SmartRoadSense: Collaborative Road Surface Condition Monitoring

G. Alessandroni, L. C. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti
B. D. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, A. Bogliolo
DiSBeF – University of Urbino
email: alessandro.bogliolo@uniurb.it

***Abstract*—Monitoring of road surface conditions is a critical activity in transport infrastructure management. Many research solutions have been proposed in order to automatically control and check the quality of road surfaces. Most of them make use of expensive sensors embedded in vehicles or mainly focus on detection of specific anomalies during monitoring activity.In this paper, we describe the design of a system for collaborative monitoring of road surface quality. The overall architecture encompasses the integration of a custom mobile application, a georeferenced database system and a visualization front-end. Road surface condition is summarized through a roughness parameter computed using signal processing algorithms running on mobile devices. The roughness values computed are subsequently transmitted and stored into a back-end geographic information system enabling processing of aggregated traces and visualization of road conditions. The proposed approach introduces a thoroughly integrated system suitable for monitoring applications in a scalable, crowdsourcing collaborative setting.**

Keywords–Roughness; Accelerometer; Smartphone; Monitoring; Cloud.

I. INTRODUCTION

Nowadays, all consumer-level mobile devices (e.g., smartphones) feature a rich set of embedded instruments. The presence of triaxial accelerometers and Global Positioning System (GPS) sensors allow the device to track its position and motion states with high degree precision.

Additionally, mobile devices also enable the development of applications that can acquire data from such instruments. Thus, it is possible to access sensor data in real time, store it in memory, handle it using the processing power of the device itself and transmit the data to remote servers using the device's connectivity features.

These features, combined with the ubiquitous and pervasive nature of smartphones and to the inherent scalability of cloud based computing, make possible the design of systems aimed at fine-grained, massive distributed sensing.

In this paper, we propose and describe a system, called "SmartRoadSense", aimed at supporting collaborative monitoring of road surface roughness using mobile smart devices. To this purpose, we designed a three-tiered architecture encompassing: i) a mobile application at user level that processes raw data from the embedded accelerometers and transmits the result of the computation (i.e., a roughness index) together with geographic localization data from GPS to a server; ii) a back-end server running a geographic information system where georeferenced data are properly aggregated, organized and stored; iii) a graphical front-end based on a cloud platform service for visualization.

In order to use data from the accelerometer to study the condition of the road surface, we propose to use Linear Predictive Coding (LPC) [1]. LPC is a method that allows us to predict a particular value in an analog signal by means of a linear combination of the past values of the signal itself. This signal processing technique is used to compute the redundant information contained in the signal. In our case, it can be used to remove accelerations not attributable to irregularities in the road surface.

The mobile application designed in the SmartRoadSense architecture exploits LPC for deriving an estimate of the roughness of a road from sampled points. The values collected by this parameter are computed on board by a smartphone, and transmitted in batch to a remote dedicated server. The back-end server functionalities are in charge of collecting data, mapping traces on the geospatial database and consistently aggregating them for further processing and statistical analysis.

A. Previous Work

Starting in the late 1950s several studies of road surface have proved that its quality is the most important criteria for the evaluation of a road path and its drive comfort. The deterioration of roads leads to added vehicle operating costs, increased fuel consumption (with more emissions to the environment) and increased pavement failures, due to the added dynamic loads of the vehicle [2][3][4].

Several studies have tried to model the road elevation profile, using sine waves, step functions, or triangular waves [5], or as the sum of randomly generated sinusoidal functions with different amplitudes and phases [6]. More recently, it was shown that the spatial Power Spectral Density (PSD) of a typical road surface has a low-pass characteristic, which decreases at the increase of the spatial frequency (measured in cycles/m) [6][7]. In these studies, the road surface profile is modeled as a white Gaussian noise filtered by a first order low-pass filter. It was also shown, that the vertical acceleration of a point following the road profile depends on the horizontal velocity, i.e., the vertical acceleration is related to the car velocity.

A consolidated approach for estimating road surface condition entails the adoption of costly and sophisticated hardware equipment such, for instance, laser profilers [8], specific accelerometers and data acquisition systems [9] whose cost (also taking into account calibration and installation) can be significant.

Another trend of studies explored the feasibility of exploiting low-cost sensors, for instance those embedded in mobile devices such as smartphones. A first work towards this direction have been proposed by Eriksson et al. [10] that built

 ISBN: 978-1-61208-353-7

a system (termed the "Pothole Patrol") targeted at monitoring road anomalies. They used a set of accelerometers and GPS devices deployed in embedded computers in cars. The sampled signals, processed by a given set of filters to remove artifacts and noise, are given as input to machine learning algorithms for detection of potholes and road anomalies. Mohan et al. introduced "Nericell", a road and traffic monitoring system based on smartphones [11]. Sensing devices embedded in smartphones (namely microphones, accelerometer and GPS) are exploited for detecting potholes, bumps and also other traffic related events such as braking and honking.

Our work shares some features with these approaches while we believe it, differs in several aspects. First, while previous works mostly focus on given events for monitoring road quality such, for instance, pothole detection, we aimed at building a continuous monitoring system by assigning a numerical value to each of the samples of the sensed signals by means of LPC algorithms, resulting into a roughness index for potentially each point the monitored road. Second, we integrate the information gathered from several different users into a single consistent aggregated stream thus opening the way to statistical analysis and data fusion techniques for possible error compensation. Third, we take advantage of scaling capabilities provided by cloud computing facilities providing a suitable interface of our system to cloud based platforms, an example of which is given by SmartRoadSense visualization engine, therefore making it possible collaborative crowd-sensing.

B. Contribution and Organization

This paper introduces a system for measuring road quality based on low-cost sensors, a mathematical model to extract a quality index from sensor data, and a software architecture system which allows measurements to be collected and aggregated in an average estimate of road roughness.

The mathematical model upon which we developed our signal processing algorithm is described in section II. The design choices and system-level features of adopted and implemented software components are introduced in section III. Finally, preliminary experiments based on the current implementation are shown in section IV. Concluding remarks, open issues and future work are discussed in section V.

II. Mathematical model

In this section, we describe the mathematical model used to extract information of the road surface conditions.

According to [6], the road surface profile $w(x)$ can be modeled as white Gaussian noise filtered by a first order low-pass filter. The white Gaussian noise has the following autocorrelation function $\rho_{ww} = q\delta(x)$, where q is the PSD magnitude and $\delta(x)$ the Dirac delta function. PSD is given by $S_{ww}(\Lambda) = q$, where Λ is the spatial frequency measured in cycles.

The first order low-pass filter has frequency response

$$H(\Lambda) = \frac{1}{p + j2\pi\Lambda}. \tag{1}$$

Thus, the PSD of the road elevation profile $S_{rr}(\Lambda)$ is given by

$$S_{rr}(\Lambda) = S_{ww}(\Lambda)\,|H(\Lambda)|^2 = q\left|\frac{1}{p + j2\pi\Lambda}\right|^2. \tag{2}$$

In this model, the statistical properties of the road profile are completely characterized by parameters q and p.

Let us consider an ideal point closely following the road profile and moving with constant horizontal velocity v. From equation (2), it can be proved that the vertical acceleration has a continuous time Fourier transform given by

$$A_y(f) = \frac{(j2\pi f)^2}{p + j2\pi f\frac{1}{v}} W(f), \tag{3}$$

and has the following temporal PSD

$$S_{A_yA_y}(f) = qv\left|\frac{(j2\pi f)^2}{pv + j2\pi f}\right|^2. \tag{4}$$

Thus, road parameters q and p of (2) can also be obtained by analyzing the PSD of the vertical acceleration.

The scenario of an accelerometer embedded in a mobile device, rigidly anchored inside the car cabin, is very different from that of an ideal point following the road profile. The accelerometer senses the road through tires, suspensions, and the mechanical coupling with the car cabin. In real applications, the PSD in (4) is sensed by the accelerometer filtered by an unknown transfer function modeling the effect of tires, suspensions, and mechanical coupling. The waveform detected by the accelerometer originated by the road profile is a noise signal with a large spectral content that depends on the road parameters q and p. The accelerometer samples the waveform at a given sample frequency F_s and outputs a discrete time vector signal composed by the triaxial components $a_x(n)$, $a_y(n)$, and $a_z(n)$, according to some internal axial reference. Figure 1 shows an example of the three components recorded by a Motorola G smartphone on a car following a straight road at 40 km/hour. The broadband noise behavior is apparent from the figure.

Other undesired contributions add to this waveform. Indeed, the accelerometer also senses the gravity acceleration, vehicle accelerations, centrifugal accelerations at curves, roll, pitch, and yaw accelerations due to road trend, and vibrations due to the engine. These contributions to the signal have a significant magnitude that can entirely mask the acceleration fluctuations due to the road profile. Nevertheless, some of these accelerations vary slowly and have a low spectral content, others have a periodic spectral content (e.g., vibrations caused by the engine). Thus, the undesired contributions can be removed with a prediction filter, that estimates the accelerometer current sample $a(n)$ (with $a(n) = a_x(n)$, $a_y(n)$, or $a_z(n)$) from past samples, i.e., with an LPC analysis [12], [13]

$$e(n) = a(n) + \sum_{i=1}^{N} \lambda_i\; a(n-i), \tag{5}$$

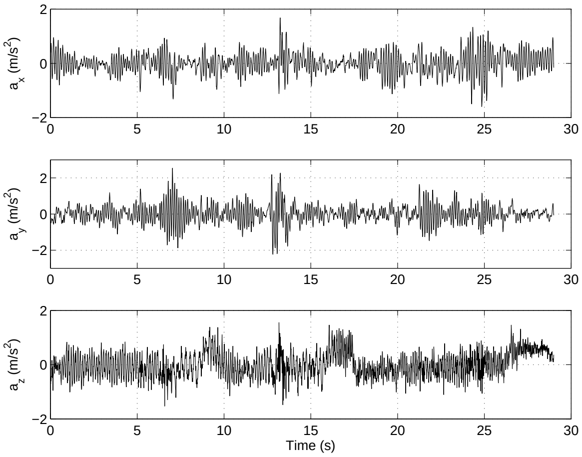

Figure 1. Triaxial accelerations components measured with a Motorola Moto G smartphone on a car at 40 km/h.

where λ_i, with $i = 1, \ldots, N$, are the LPC coefficients, N represents the prediction filter memory length, and $e(n)$ the residual prediction error.

In order to compute the prediction filter and the prediction error, a block based approach is applied: the signal $a(n)$ is split in segments of length M, with M sufficiently large to have an accurate estimate of the prediction filter and, at the same time, sufficiently small to be able to consider the signal stationary. The prediction filter is computed with the Levinson-Durbin recursion [14][15] summarized in Table I (using the Matlab notation). $R(0)$, $R(1)$, $\ldots$, $R(N)$ is the autocorrelation sequence on $a(n)$ estimated over a segment; $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$ is the prediction filter coefficient vector.

TABLE I. PSEUDO-CODE FOR LEVINSON-DURBIN RECURSION.

```
k = R(2)/R(1);
λ = k;
E = (1 − k^2) ∗ R(1);
for i = 2 : N
    k = (R(i + 1) − λ ∗ R(2 : i))/E;
    λ = [k, λ − k ∗ λ(i − 1 : −1 : 1)];
    E = (1 − k^2) ∗ E;
end
```

The prediction error $e(n)$ maintains the information on the road parameter q (which is a proportionality parameter in the PSD) while the information on the parameter p is lost in the signal whitening produced by the prediction filter. Thus, a parameter proportional to q can be obtained estimating the power of the prediction error P_{P_E} on each segment

$$P_{P_E} = \frac{1}{M} \sum_{n=1}^{M-1} e(n)^2. \tag{6}$$

An index of the road roughness, R_I, is eventually obtained by averaging the power of the prediction error for the three axial components

$$R_I = \frac{1}{3} \left(P_{P_{E_X}} + P_{P_{E_Y}} + P_{P_{E_Z}} \right). \tag{7}$$

Figure 2 shows the behavior of the roughness index R_I and the smoothed roughness index as a function of time on a car following a straight road at 40 km/h. The smoothed roughness index has been obtained by averaging R_I with a sliding window of 11 samples.

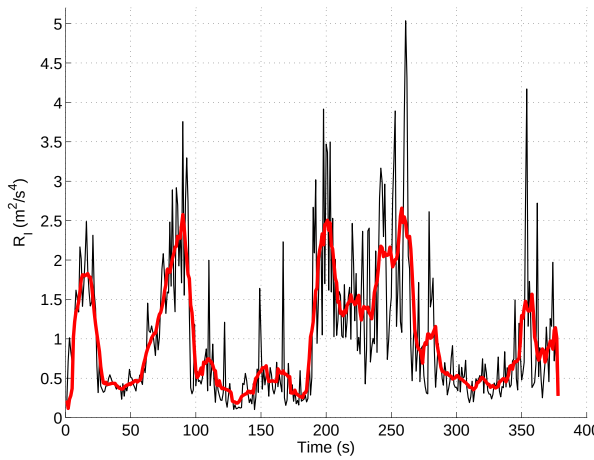

Figure 2. R_I (black curve) and smoothed roughness index (red curve) on a car at 40 km/h.

While this parameter is sensitive to the specific car, the mobile device, and guide style of the driver, the compact information contained in this single parameter can be easily collected and stored. Data from different vehicles, different devices and different drivers on the same road can be combined to achieve a meaningful metric of the road quality.

III. SYSTEM ARCHITECTURE

The mathematical process described above was provisionally implemented using Matlab. After testing on tracks of data collected using real devices, the algorithm was ported to the Java programming language.

In order to perform experiments on real data and to aggregate information collected about road conditions, the following system architecture has been devised: the Java algorithm is embedded in an Android application which runs on an Android mobile device. The application gathers accelerometer data, thus computing and storing P_{P_E} results annotated with GPS data about the device's position. Data is periodically sent to a remote server, which uses a geolocalized database to link each data point to specific roads. Results are aggregated and provided to the user as a geographical map whose roads are enriched with data about their estimated roughness.

The software architecture, graphically displayed in Figure 3, is described in more detail in the following sections.

A. *Android application*

The SmartRoadSense project is built around an Android Application, displaying a user-friendly interface and relying on a background service that gathers data from the device's sensor and processes it to compute P_{P_E} values in real-time. Results are stored in memory along with GPS data, bearing,

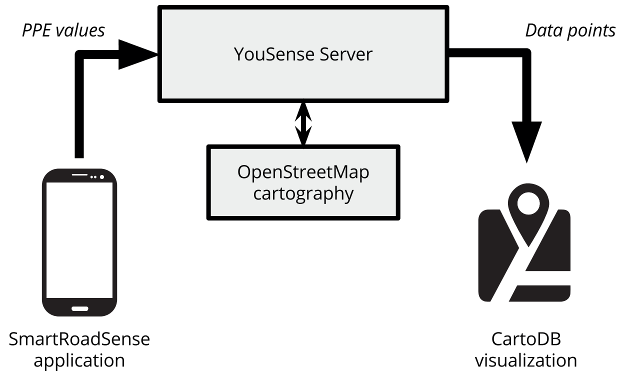

Figure 3. Software architecture of SmartRoadSense.

velocity, timestamp and other metadata. Each recording session is identified by a unique "track ID" (identifying both the device and the session).

Both sensors, triaxial accelerometers and GPS, are run at their highest possible frequency. The resulting data rate depends on the device used: usually accelerometers provide data with a frequency of 100 Hz (or higher) and GPS works at 1 Hz once fixed.

Data gathering is limited by the lower frequency (i.e., the GPS sensor), thus the application records a sample of data once per second on average. As described before, to allow real-time computation, the algorithm operates on windows of data: currently a single P_{P_E} result is extracted from a total of 100 samples, of which 25 are taken from the previous window and 25 from the next window (giving a total overlap of 50 samples). Thus, each result is computed from a total of 100 samples, extracted from an average of 100 seconds of data gathering.

The collected results are periodically transmitted to a remote server (each 15 minutes on average, when a data connection is available).

B. Data collection and aggregation

Data collection server is implemented using the ASP.NET platform on a Linux machine running Mono. A PostgreSQL database with PostGIS extensions acts as the storage back-end for all collected data.

The server application exposes a set of RESTful HTTP APIs that can be used by registered users in order to submit data. As described before, raw roughness data computed by the devices is gathered together with accessory GPS information and track ID metadata. Data points are indexed by geographical position for fast access.

A background process collects all new data points recorded and uses the new data to update information about road roughness. This process is executed periodically (at the time it runs once a day). This process works as follows: the set $P_{new} = \{P_1, P_2, \cdots, P_n\}$ of new data points registered by the server is collected; each point in P_{new} is mapped to the closest road, using a geographical database (in our implementation, we use the open road data available from OpenStreetMap). Roads are represented by a geometric path, thus the mapping of points will yield a set $R_{new} = \{R_1, R_2, \cdots, R_m\}$ of paths, representing all roads for which the database has new data points; each road in R_{new} is updated by extracting points at regular distance intervals from one end to the other (see Figure 4). Thus, each road contributes with one or more averaging points for which the roughness will be estimated; for each averaging point (shown as black dots in Figure 4) all existing data points in a given range are extracted from the database and contribute to the final average roughness value. At the moment, these values are computed as the average between all values.

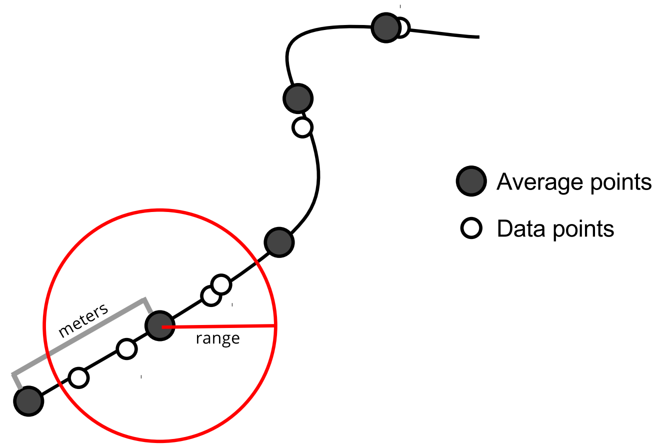

Figure 4. From single data points to average roughness points.

After this process has completed, the database has a new set of average roughness points for each road for which new raw data was sampled.

C. Data visualization

Once the average roughness data has been extracted from the raw dataset and mapped to a road, the final data is pushed to an external CartoDB server: CartoDB is an online service allowing easy visualization and handling of maps with rich data overlays. It is well suited to represent geographic maps with a great amount of data points, while allowing the user to filter and manipulate the data.

An example of the CartoDB interface (using Google Maps cartography) with the average roughness points overlay is shown in Figure 5: the road is filled with equidistant points, colored ranging from green to red. Red points mark positions where the average P_{P_E} value was relatively high, indicating a bumpy road. Green points, on the contrary, indicate low P_{P_E} values, which means that the vehicle was traveling smoothly.

On the left of Figure 5 a sample screenshot of the Android application is shown, indicating the last P_{P_E} value computed.

IV. PRELIMINARY EXPERIMENTS

Two Motorola Moto G smartphones have been equipped with the SmartRoadSense application and have been setup in order to automatically record track data when in movement and to transfer the collected data to the central server every 15 minutes.

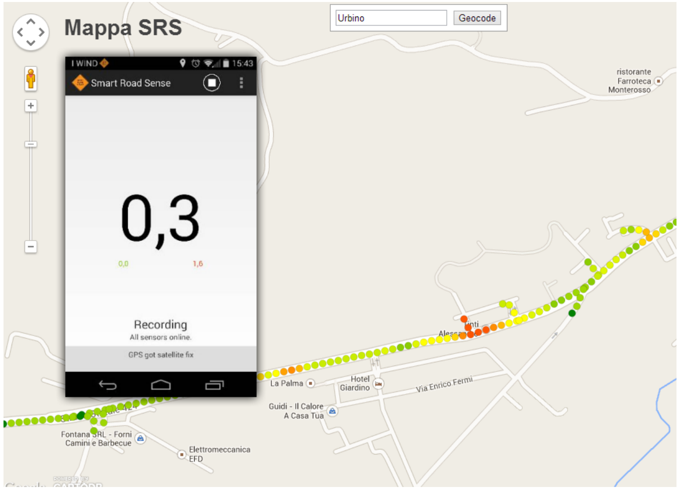

Figure 5. Application screenshot and sample data projection on the map.

The devices have been installed inside two public busses, owned by a local transportation company, using a mobile device rigidly anchored inside the bus cabin. Both busses run twice daily between the cities of Fano, Marotta and Pergola (Marche, Italy). Data was collected over the course of two weeks, from April 1st 2014 to April 16th 2014, totalling ca. 215300 data points. Those points match a total of 744 roads, according to the OpenStreetMap database used (which includes various segments of roads, crossings, etc.), which account for 275089 meters of coverage. On average, data points collected were associated to a road at 5.19 m of distance during the road matching phase of the aggregation algorithm. This gives an estimation the raw GPS data precision. Data occupation of all the collected data amounts to approximately 95 MB (including the overhead given by the PostgreSQL database and indexes).

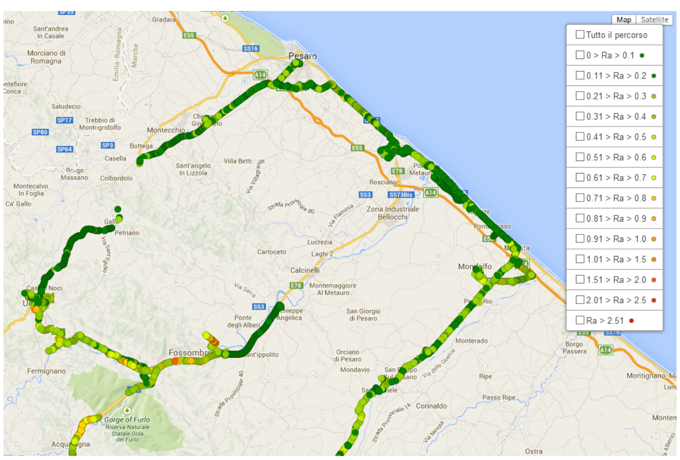

Figure 6. Map of data collected from the experiment.

The resulting data, processed by the server and averaged over equidistant aggregation points, is shown in Figure 6 as an overlay to a geographical map provided by Google Maps. Each shown point represents an aggregated roughness point, whose color varies from red to green as previously described in III-C.

V. Conclusions

In this paper, we have shown how a standard mobile device, running a custom-built application while being anchored to a vehicle in movement, is able to collect data that can be used to detect the quality and irregularities of the road surface.

Such data, collected from the triaxial accelerometer and the GPS sensor, can be appropriately processed by the computational power of a mobile device. Moreover, the dependence of the measurements on vehicle's velocity can be appropriately compensated.

As shown in section III, a data acquisition system has been built that post-processes the data collected by mobile devices and is able to compute a compound roughness index, which is reliably applied to roads marked on a geographical map. Raw data collected by inexpensive devices on personal vehicles, thus transformed into a clear overview of road quality, can be used for the benefit of institutions or drivers. For instance, it may be used by local authorities to detect the presence of critical road surface segments, thus focusing expenditure on roads showing higher maintenance needs.

A. *Future work*

The mathematical model described in section II could be improved, for example giving the possibility to analyze the characteristics of the road surface without the constraint of having the mobile device rigidly anchored to the vehicle.

Moreover, the interface of the central data server will be improved in order to expose a well documented Application Programming Interface (API), allowing client applications to submit data, manipulate it and handle registered tracks.

Future work also includes improvements to the SmartRoadSense mobile application, possibly polishing the experience for end-users and providing means of user registration, in order to enable the distribution of the application via "Google Play Store" and make it possible to collect road data from virtually any user willing to contribute to the project.

Finally, we plan further testing and improvements to the aggregation method used by the server to compute final roughness values: at the time of writing, all data points contribute to an unweighted arithmetic average. However, we can envision averaging methods which weigh contributes (for instance based on age) and/or an evaluation of the data point's source quality.

Acknowledgments

The project was partially supported by "Cassa di Risparmio di Pesaro" foundation and the NeuNet cultural association. The experiment was made possible thanks to the cooperation with "Autolinee Vitali" of Fano.

REFERENCES

[1] L. Deng and D. O'Shaughnessy, Speech processing: a dynamic and optimization-oriented approach. CRC Press, 2003.

[2] Society of Automotive Engineers, Ride and Vibration Data Manual (SAE J6A), ser. SAE information report. Society of Automotive Engineers, 1965.

[3] T. Dahlberg, "Optimization criteria for vehicles travelling on a randomly profiled road–a survey," Vehicle system dynamics, vol. 8, no. 4, 1979, pp. 239–352.

[4] National Quality Initiative Steering Committee, "National highway user survey," Washington, DC, 1996.

[5] J. Y. Wong, Theory of ground vehicles. Jhon Wiley and Sons, Inc, 2001.

[6] T. D. Gillespie, Fundamentals of vehicle dynamics (R-114). SAE International, March, 1992.

[7] M. Ndoye et al., "Sensing and signal processing for a distributed pavement monitoring system," in Digital Signal Processing Workshop, 12th-Signal Processing Education Workshop, 4th. IEEE, 2006, pp. 162–167.

[8] J. Laurent, M. Talbot, and M. Doucet, "Road surface inspection using laser scanners adapted for the high precision 3d measurements of large flat surfaces," in 3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in. IEEE, 1997, pp. 303–310.

[9] M. Ndoye, A. M. Barker, J. V. Krogmeier, and D. M. Bullock, "A recursive multiscale correlation-averaging algorithm for an automated distributed road-condition-monitoring system," Intelligent Transportation Systems, IEEE Transactions on, vol. 12, no. 3, 2011, pp. 795–808.

[10] J. Eriksson et al., "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. ACM, 2008, pp. 29–39.

[11] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in Proceedings of the 6th ACM conference on Embedded network sensor systems. ACM, 2008, pp. 323–336.

[12] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals," in Transactions on Acoustics, Speech and Signal Processing. IEEE, Jun. 1979, pp. 247–254.

[13] L. B. Jackson, Digital filters and signal processing, 2nd ed. Boston: Kluwer Academic Publishers, 1989.

[14] N. Levinson, "The wiener RMS error criterion in filter design and prediction," J. Math. Phys., vol. 25, 1947, pp. 261–278.

[15] J. Durbin, "The fitting of time-series models," Revue de l'Institut International de Statistique, 1960, pp. 233–244.

 ISBN: 978-1-61208-353-7