.