Acceleration-based Automated Vehicle Classification on Mobile Bridges using Computer Vision Techniques

Chul Min Yeum, Shirley J. Dyke, Christian Silva, and Ricardo E. Basora Rovira

chulminy@gmail.com
sdyke@purdue.edu
silva15@purdue.edu
rembasora@gmail.com

Report IISL – 009, December 2015

Intelligent Infrastructure Systems Laboratory

Abstract

Mobile bridges have been widely used for a broad range of applications including military or disaster restoration. Because the bridge is rapidly deployed under a variety of boundary and environmental conditions, and has irregular usage patterns, a detailed record of usage history is crucial for ensuring structural safety. To address this issues, a new acceleration based vehicle classification technique is proposed to automatically identify the class of each vehicle. The proposed technique is based on the premise that each class of vehicles produces distinctive dynamic patterns while crossing this mobile bridge, and those patterns can be extracted from the acceleration responses. Measured acceleration signals are converted to time-frequency images to extract visual patterns. Object recognition techniques that originate from computer vision methods are repurposed to uniquely extract and classify those patterns. The effectiveness of the proposed technique is successfully demonstrated using a laboratory-scale bridge by simulating various real scenarios.

Keywords: Mobile bridge, Vehicle classification, Computer vision, Structural monitoring

Acknowledgement: The authors would like to thank Jeff Demo at Luna Innovation Inc. for providing invaluable comments for this study. The authors acknowledge support from Small Business Innovative Research (SBIR) Program and the Engineering Research and Development Center - Construction Engineering Research Laboratory (ERDC-CERL) under Contract No. W9132T-12-C-0020. The authors would also like to acknowledge fellowship support through the Purdue Military Research Initiative.

TABLE OF CONTENTS

1. Introduction

Mobile bridges are an essential structure to facilitate short-term mobility when faced with natural or man-made obstacles. When a mobile bridge is deployed, it is necessary to rapidly confirm that it is safe to cross before use or entry. Typically, such evaluation is based on the usage history, and thus a simple and automated approach to record actual usage is needed. Specifically, it would be best to identify each vehicle that crosses the bridge and determine the class of vehicle as the actual usage is monitored. Historically, a passive sensor known as a Remaining Service Life Indicator (RSLI) has been used (Department of the Army, 2006). It consists of four metallic "filaments" that are designed to fracture progressively as the bridge is subjected to loading corresponding to a specific number of cycles due to vehicle crossings. However, this approach yields a conservative and imprecise evaluation of a bridge's condition because the sensor is designed to only indicate when the bridge has been subjected to an allowable number of vehicle crossings. The design of the sensor is based on a single maximum load and fixed installation setup. Moreover, this approach was not intended to have the fine granularity needed to identify the number of vehicles in a particular class or weight that cross the mobile bridge. To address this need, Luna Innovation Inc. (Luna) and Purdue University have been collaborating to develop an approach to monitor actual usage of such a mobile bridge. An algorithm has been developed that has the ability to identify and record the class of each vehicle traversing a given bridge. Acceleration measurements can be readily implemented on such a bridge, and are the basis of the approach.

Vehicle classification techniques on the permanent bridges or roads have broadly researched last decades (Wall et al., 2009; Ma et al., 2014; Cestel, 2015; Cardinal, 2015; TDC, 2015). However, conventional vehicle classification methods are not suitable to mobile bridges for some reasons. First, any modification of the bridge itself for sensor installation is prohibited. Limited space is allocated for placing sensors, and even a small intrusion to bridge surfaces would compromise the integrity of the bridge itself. Second, due to the broad range of operating conditions under which the bridge is used as well as practical aspects associated with implementation in the field, manual in-situ calibration is not possible. Third, the typical sensors used in such systems are not suitable for long-term use in extremely harsh environmental conditions. For example, conventional strain gauges are subject to challenges regarding adhesion of the sensors and compensation for temperature drift (Espion and Halleux, 2000).

The characteristics of mobile bridges do lend themselves well to the vehicle classification problem. First, the dimensions and properties of each physical bridge of a given model that are manufactured are quite standardized. Thus the specifications and dynamic characteristics of the bridge can be assumed to be consistent from bridge to bridge. Second,

vehicles are classified into general load classes, and the classification system to be used may be defined in advance. For instance, the military load classification (MLC) system may be used up to 40 MLC, which is a standard vehicle classification system used by the Army based on hypothetical vehicles. Any vehicle around the world can be classified according to its MLC value (Department of the Army, 2008). Third, although the vehicles can traverse the bridge in either direction, the actual speed variation of vehicles is relatively small. The speed limit over such bridges is low (under 40 km/h) and stopping and accelerating are not permitted during driving, which would produce significant impact loading on the bridge (Department of the Army, 2006). This restriction indicates that similar and consistent speed patterns will be observed for the vehicles. Finally, the significant dynamic vehicle-structure interactions due to the similar mass and modal characteristics of the vehicle and the bridge, suggest that there will be similarities in the acceleration signals which can be extracted and exploited for vehicle classification. These specific advantages highlight the potential for implementation of a pattern recognition based vehicle classification.

Herein we propose a technique to monitor the bridge usage patterns based on acceleration measurements. First, acceleration signals recorded when each vehicle traverses the bridge are converted into time-frequency images through suitable post-processing. Then, unique patterns contained within those images are extracted for classification. An image based object classification algorithm is implemented to extract those patterns and train robust classifiers for the vehicles. Training of the classifiers is performed in advance under a variety of vehicles and bridge setups. Once classifiers are constructed for each vehicle, the user must simply apply those to the acceleration measurements for classification when a vehicle crosses the bridge. There is no need for manual in-situ calibration or additional sensor measurements.

Herein we develop and experimentally validate an automated vehicle classification method for mobile bridges based on acceleration measurements. We demonstrate this approach in two experiments by categorizing vehicles into multiple classes according to the vehicle type. We are not focused on measuring the dynamic load imparted by individual vehicles. The proposed approach has several advantages over existing vehicle classification techniques, which are commonly used in typical bridges and roads. It enables fully automated vehicle classification without manual calibration, and uses inexpensive acceleration sensors that are easy to install and maintain without modification to the bridge. Furthermore, the approach is developed with field applications in mind, and relies on the fact that training can easily be performed in advance with a variety of vehicles, speeds and bridge setups based on the needs of the bridge owner. Thus, classifiers trained in advance can be applied to all bridges of that design/model. Experiments are conducted to realistically consider and evaluate the potential for this technique to be applied to a real mobile bridge,

and to systematically examine the capabilities of the technique. High success rates are achieved.

2. Vehicle Classification Approach

Figure 1. Overview of the proposed technique

The fundamental assumption behind the proposed technique is that each class of vehicles crossing a mobile bridge produces unique and distinguishable dynamic (acceleration) patterns. Such patterns would be algorithmically observable for classification. These patterns are preserved even with reasonable variations in the vehicle's speed or mass.

Conceptually, this problem is analogous to visual object recognition. In facial recognition, a human observer can intuitively detect the "differences" among various faces by automatically integrating several specific features as well as overall patterns even under different angles or lighting conditions. Modern object classification algorithms in the computer vision field have sought to mimic the human's "difference" detection capability and have made tremendous gains. Our proposed technique applies those powerful algorithms to tackle this vehicle classification problem for vehicles traversing a mobile bridge.

There are a number of challenges to overcome for real world implementation of vehicle classification on a mobile bridge. Because in each new installation the bridge will be emplaced in a new location with an entirely new setup, the dynamics of the bridge itself may

vary greatly. Boundary conditions, environmental conditions and even the length of the bridge will vary, and thus the dynamic patterns produced by a particular vehicle traversing the bridge may not be consistent across this range of setups. In other words, the various dynamic patterns associated with different lengths or boundary conditions will be uncorrelated or have much weaker correlation than the patterns associated with similar bridge setups. Further, it is unreasonable to collect training data for every possible boundary condition and length. To address this challenge, our implementation of the technique first calls for the identification of the closest bridge setup by driving a known vehicle, called a reference vehicle, across the bridge just after bridge installation. Because the training data collected from a variety of bridge setups already includes data obtained with the reference vehicle, we first identify the closest match to data obtained from a new bridge setup to be used to automatically classify a vehicle using vehicle classifier corresponding the bridge setup. This approach is demonstrated in the experiment discussed in Section 3. The reference vehicle can be, for example, a truck responsible for transporting the bridge, although any vehicle can be used for this purpose.

The overall procedure of the proposed technique is shown in Fig. 1. In the training stage, a variety of bridge setups are replicated by altering the bridge length, boundary materials or boundary elevation to consider the range of setups under which the bridge is expected to be deployed. In each bridge setup (or configuration), acceleration data are collected across the general MLC classes of vehicles including a selected reference vehicle. However, the user may freely assign the classes for the vehicle types, and use of the MLC classes is not required. A bridge setup classifier, denoted BSC in the figure, is trained using only data associated with a reference vehicle. The vehicle classifiers are also trained using data collected from each class of vehicle under the corresponding bridge setup, denoted VCB in the figure. Recall that this process is implemented one time and then the results are applicable to all bridges of that model that are manufactured. At the time of deployment, a reference vehicle first traverses the bridge several times and acceleration responses are collected. The proposed system will automatically apply the bridge setup classifier to those data to identify the bridge setup that is closest to the current one. Once that similar bridge setup is detected, the corresponding vehicle classifier is simply applied when new data is acquired from vehicles needing classification.

Before introducing the proposed technique, some assumptions are made, which are quite reasonable for such a mobile/temporary bridge: (1) one vehicle is crossing the mobile bridge at a time, (2) a given bridge setup does not vary greatly during usage, (3) vehicles traverse the bridge with a reasonably constant speed, and (4) vehicles within a given class of vehicles all produce similar dynamic patterns. The foundation for assumption (4) is that the axle load and the number of wheels and wheelbases, which are used for constructing the standardized MLC index, are the dominant factors in producing the dynamic patterns to a

mobile bridge. The steps required to implement the proposed technique are discussed subsequently.

2.1 Data acquisition

Step 1 is to acquire vertical acceleration signals from sensor nodes when the vehicle is traversing a mobile bridge. All sensors are triggered simultaneously when a vehicle enters the bridge. There are a number of techniques available to detect true event signals and differentiate them from noise, such as a STA/LTA (short term average/long term average) ratio method or threshold triggering (Jakka et al., 2015). The basic idea behind such methods is to capture sudden amplitude changes (increase/decrease) or fluctuations. These large changes/fluctuations do not typically occur in noise. In general, a vehicle entry event produces large accelerations similar to an impact load and the change can easily be distinguished. Thus, such methods are effective and applicable for our purpose.

Step 2 is to estimate the time at which the vehicle exits the bridge. The acceleration record to be used for classification should only contain data acquired when a vehicle is traversing the bridge. In general, the triggering algorithms used in *step 1* can be applied in reverse to stop (exit triggering) recording data in the sensors. However, if the sudden exit of a vehicle acts as a large impact (releasing) force on the bridge and produce transient vibration, it would be difficult to estimate the time a vehicle exits using a simple amplitude-based threshold triggering method. Fig. 2(a) presents a typical signal acquired in *step 1* collected from the lab-scale experiment in Section 3. Unlike the sudden jump at the entrance of the vehicle, the response gently decreases at the exit.

To address this issue, we use the fact that the transient vibration is determined by the bridge's dynamic characteristics, which are mainly composed of low frequencies. A high-pass filter is implemented to filter out the transient vibration of the bridge and retain the high-frequency components induced by vehicle crossing and noise. This process enhances the transition from a valid signal (the vehicle is on the bridge) to noise (the vehicle has exited the bridge) by reducing the free vibration of the bridge due to an impact load resulting from a vehicle exit. With this filter in place, the triggering methods used in *step 1* are applied to the filtered signals in reverse order to identify the exit time. In this study, a threshold triggering method is used to identify the exit time, and its threshold is estimated based on the background noise floor, which is measured with ambient conditions. In the threshold computation, the noise signals are assumed to follow Gaussian distribution with a zero mean and standard deviation of σ, the threshold limits are set to $\pm \alpha\sigma$, where α is a scaling factor and typically set to above 3.

Fig. 2(b) represents the signal in Fig. 2(a) after a high-pass filter has been applied. The boundary between the valid signal and noise is much clearer than in the record in Fig. 2(a). The red dotted line denotes the estimated exit time using this reversed threshold triggering method. Note that the filtered signal is only used for determination of the exit time and cropping the original signal.

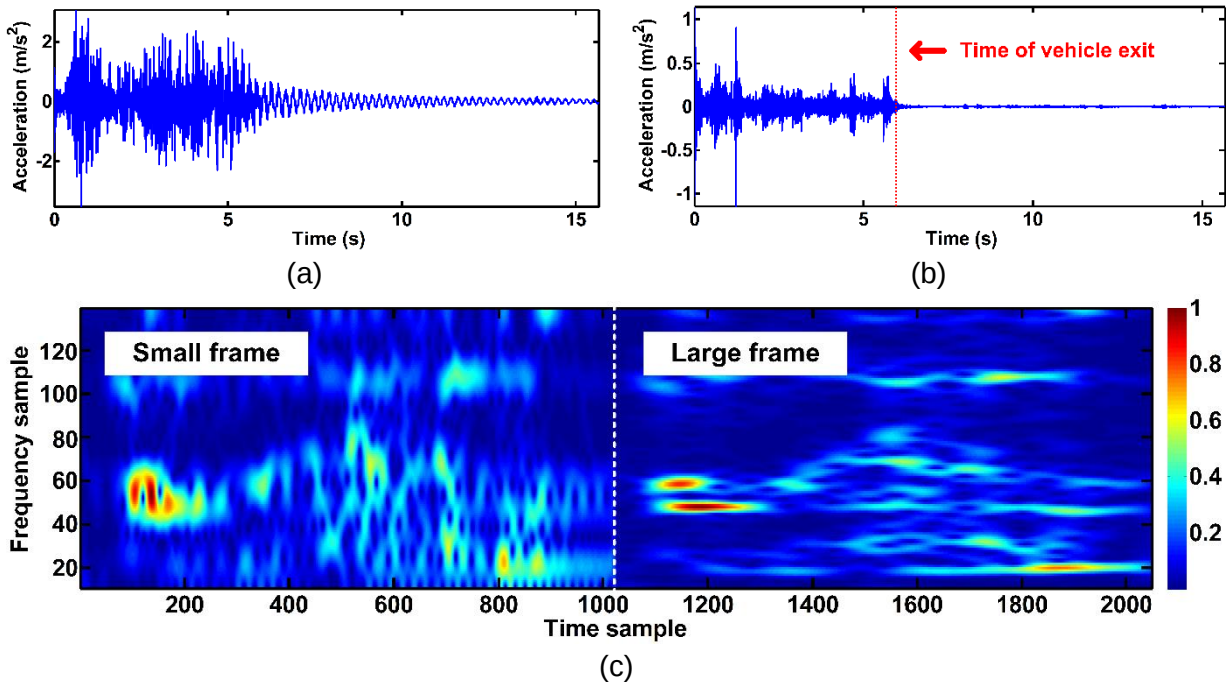


Figure 2. Transformation of an acceleration signal to an image: (a) raw signal in *step 1*, (b) applying a high-pass filter for estimating vehicle exit time in *step 2*, and (c) spectrogram image in *step 3*.

2.2 Image transformation

Using the prior steps, valid acceleration signals, which are restricted to the time when a vehicle is crossing the bridge, are obtained by cropping the measured raw signals from the entrance time to the exit time. In this subsection we address how this time signal is transformed into an image to become the subject image for object classification. A spectrogram, which is a visual representation of a time-frequency signal, is selected for image transformation because this enables extracting features for classification from both time and frequency at the same time. A wavelet transform may also be used for this process. This spectrogram is viewed as a simple 2D grayscale image. However, further manipulation

of this transformation is necessary to extract consistent and robust features for classification from images.

The image transformation and post-processing techniques proposed in this study can be understood within the context of general image-based object classification. First, images should be normalized with respect to their size and scales. Here, size and scale correspond to the dimension (resolution) and object scales of the images, respectively. Such normalization can preserve positional information of unique features in object appearances. For example, in facial recognition, the location of the eyes on a face should be similar pixel locations across all training and testing images. Similarly, spectrogram images transformed from acceleration time signals should be of identical size and scale to facilitate their use for vehicle classification. Thus, along the frequency axis, the number and resolution of the spectral lines (i.e., frequency increments) should be identical in all images (i.e., the same bandwidth is used). Along the time axis, the number of time steps must be identical although the time resolution may be different due to the variation in the speed of each vehicle. In this sense, if a vehicle is assumed to be moving with a constant speed, the time axis in the spectrogram images may be interpreted as the vehicle location on the bridge, rather than the actual time.

Second, the most suitable image resolution must be determined. In computer vision, general pattern and object classification do not require high-resolution images (Dalal and Triggs, 2005; Torralba et al., 2007; Viola and Jones, 2001). The performance of these methods typically converges above a certain resolution. An intuitive example is that objects are still recognizable even in thumbnail images. The choice of resolution can be interpreted as the desired frequency bandwidth of the measured acceleration record because the bandwidth determines the resolution in both time and frequency. Unless the informative dynamic pattern spreads into the higher frequency range, dense time and frequency resolution are not required. Furthermore, high frequency signals are often susceptible to noise and may not be suitable for detection of robust features. Aside from performance concerns, processing high-resolution images is also computationally expensive and time-consuming. Thus, reductions in the sampling rates and image sizes are beneficial for optimizing computational time/resources and power consumption for computation and data transmission.

The third consideration is variance normalization. For object classification, normalization of pixel values in images is a necessary process to minimize the effects of lighting variations. Normalization is interpreted as putting more weight on relative difference of pixel values rather than individual absolute ones when features are extracted. However, such a normalization process is not applicable for our purpose. Intuitively, large and heavy vehicles produce larger amplitudes of acceleration, which may become a strong feature to use for discriminating between the classes of vehicles. Furthermore, preserving amplitude

features is important to provide strong correlations between vehicles in same class. In this study, signals are processed without variance normalization before converting them to a spectrogram image. Depending on the learning (training) algorithm, the normalization process may be required such as with the K-nearest neighbor algorithm (KNN), which computes the distance between feature vectors (Bishop, 2006). The boosting learning algorithm used in this study, introduced in next subsection, constructs classifiers by summing weak classifiers generated from individual features and thus does not require any such normalization process (Friedman et al., 2000; Torralba et al., 2007).

Step 3 is to transform the valid acceleration signals obtained using the entrance and exit times into spectrogram images normalized in size and scale. As discussed previously, all images have the same number of points in the *x* (time) and *y* (frequency) axis, and a same spectral lines and frequency resolution. Suppose that all original signals are measured with the same sampling frequency using the sensors installed on the bridge. The sampling frequency should be determined based on the frequency bandwidth of interest. For simplicity, the number of time points in the spectrogram images is set to be higher than the maximum expected number of time points. This approach is used because down sampling will result in aliasing. Applying a low-pass filter in advance to prevent aliasing is not applicable in this case because a low-pass filter changes the frequency content in the signal and thus changes the spectrogram, making it difficult to compare the features in the spectrogram for classification. Note that the time resolution of the spectrogram image is also varied depending on the overlap of the FFT frame. In this study, the FFT frame moves forward one time point at a time, and it produces same number of time points with time signals.

Spectrograms are computed from original signals with the same number of points in the FFT and two different sizes of frames. The frame lengths of the short-time Fourier transform for spectrogram conversion vary by resolution in both time and frequency. Good resolution cannot be achieved in both time and frequency with only one frame size. Thus, in this study, the spectrogram image is constructed with two spectrogram transformations using two different sizes of frames, and extracting features from appropriate time and frequency resolutions, respectively. Features used for classification will be extracted from both images, as discussed in the next subsection.

The spectrograms generated from all original signals have the same frequency values along the *y* axis, but are not still consistent along the time axis due to variations in the speed. To correct this inconsistency, all spectrogram images are resized to have the same number of points along the *x* (time) axis. Fig. 2(c) presents spectrogram images generated with the signal in Fig. 1(a) after extracting the valid acceleration record. Each portion of the combined image has either better time or frequency resolution depending on frame size used for spectrogram. Each portion of the image has been normalized here for visual clarity in this explanation.

2.3 Training bridge setup and vehicle classifiers

Step 4 is to extract features from the processed spectrogram images obtained in the previous the step. Extracted features are used for distinguishing between images generated from different vehicles. A number of feature extraction techniques have been introduced in the literature for general or specific object detection and classification (O'Byrne et al., 2013; Torralba et al., 2007; Yeum and Dyke, 2015). In this study, Haar-like features are implemented due to their domain-independency and computation speed (Viola and Jones, 2001). Features are computed by summing across a local rectangular region using Haar-like wavelets. For simplicity, 1, 2, 3 and 4 rectangular Haar-like feature windows are used, which was originally proposed by Viola and Jones (Viola and Jones, 2001). However, any feature extraction technique that can represent the unique patterns contained in such images can be applied for this purpose.

Step 5 is to learn the vehicle classifiers using the extracted features. A set of robust binary classifiers of each vehicle are designed to determine whether or not the features in a test image point to the existence of a corresponding vehicle in the training data. In this study, a boosting algorithm is implemented to generate a robust classifier. Boosting is a way of combining many weak classifiers to produce a strong classifier. By updating the different weights of weak classifiers adaptively, depending on misclassification errors, the optimum strong classifier is obtained, thus minimizing misclassification errors. There are several boosting algorithms introduced in the literature, but in this study, the gentle boost algorithm, proposed by Friedman, is used because it is known as simple to implement, numerically robust and experimentally proven for object detection (Friedman et al., 2000; Torralba et al., 2007). Similar to *step* 4, in this step, feature learning (training) techniques, which can produce robust classifiers from extracted features, can be implemented for this purpose.

For multi-class classification problems, there are two general strategies: One against Rest (OvR) and One against One (OvO) (Bishop, 2006). OvR generates a single classifier per class trained from samples of the corresponding class as positive, and all other samples as negative. On the other hand, OvO produces a multiple classifier per class trained from samples of corresponding class as positive, and samples of other individual classes as negative. For the K-class classification problem, OvR and OvO produce the number of K and K(K-1)/2 binary classifiers, respectively. There is an open question for selecting the strategy, which depends on data and applications. In this study, we adopt OvO for our application because it heuristically provides better classification outcomes. Hereafter, these binary classifiers are called as candidate classifiers, all of which are used for making a single final classification.

Use of multiple sensor measurements provides two means for classifier designs. One way is to train measurements from all sensors together based on the assumption of

strong correlation between them. The other way is to train measurements of individual sensor installed at a certain location based on the assumption that they have stronger correlation than one from all sensors. If each vehicle produces specific features at a specific sensor location, the latter way provides better classification results. However, the number of data available for training is reduced as one over the number of sensors, and it fails to extract general features unless there is enough data. In this study, we adopt the former, which considers all sensor data at once for training.

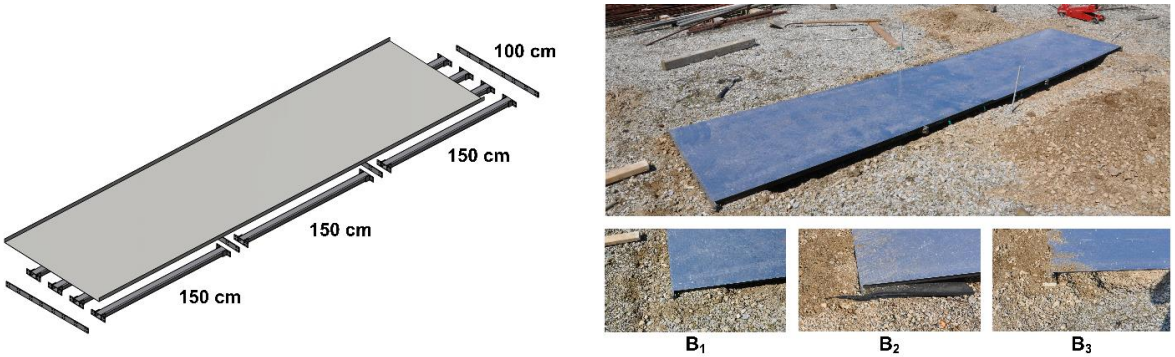
Aside from vehicle classifiers, the bridge setup classifier must be trained, as mentioned at the beginning of Section 2. This bridge setup classifier detects the closest bridge setup, which points to the data collected under conditions similar to the current one. Only data from the reference vehicle is used to construct this classifier. Features from the reference vehicle data in *step* 5 are still used, but class labels assigned in *step* 6 are for bridge setups, not vehicles.

2.4 Vehicle classification

Once the bridge is emplaced at a new site, the current bridge setup is first identified using the bridge setup classifier, which is trained in advance. The reference vehicle crosses over the bridge to acquire acceleration data from multiple sensors. Features are computed by executing *steps* 1 to 6. Then, the bridge setup classifier is applied to the extracted features to identify the closest bridge setup to the current one. Vehicle classifiers trained from the detected bridge setup are applied to new measurements for future vehicle classification.

3. Experimental Validation

3.1 Experimental setup



(a) (b)

Figure 3. Description of a lab-scale test bridge: (a) drawing of the bridge and (b) emplacement of the bridge with three different bridge setups

To validate the technique, an experiment is conducted using a lab-scale bridge, shown in Fig. 3. This lab-scale experiment is intended to validate the following aspects of the technique: (1) distinguishable dynamic patterns (features) are generated and are contained within the acceleration signals; (2) the proposed technique can successfully classify class of vehicles by extracting such patterns; (3) reasonable variations in vehicle speed or the crossing direction do not affect the classification outcomes; and (4) the proposed technique is successful even when the bridge setup under which testing data are to be collected is not specifically used in the training stage.

This lab-scale bridge is a scaled down version that has similar dynamics to those of the real mobile bridge. It is designed to be portable, easy to mount and dismount, and can be installed using various boundary conditions and span distances. The bridge itself in Fig. 3(a) is constructed using four hollow squared section beams oriented lengthwise, constructed as three 150cm long segments interconnected through connection plates. The 450 cm by 100 cm bridge surface is a thin aluminum sheet with 0.1 cm thickness covering the entire area spanned by the end beams in the lateral direction; and by the boundaries, in the longitudinal direction. The bridge is emplaced across a pit in the soil with enough clearance under the bridge to eliminate any chance of contact with the ground.

To evaluate the performance of the technique, three different bridge setups are used, including: gravel (baseline ground surface condition in this location), rubber pads, and wooden supports, denoted herein as B_1, B_2, and B_3, respectively, as shown in Fig. 3(b). A slight ramp at each end of the bridge is created using extra soil to provide a relatively smooth entrance and exit. The three different bridge setups are collected by varying boundary materials, although all tests use the same bridge length and elevation.

A total of eight accelerometers (PCB model 333B40), four placed along each side of the bridge, are installed to measure high-quality vertical vibration responses (PCB Piezotronics, 2015). An m+p VibPilot data acquisition system is used with 24-bit A/D converters, simultaneous sample and hold, and built-in anti-aliasing filters linked to the sampling rate (m+p International, 2015). A sampling frequency of 1024 Hz is employed to acquire high frequency signals, and also to enable an investigation of the impacts of signal processing. After data acquisition, a low-pass Butterworth filter with a 60 Hz cutoff frequency is applied to the original acceleration records, as the data is down-sampled to 120 Hz. These processed signals are meant to represent measurements from typical low-cost sensors that would be used a real implementation as well as to reduce the number of time points of the spectrogram image. To identify the relevant portion of the data associated with the time that

the vehicle is traversing the bridge, the data acquisition is started manually before each vehicle enters the bridge. The velocity of the vehicle is not known (as it is not needed by the technique), and thus a threshold for estimating exit time based on the amplitude of the response is computed from a baseline noise signal that is then scaled by a factor of 5 (α).

Six vehicles associated with five different vehicle classes (K), denoted as V_1 to V_6, shown in Fig. 4, are driven across the bridge. V_1 and V_2 are of the same type (class) of a vehicle and data from V_2 is only used for testing, and thus is not included in the training data. Each vehicle crosses the bridge six times total, including three times from right to left and three times from left to right. All vehicles are pulled manually by three different individuals. Each vehicle begins from rest at a position outside of the bridge. Crossing durations are varied (i.e., speed is varied) from approximately 3 to 6 sec depending on the vehicle classes or pulling force. Two two-wheeled vehicles (bicycle) are included to consider different numbers of wheels and types of tires.

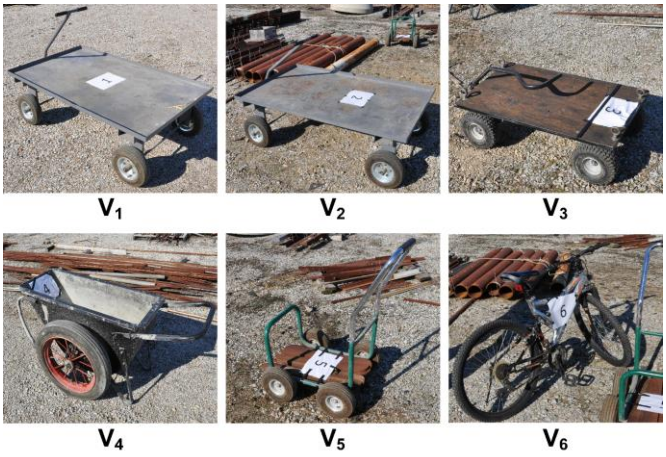


Figure 4. Six vehicles with five different types (Note that V_1 and V_2 are same type of a vehicle)

Hanning windows with 32 and 128 frame sizes are used for computing the spectrograms. The number of FFT in each frame and time samples are consistently set to 256 and 1024, which are then the height (*y*) and width (*x*) of the spectrogram, respectively. The resolution of spectrogram images are selected to be larger than the frame size and time points. Twenty thousand features are generated in each spectrogram image. A “strong” classifier is trained using 100 selected “weak” classifiers, which is sufficient to ensure convergence of classification (Torralba, 2007).

3.2 Vehicle classification results

Initially the spectrogram images are examined visually to qualitatively assess the existence of distinguishable dynamic patterns available for such classification. Fig. 5 shows spectrogram images for each of the vehicles in a select number of runs associated with B_1. Clearly, images from different runs of the same vehicle have quite similar visual patterns regarding the location and intensity of features, and images from different vehicles are often visually discernable. Also, the images from V_1 and V_2 are very similar each other because they are the same vehicle class. This demonstrates the existence of unique and visually distinguishable dynamic patterns associated with each vehicle crossing, and indicates there is potential for interpretation of this problem within the domain of image classification problem.

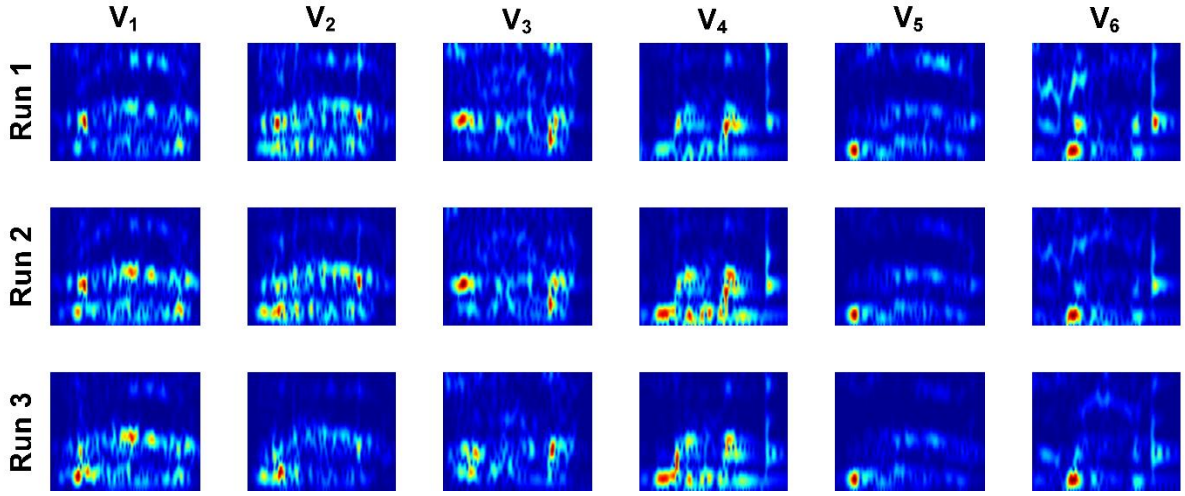


Figure 5. Spectrogram images generated from three different crossings of six vehicles under B_1. For visual clarity, these images show only the left portion of the corresponding original spectrogram images, and are normalized.

As explained previously, it is anticipated that in the real world implementation the bridge installation would be immediately followed by a brief series of baseline crossings with a reference vehicle to identify the bridge setup based on the existing training data, and thus increase the likelihood of successfully classifying subsequent vehicles. For this experiment, V_4 is arbitrarily assigned as the reference vehicle.

All data of V_4 is partitioned into training and testing data for cross-validation. A round is defined as when data from a single run (one record of data from a vehicle crossing) is selected for testing and the remainder are used for training. Thus, a total of 18 rounds are produced from six runs of a vehicle and three different bridge setups. For example, if a single

run from B_1 is assigned for testing, the data from the other five runs from B_1 plus those from the six runs from B_2 and B_3 are used for training of each bridge setup (herein, class). A single run contains eight sensor measurements. Three candidate classifiers learned from the training data are applied to each measurement. The three candidate classifiers here are binary classifiers to differentiate between B_1 and B_2, B_1 and B_3, and B_2 and B_3. This process produces 24 classification outcomes. Among these 24 outcomes, the outcome obtained in the majority of the classified classes is selected as the final one if the number of occurrences exceeds a particular threshold. If not, the outcome of the corresponding round is assigned as unclassified (UC). The maximum number of true classification will be 16 among 24 because candidate classifiers related with a specific class are two among three and they are applied to eight measurements. Here, the threshold is thus set to 12, which is three quarters of the maximum number of true classifications. The unclassified class is intended to reduce the chance of false classification due to erroneous estimation of the exit time or unusual speed variations, such as breaking or accelerating. It potentially would also capture situations in which vehicles are not the source of the measured vibrations, perhaps including human, winds, or debris strikes, which do not generate the specific dynamic patterns that vehicles produce.

The classification results are provided in Table 1, in the form of a confusion matrix. The confusion matrix is a common way to represent the performance of classification algorithms. Each entry in the matrix indicates the total number of instances the runs in the actual class are classified in the corresponding predicted class, respectively. Here, in our initial classification using V_4 as the reference vehicle, we aim to classify only the V_4 data into the three classes of bridge setups. All predicted classes are correctly matched with actual ones yielding 100% classification.

Table 1. Confusion matrix of bridge setup classification using V_4 data

Actual \ Predicted	B_1	B_2	B_3	UC	Accuracy
B_1	6	0	0	0	100 % (6/6)
B_2	0	6	0	0	100 % (6/6)
B_3	0	0	6	0	100 % (6/6)

Next, we validate the algorithm's ability to perform vehicle classification by classifying data records from all five classes of vehicles. Similar to the previous example, in each round, a single run data is selected for testing and the remainder are used for training. Thus, a total of 108 rounds are generated, which is the product of six runs, six vehicles and three bridge setups. It is assumed that all bridge setups are correctly identified in advance. This means, in

each round, vehicle classifiers trained with data from the identical bridge setup are applied to a single run. For example, if a single run data from B_1 is assigned for testing, vehicle classifiers, which are trained using the other 5 runs from B_1, are applied to this data for classification. In each round, 80 classification outcomes are produced and the UC threshold is set to 24, which is three quarters of the maximum number of true classification, in this case 32.

Table 2 shows the resulting vehicle classifications. The results are quite accurate with nearly all classes being correctly classified, and only four of the 108 rounds being incorrectly classified. Recall that V_2 is the same vehicle as V_1, and is not included in the training, and these data records are only used for testing. Thus, the true predicted class of V_2 should be V_1, as obtained in the results. There is no predicted class of V_2.

The results of this evaluation show that (1) unique dynamic patterns of vehicle crossings are successfully extracted and used for classification; (2) the proposed technique is successful even under different bridge setups; and (3) reasonable actual speed variations across all classes of vehicles do not typically affect classification results.

Table 2. Confusion matrix of vehicle classification (Use of training and testing data collected from an identical bridge setup)

Actual \ Predicted	V_1	V_3	V_4	V_5	V_6	UC	Accuracy
V_1	18	0	0	0	0	0	100 % (18/18)
V_3	0	16	0	1	0	1	88.9 % (16/18)
V_4	0	0	17	1	0	0	94.4 % (17/18)
V_5	0	0	0	16	0	2	88.9 % (16/18)
V_6	0	0	0	1	17	0	94.4 % (17/18)
V_2*	18	0	0	0	0	0	100 % (18/18)

*V_2 and V_1 are the same class of the vehicle

Next, we impose a realistic challenge by implementing the classification technique using testing data that is collected from different bridge setup from any bridge setup included in the training data. In reality, although a range of conditions must be used to develop a suitable set of training data, the bridge setup in the field will not be identical to one of those used for training. Therefore, the prior results are not entirely sufficient to fully demonstrate the effectiveness of the technique. To address this situation, we simulate these real world challenges using our laboratory scale data. Here data from the same bridge setup are not used for testing and training at the same time. For example, if a single run is extracted from B_1 in a round, vehicle classifiers created only from B_2 or B_3 training data are used for

classification. This mimics the situation in which the current bridge setup to be used for testing is not included in the training data used to develop the classifiers.

First, the bridge setup classifier using V_4 (our reference vehicle) is applied to data from other two bridge setups to decide which bridge setup is closer to the testing setup, even if they are not identical. The bridge setup classification results show that the bridge setup of B_1, B_2 and B_3 are the closest to B_3, B_1, B_1, respectively. Then, vehicle classifiers of corresponding bridge setup are applied to the testing data. For example, if the single run selected for testing is from B_1, vehicle classifiers trained from B_3 are used for vehicle classification.

The number of rounds in this evaluation is identical to that in the previous study. The only difference is that the bridge setup and vehicle classifiers are trained using different subsets of data. Table 3 provides the classification results in this study, yielding an accurate classification of most vehicles, and less accurate results for two of the vehicles. But the success in classification is always greater than 66.7% except for V_3. With different bridge setup data the accuracy is reduced, but is still much greater than a random guess, which would only yield 20 % (1/K).

The relatively high accuracy obtained in these results clearly demonstrates that the same vehicle produces similar and unique dynamic patterns, even under different bridge setups. This indicates that there is great potential to apply this technique in field mobile bridge applications. Furthermore, for a given bridge the number of bridge setups needed for training of the technique may be reduced, or even optimized, if the range of conditions can be defined. Thus, the owner may need to only replicate a few different bridge setups corresponding to the most common situations and conditions in which the bridge is likely to be emplaced. It is not necessary to simulate all such situations, resulting in a saving in time and money.

Table 3. Confusion matrix of vehicle classification (training and testing data collected from different bridge setups)

Actual \ Predicted	V_1	V_3	V_4	V_5	V_6	UC	Accuracy
V_1	18	0	0	0	0	0	100 % (18/18)
V_3	6	8	0	2	0	2	44.4 % (8/18)
V_4	0	0	14	3	0	1	77.8 % (14/18)
V_5	2	0	3	12	0	1	66.7 % (12/18)
V_6	0	0	2	4	12	0	66.7 % (12/18)
V_2	18	0	0	0	0	0	100 % (18/18)

4. Conclusion

An automated, acceleration-based vehicle classification technique is developed to specifically address the complex set of operating conditions that mobile bridge structures are subjected to in the field. The technique exploits the availability of computationally efficient image-based object classification methods, and considers a novel application of these methods to spectrograms of the vibratory responses of mobile bridges. A laboratory-scale tests are conducted to demonstrate the effectiveness of the technique by evaluating many different scenarios. Several combinations of realistic variations in the bridge setup and vehicle speed are considered in each of these cases. The technique classifies the subject vehicles with high success rates.

5. References

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Cardinal. (2015). www.wimscales.com (date last viewed 08/08/15).

Cestel. (2015). www.cestel.eu (date last viewed 08/08/15).

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005, IEEE Computer Society Conference on (Vol. 1, pp. 886-893).

Department of the Army. (2006). Operator's manual for rapidly emplaced bridge (REB), TM 5-5420-280-10.

Department of the Army. (2008). Field Manual 3-34.170 Engineer Reconnaissance, Appendix E.

Espion, B. & Halleux, P. (2000). Long-term measurements of strains with strain gauges and stability of strain gauge transducers, Reports in Applied Measurement, 3, 1–11.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2), 337-407.

Jakka, R. S. & Siddharth, G. (2015). Suitable triggering algorithms for detecting strong ground motions using MEMS accelerometers. Earthquake Engineering and Engineering Vibration 14.1: 27-35.

m+p International. (2015). www.mpihome.com (date last viewed 08/08/15).

Ma, W., Xing, D., McKee, A., Bajwa, R., Flores, C., Fuller, B., & Varaiya, P. (2014). A wireless accelerometer-based automatic vehicle classification prototype system. Intelligent Transportation Systems, IEEE Transactions on, 15(1), 104-111.

O'Byrne, M., Schoefs, F., Ghosh, B. & Pakrashi, V. (2013) Texture analysis based damage detection of ageing infrastructural elements, Computer-Aided Civil and Infrastructure Engineering, 28(3), 162–177.

PCB Piezotronics. (2015). www.pcb.com (date last viewed 08/08/15).

TDC system (2015). www.tdcsystems.co.uk (date last viewed 08/08/15).

Torralba, A., Kevin P. M., & William T. F. (2007). Sharing visual features for multiclass and multiview object detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.5: 854-869.

Wall, C. J., Christenson, R. E., McDonnell, A. M. H., & Jamalipour, A. (2009). A non-intrusive bridge weigh-in-motion system for a single span steel girder bridge using only strain measurements (No. CT-2251-3-09-5).

Viola, Paul, & Michael Jones. (2001). Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE.

Yeum, C. M., & Dyke, S. J. (2015). Vision-Based Automated Crack Detection for Bridge Inspection. Computer-Aided Civil and Infrastructure Engineering.

6. Appendix

6.1 Appendix A: Pseudocode

Here is a brief pseudocode of the proposed method. We simulated our algorithm using MATLAB interface and several embedded or user-developed algorithms were integrated.

Nomenclature

n	: number of sensor locations (size: scalar)
nCT_i ($i = 1..n$)	: number of time pts in cropped measured acceleration (size: scalar)
nRT	: number of resampled time points (size: scalar)
A_i	: measured acceleration (size: $1 \times nT_i$) ($i = 1..n$)
AC_i	: cropped measured acceleration (size: $1 \times nCT_i$) ($i = 1..n$)
AR_i	: resampled cropped measured acceleration (size: $1 \times nRT$) ($i = 1..n$)
$AccThresh$	: acceleration threshold for cropping (scalar)
$nfft$	: FFT length
$imgSp_i$	: spectrogram of AR_i (size: $nfft \times nRT$) ($i = 1..n$)
$integralSp_i$	: integral image of $imgSp_i$ (size: $nfft \times nRT$) ($i = 1..n$)
$windowS$	: (Hamming) window of length for spectrogram (size: scalar)
$noverlapS$	: number of samples that each window overlaps for spectrogram (size: scalar)
$windowR$	: (rectangular) window of length for moving RMS (size: scalar)
$noverlapR$	: number of samples that each window overlaps for moving RMS (size: scalar)

1. Acceleration measurement at n locations

2. Signal cropping using moving RMS

$$Tmp_i = rms(A_i, windowR, noverlapR, 0) > AccThresh \quad for\ all\ i$$

$$AC_i = A_i(Tmp_i(1) : Tmp_i(end))$$

Source code: http://www.mathworks.com/matlabcentral/fileexchange/11871-signal-rms

3. Resampling of all cropped signals

$$AR_i = resample(AC_i, nRT, nCT_i, 0) \quad for\ all\ i$$

Source code: http://www.mathworks.com/help/signal/ref/resample.html

4. Spectrogram computations

$$imgSp_i = abs(spectrogram(AR_i, windowS, noverlapS, nfft)) \quad for\ all\ i$$

Source code: http://www.mathworks.com/help/signal/ref/spectrogram.html

5. Integral image computations

$$integralSp_i = integralImage(imgSp_i) \quad for\ all\ i$$

Source code: http://www.mathworks.com/help/vision/ref/integralimage.html

6. Feature value computation

7. Applying a trained classifier

6.2 Appendix C: Resources

The videos of actual experiments were recorded for documentation. Videos were uploaded at YouTube and can be accessed using the following links:

Vehicle 1: https://youtu.be/Kf9QXElma1k
Vehicle 2: https://youtu.be/S6O7MvFTpDA
Vehicle 3: https://youtu.be/lUI9LdsONZE
Vehicle 4: https://youtu.be/dD-6NVGZojU
Vehicle 5: https://youtu.be/1j6A6xBtELs
Vehicle 6: https://youtu.be/W7okgz-kdCE