Road Anomaly Detection Using Smartphone: A Brief Analysis

Van Khang Nguyen[1,2](✉), Éric Renault[1], and Viet Hai Ha[2]

[1] Institut Mines-Télécom/Télécom SudParis, CNRS UMR 5157 SAMOVAR, 9 rue Charles Fourier, 91011 Evry Cedex, France
{van_khang.nguyen,eric.renault}@telecom-sudparis.eu

[2] College of Education, Hue University, Hue, Vietnam
haviethai@gmail.com

Abstract. Identification of road anomaly not only helps drivers to reduce the risk, but also support for road maintenance. Arguably, with the popularity of smartphones including multiple sensors, many road anomaly detection systems using mobile phones have been proposed. This paper aims at analyzing a number of typical road anomaly detection methods in terms of resource requirements, energy consumption, fitness conditions. From these measurements, we suggest some improvement directions to build road anomaly detection algorithms appropriate for smartphones.

Keywords: Road anomaly · Pothole · Road condition
Sensors network

1 Introduction

Today, road systems are dense all over the world. Unfortunately, under the influence of vehicles and weather, roads gradually deteriorate, to get poorer and poorer and lead to the appearance of anomalies. This causes huge economic losses and endangers people in traffic. Determining the quality of roads and finding potholes to repair is a very challenging and much needed effort. As a result, many systems for road monitoring and identification of potholes have been developed or proposed.

The pothole detection is based on a variety of information like audio, image, video, 3D laser or vibrations [11]. Sensors used to identify potholes can be either dedicated hardware mounted on vehicles or smartphone sensors [4]. Recently, smartphones have become very popular with low-cost Internet, and many researches on pothole detection based on smartphones' accelerometer and GPS sensors have been conducted.

Anomaly detection can be processed on phones or in a center or both. If the processing is done in a datacenter, a large amount of data might be exchanged the smartphone and the datacenter, while if the processing is done in the smartphone, the detection program should be designed so that it does not consume too

© Springer Nature Switzerland AG 2019
É. Renault et al. (Eds.): MSPN 2018, LNCS 11005, pp. 86–97, 2019.
https://doi.org/10.1007/978-3-030-03101-5_8

much resources like CPU, memory and energy. The purpose of this article is to analyse and compare current road anomaly detection methods that aim at being executed on the phone in real time. The main focus is set on vibration-based solutions.

The rest of the paper is organized as follows: Sects. 2 and 3 describe different pioneering researches on pothole detection based on acceleration and GPS data. These methods are divided into two classes: *threshold based* and *classification based*. In Sect. 4, each method described in Sects. 2 and 3 is analyzed in terms of (a) memory requirements and computing resources, (b) compatibility with qualitative conditions like road quality, vehicle speed and type of vehicle, and (c) applicability under unspecified telephone conditions. Improvements to some issues are also discussed in this section. Finally, Sect. 5 concludes the paper.

2 Threshold-Based Methods

The common feature of threshold-based methods is the use of simple algorithms to determine potholes according to the principle that if a component or a derivative of the components of a measured acceleration exceeds a specific threshold: a road abnormally is detected (see Fig. 1).

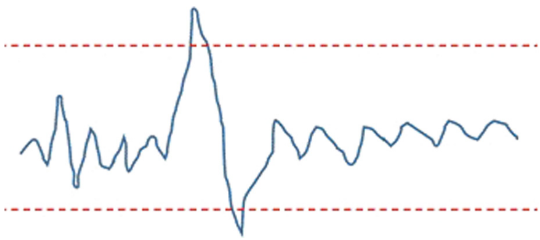

Fig. 1. Ideal of threshold-based methods.

The Pothole Patrol System [8]**.** Eriksson et al. proposed a system called Pothole Patrol (P^2). In order to detect road anomalies, they use accelerometers and GPS sensors installed in taxis. Gathered information from the sensors are:

$$< \texttt{time, location, speed, heading, 3-axis acceleration} >$$

The detection is made by a vehicle embedding a computer. The results (potholes) are sent to a control center through a WiFi network. The server compute clusters of potholes to prevent false positive analysis. The pothole detection algorithm proposed in P^2 consists of five filters. Information received from the sensors are first segmented into windows of 256 samples. Each window then is tested by the filters to reject non-pothole event types (see Fig. 2). These filters are:

- **Speed:** this filter rejects windows in which the speed is too small.
- **High-pass:** this stage rejects low-frequency components from acceleration measurement in x-axes and z-axes. This filter removes events like acceleration, turning, veering and braking.
- **z-peak:** this important filter rejects all windows with a z-axis of acceleration less than threshold t_z.
- **xz-ratio:** this step rejects events that affect both sides of a car such as railway crossings, speed bumps and expansion joints. A window is rejected if the x-acceleration peak in a sub-window from the read z-acceleration peak, is less than some factor t_2 times the z-acceleration peak. Let $\triangle w = 32$ samples be the size of the subwindow.
- **speed vs. z-ratio:** this step removes windows in which the z-acceleration peak is less than factor t_s times the speed.

In order to detect parameters t_z, t_x and t_s, authors use sample data to train the pothole detector.

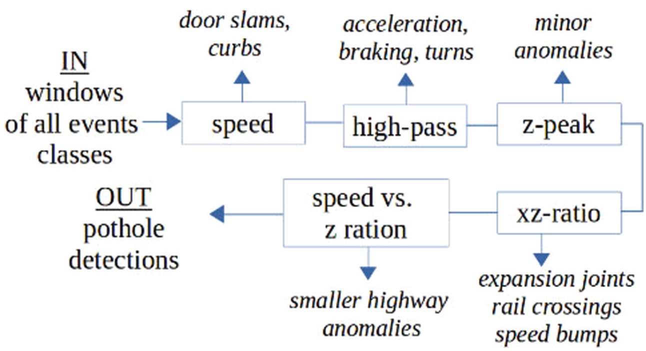

Fig. 2. P^2's pothole detector.

Nericell Road System Monitoring [13]. Mohan et al. present a system to monitor road conditions. Detection is performed on a smartphone. The system uses accelerometer, microphone, GSM radio and GPS sensors to detect potholes, bumps, brakings and honkings. Authors also propose to use Euler's angles to represent the orientation of the accelerometer and a method to estimate the angles based on gravity, and braking actions or accelerations.

Algorithms Used to Determine Anomalies:

- **Braking Detection:** Let a_x be the acceleration on the x-axis. The algorithm consists in computing $\overline{a_x}$ over a sliding window. When $\overline{a_x}$ is greater than threshold T, a braking event is detected. The windows size used by the authors is 4 s.

- **Bump and pothole detection:** Nericell does not distinguish between bump types such as potholes and speed bumps. At high speed, (i.e $\geq$25 km/h), authors use the z-peak method [8]: a bump is detected when a_z is greater than a given threshold. At low speed, authors propose a filter called z-sus. With z-sus, a bump is detected when there is a sustained dip, e.g a_z is less than threshold T for at least 20 ms.

Mednis et al. [12] proposed four algorithms for real-time road anomaly detection on smartphone sensing system. All of these algorithms are based on 3-axis acceleration data:

- **Z-THRESH:** is similar to the z-peak algorithm used in [8] and [13]. Events are detected when the amplitude of the z value of the acceleration exceeds a specified threshold.
- **Z-DIFF:** events are detected when the difference between two consecutive values is greater than a specific threshold.
- **Z-STDDEV:** the standard deviation on a small sliding window is computed. When the standard deviation is greater than a specific threshold, an event is detected.
- **G-ZERO:** an event is detected when the value of all three axes is less than a specific threshold.

Vittorio et al. [18] applied a anomaly detection method on mobile devices based on both acceleration signal and GPS signal but using only the vertical acceleration. Authors proposed to apply a partial ri-orientation to recompute the vertical value of acceleration (a_z). Since GPS data's frequency was 1 Hz and acceleration data's frequency was at least 5 Hz, authors grouped accelerometer data in groups of 1 s by computing a_{z_min}, a_{z_max} and a_{z_avg}. The detection is based on the vertical acceleration impulse (DVA) defined by DVA $= a_{z\_max} - a_{z\_min}$.

Authors have proposed a background noise removal:

$$DVA = \begin{cases} 0 & \text{if } DVA \leq DVA^{st} \\ DVA - DVA^{st} & \text{if } DVA > DVA^{st} \end{cases}$$

where DVA^{st} is the maximum DVA in a stationary condition. In order to filter anomalies, DVA is filtered as following:

$$DVA = \begin{cases} 0 & \text{if } DVA \leq DVA^{th} \\ DVA & \text{if } DVA > DVA^{th} \end{cases}$$

where DVA^{th} is a reference set of DVA values. Experiments allow to determine the optimal DVA^{th} value.

3 Classification-Based Methods

For classification-based methods, firstly features are extracted from the data collected by various methods. Then a classification will be applied to these features to classify them to detect road abnormally and types of abnormally (see Fig. 3).

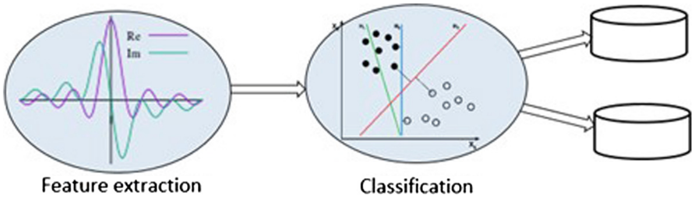

Fig. 3. Ideal of classification-based methods.

Perttunen et al. [15] proposed a road anomaly detection method which consists of many complex stages. Data (GPS and acceleration signals) first pass a preprocessing in which GPS outlier rejection and Kalman filters are applied to reduce noise. Then, the signals are framed using a sliding window to extract some features. Many of them are extracted from the acceleration signal: mean, variance, standard deviation, peak-to-peak, signal magnitude area, 3-axis autoregressive coefficients, tilt angles, root mean square for x-axis, y-axis and z-axis, correlation of signals between three dimensions. A Fast Fourier Transformation is also applied to incorporate information from specific frequencies. Authors also apply a removing linear dependence method [17] to remove dependence speed from the features. Finally, they apply support vector machines [6] to classify the windows. The result of this classification allows them to identify the road anomalies.

Cong et al. [5] proposed to apply Wavelet Packet Decomposition (WPD) and One-Class Support Vector Machine (SVM) on acceleration data to detect road anomaly. WPD is used for feature extraction. Acceleration signal is segmented into 114 sample windows. Then, a WPD using Daubecheis wavelet [7] is applied to extract features. The second stage is feature selection. Authors tested four selection methods: forward selection [2], backward selection [2], genetic algorithms [10] and Principal component analysis (PCA) [2]. At the end, a one-class SVM is used to classify the features to detect road anomalies.

Seraj et al. [16] proposed the Road pavement Anomaly Detection System (RoSDS), a smartphone-based system. Sensor data are first preprocessed to remove speed dependency by an empirical decomposition mode [9]. Like other classification-based methods, RoSDS extracts features form data. The window size is 256 samples with a 170-sample overlap for data sampled at 93 Hz or 128/85 samples with a data frequency of 43 Hz. The extracted features are:

- Time-domain: mean, variance, standard deviation, root mean square, peak to peak, zero crossing rate, mean of absolute value, correlation between the axis, tilt angles, etc.
- Frequency-domain by means of a Fast Fourier transform.
- Stationary Wavelet Transform [14].

Finally, support vector machines are applied to classify the features. The classification consists of two steps: the first step identifies the anomalous windows and the second step classifies the type of anomaly. The training is performed using labelled data consisting of 3066 windows including 2073 normal windows and 993 anomalous windows to detect the parameters.

Wolverine method [3] introduced by Bhoraskar et al. uses smartphone sensors to monitor road condition and detect bumps. Authors have proposed a reorientation using magnetometer beside accelerometer and GPS sensors. The reading of acceleration on axes of the vehicle could be recalculated by rotation matrices based on Gravity vector, Magnetic vector and vehicle motion direction. A bump detection method and a braking detection method based on an SVM classifier [1] has been proposed in Wolverine. Reoriented accelerometer data are devised into 1 s duration windows (i.e 50 samples). Several features are computed from these windows: means and standard deviations in three axes (μ_x, μ_y, μ_z, σ_x, σ_y and σ_z) which are used to detect bumps and three difference values (δ_x, δ_y and δ_z) which are used for the braking detection. Features δ_x, δ_y and δ_z are defined as follows:

$$\begin{aligned}\delta_x &= \max a_x - \min a_x \qquad \forall a_x \in \text{window}\\ \delta_y &= \max a_y - \min a_y \qquad \forall a_y \in \text{window}\\ \delta_z &= \max a_z - \min a_z \qquad \forall a_z \in \text{window}\end{aligned}$$

The features are then classified by SVM to predict the vehicle state. In order to create labelled data for SVM training, authors used the K-means clustering algorithm to classify the features into two classes and manually labelled smooth or bumpy (for bump detection) and "brake" or not (for braking detection).

4 Comparison of Road Anomaly Detection Methods

This section presents an analysis of some of the methods developed above for the purpose of examining the relevance of such methods when installed on a smartphone. The results are not compared as each research was examined on separate data. The comparison is made in terms of resource requirements, ability to adapt to different road conditions, vehicle type and speed. Table 1 qualitatively summarizes the results of the evaluation.

4.1 Memory and Computational Resources

The first part of this section is dedicated to the resource requirements as this determines the power consumption of the smartphone. However, note that most of the detection methods need a training. The training needs to be realised one time and it can be performed on computers. Thus, there is no need to mention the resources needed for the training or for the experiments. As a result, only the detection after the training, that can be executed on a smartphone, is considered.

Threshold-Based Methods. All these algorithms require the least memory and computational resources. As the algorithms use direct data gathered from sensors, the current window is the only one that needs to be stored in memory. In addition, the complexity of these threshold-based algorithms is quite small.

- The P^2 *system*. Assume n is the number of acceleration samples captured on the entire road that need to be monitored. The signal is segmented into windows of size k. Each window passes through five filters. For filters *speed*, *High-pass*, *z peak* and *speed vs. z ratio*, the detector has to loop through windows one time only. However, for filter *xz-ratio*, the program first searches for the peak on the z-axis of acceleration and then, for each peak of the z-acceleration, it searches for a peak on the x axis of the acceleration in $\triangle w$. Since the size of $\triangle w$ is composed of only 32 samples, the complexity of the z-acceleration peak filter is $O(k)$. As a result, the complexity of the algorithm becomes $O(n)$.
- The *Nericell system* uses a sliding window. The time complexity of the algorithm is $n \times k$ where n is the total number of acceleration samples and k is the window size. As a result, the complexity is $O(k \times n)$.
- For the *bump detector*, both z-peak and z-sus cases may loop through samples one time. Thus, the complexity of bump detector is $O(n)$. When two detectors are applied, the total complexity becomes $O(n \times k)$. Authors chose a window size of 4 s, i.e. $k = 4 \times f$, where f is the frequency of the accelerometer.
- Mednis et al. also proposed four different algorithms. One can easily domnstrate that the complexity for three first ones, namely *Z-THRESH*, *Z-DIFF* and *G-ZERO* is $O(n)$. For *STDEV(Z)*, the complexity is $O(n * k)$ where k is the window size, since the standard deviation must be computed on a sliding window. However, the complexity of *STDEV(Z)* may be reduced to $O(n)$ if the standard deviation can be progressively computed.
- The algorithm proposed by Vittorio et al. [18] is the same as *Z-THRESH* ans its complexity is $O(n)$.

Classification Methods. In contrast to the previous algorithms, these methods require much more resources. Moreover, all the following algorithms share a common aspect: before being proceed, the data from the sensors are first divided into windows. Then, a feature extraction method such as a *Fast Fourier Transformation* (FFT) or a *Wavelet Packet Decomposition* (WPD) is applied. After the feature selection, a classification using SVM (for Support Vector Machine) is done.

- Perttunen et al. [15] have used several tools like Kalman filters, FFTs, linear dependence methods and/or SVM with many extracted features.
- In Cong et al. [5], WPD is used to extract the features, then one selection feature method is applied, and finally a SVM is executed.
- FFT, Stationary Wavelet Transform and SVM are also used in RoSDS [16].
- Wolverine [3] is the least computationally intensive method in the set of classification-based methods. Nine features are computed from the windows and are classified using a SVM.

In short, these methods require more memory and computation. Therefore, if these methods would be deployed on a smartphone, they would consume a lot of energy. In addition, the detection program should run in the background. If the program takes up too much resources, they can be prioritized and/or cleaned by the operating system.

Processing time is also a factor to be considered. Typically, a condition for which an algorithm can execute in real time without losing data is that the processing time of a data block is smaller than the time between two block arrivals. Moreover, the anomaly detection program should not take up a large portion of the smartphone's processor.

4.2 Adaptation to Real Conditions

One of the main challenges for road-anomaly detection algorithms based on vibrations is that aptitude and vibration frequency both depend on road conditions, speed, vehicle type and suspension system quality, etc.

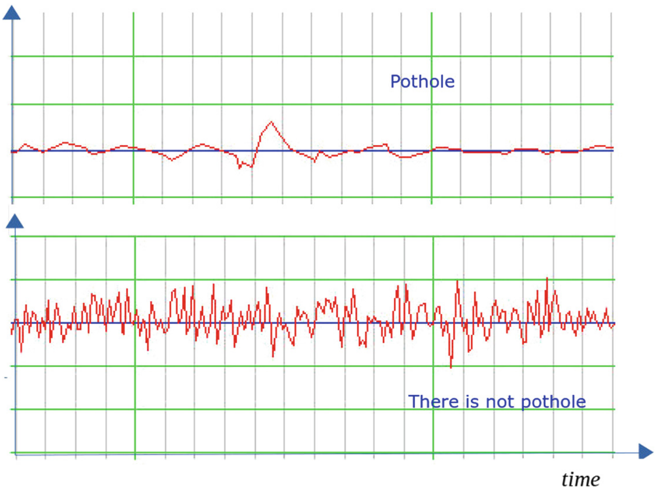

Fig. 4. Two different examples of road conditions

Current Threshold-Based Algorithms are Less Adaptable to Real Conditions. They use fixed thresholds and compare sensed data to peak values or the standard deviation of vertical values of acceleration data to find out road anomalies. Although these thresholds are determined from experiments or machine learning, they cannot be applied to all conditions where the magnitude and the standard deviation of the acceleration depend on the velocity, the road type and/or the vehicle type. As shown on Fig. 4, the amplitude of the acceleration (vertical component from the accelerometer) as the vehicle enters the pothole is smaller than in the other cases when the car does not pass through the pothole, but is driven on the worse roads. In both cases, cars run at about 7 m/s. As a result, a dynamic threshold, that would adapt to the quality of the road, would be far more appropriate for these algorithms. A dynamic threshold would be computed/adapted regularly to suit the current road conditions.

Classification-Based Methods Promise Better Adaptability. Window classifications based on many attributes allow a better identification of potholes under different conditions of speed, quality, etc.

However, to be able to classify accurately, training data should be a combination of different types of roads and vehicles at different speeds. In fact, building a complete labelled data set is practically difficult. It might be interesting to consider unsupervised learning approaches to improve the training. In addition, the training should be done in a data center and the parameters later uploaded on the smartphone to save energy.

4.3 Adaptation to Random Orientation

During operation, the smartphone may not be fixed on a support in the car or on the motorbike. As a result, it might be oriented randomly or, even worse, the orientation may change all the time while sensing. Thus, all three components of the acceleration data must be computed in terms of the coordinate axes attached to the vehicle (see Fig. 5). This adjustment called reorientation is also an important challenge today.

Fig. 5. The coordinate system of the car and of the phone do not coincide.

Tools have been used to determine the orientation of a smartphone, like Euler angles [13,18] or rotation matrices [3]. The general idea for determining the direction of a smartphone is based on two directions: the first one consists in considering the vertical direction that can be determined from gravity, and the second one is the motion direction. In [13], the motion direction is determined when there is a braking on the straight line as the deceleration is necessarily in the direction of the rotation. Another way to determine the motion direction consists in using the magnetic sensors: the combination of the magnetic vector and the vehicle motion direction (based for example on the GPS) allows to determine the angle of the smartphone deviation from the vehicle [3].

In fact, determining the orientation of the smartphone may face some difficulties. For example, the use of the vehicle braking to determine the vehicle direction from the smartphone may not be straightforward. If the smartphone is rotated but the vehicle does not change the speed at the same time or if it does not move in a straight line, one cannot determine the vehicle direction from the acceleration data. If the motion direction is determined using a magnetic sensor, the main difficulty is that many smartphones do not include such a magnetic sensor... Moreover, it is also important to bear in mind that a magnetic sensor may produce inaccurate results when located close to iron, which is often the case in a vehicle.

4.4 Removal of User Actions

At last, another very important element always omitted in the literature is the exclusion of the actions taken by the smartphone's user. Mohan et al. [13] are able to detect user interactions by looking for pressed keys, mouse events and/or phone calls. However, the user can move the smartphone without typing any key and/or touching the screen. In order to overcome this issue, it seems necessary to take into account the variations of the acceleration vector to identify the unusual motions of the device.

Table 1. Summary of methods based on features that are suitable for smartphones.

Method	Complexity	Memory[1]	Conditions[2]	Orientation[3]
Threshold-based methods				
Pothole Patrol [8]	Low	Low	Low	Low
Nericell [13]	Low	Low	Low	Good
Algorithm Z-THRESH Z-DIFF, STDEV(Z) [12]	Low	Low	Low	Good
Algorithm G-ZERO[12]	Low	Low	Low	Medium
Method of Vittorio et al. [18]	Low	Low	Low	Good
Classification-based methods				
Perttunen et al. [15]	High	High	Good	Medium
Method of Cong et al. [5]	High	Medium	Medium	Medium
RoSDS [16]	High	Medium	Medium	Medium
Wolverine [3]	Medium	Medium	Medium	Good

[1]Use of the memory—[2]Adaptation to real conditions—[3]Adaptation to random orientation

5 Conclusion and Future Work

Connecting smartphones to take advantage of their sensors to monitor road condition promises to be widely applied in the near future. This field of research has been active for few years and several researches have been conducted in this way. However, no method has proven enough accuracy to be used in widespread applications. This article explored the different pothole detection methods that have been released so far and presented both advantages and drawbacks. We showed that threshold-based methods require the least resources but are less adapted to effective road conditions. As a conclusion, we suggest using dynamic thresholds for these methods. For the classification-based methods, data required for the training are usually quite large. In the future, it shall be considered to train these algorithms at a data center constantly and then send the parameters back to the smartphones.

In the future, we will concentrate on determining the best way to divide the data processing between smartphones and datacenters to reduce communications and limit energy consumption.

References

1. Support vector machine. https://en.wikipedia.org/wiki/Support_vector_machine
2. Alpaydin, E.: Introduction to Machine Learning. Adaptive Computation and Machine Learning, 2nd edn. MIT Press, New York (2010)
3. Bhoraskar, R., Vankadhara, N., Raman, B., Kulkarni, P.: Wolverine: traffic and road condition estimation using smartphone sensors. In: 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–6. IEEE (2012)
4. Chugh, G., Bansal, D., Sofat, S.: Road condition detection using smartphone sensors: a survey. Int. J. Electron. Electr. Eng. **7**(6), 595–602 (2014)
5. Cong, F., et al.: Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection. In: Guo, C., Hou, Z.-G., Zeng, Z. (eds.) ISNN 2013. LNCS, vol. 7951, pp. 291–299. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39065-4_36
6. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
7. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia (1992)
8. Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H.: The pothole patrol: using a mobile sensor network for road surface monitoring. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, pp. 29–39. ACM (2008)
9. Feldman, M.: Signal Demodulation. Wiley, New York (2011)
10. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence, pp. 439–444. University of Michigan Press, Ann Arbor (1975)
11. Kim, T., Ryu, S.K.: Review and analysis of pothole detection methods. J. Emerg. Trends Comput. Inf. Sci. **5**(8), 603–608 (2014)
12. Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., Selavo, L.: Real time pothole detection using android smartphones with accelerometers. In: 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), pp. 1–6. IEEE (2011)
13. Mohan, P., Padmanabhan, V.N., Ramjee, R.: Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, pp. 323–336. ACM (2008)

14. Nason, G.P., Silverman, B.W.: The stationary wavelet transform and some statistical applications. In: Antoniadis, A., Oppenheim, G. (eds.) Wavelets and Statistics. LNS, vol. 103, pp. 281–299. Springer, New York (1995). https://doi.org/10.1007/978-1-4612-2544-7_17
15. Perttunen, M., et al.: Distributed road surface condition monitoring using mobile phones. In: Hsu, C.-H., Yang, L.T., Ma, J., Zhu, C. (eds.) UIC 2011. LNCS, vol. 6905, pp. 64–78. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23641-9_8
16. Seraj, F., van der Zwaag, B.J., Dilo, A., Luarasi, T., Havinga, P.: RoADS: a road pavement monitoring system for anomaly detection using smart phones. In: Atzmueller, M., Chin, A., Janssen, F., Schweizer, I., Trattner, C. (eds.) Big Data Analytics in the Social and Ubiquitous Context. LNCS (LNAI), vol. 9546, pp. 128–146. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29009-6_7
17. Tanaka, N., Okamoto, H., Naito, M.: Detecting and evaluating intrinsic nonlinearity present in the mutual dependence between two variables. Phys. D: Nonlinear Phenom. **147**(1–2), 1–11 (2000)
18. Vittorio, A., Rosolino, V., Teresa, I., Vittoria, C.M., Vincenzo, P.G., et al.: Automated sensing system for monitoring of road surface quality by mobile devices. Procedia - Soc. Behav. Sci. **111**, 242–251 (2014)