

Automatic Classification of Road Traffic with Fiber Based Sensors in Smart Cities Applications

Antonio Balzanella, Salvatore D'Angelo(✉), Mauro Iacono, Stefania Nacchia, and Rosanna Verde

Dipartimento di Matematica e Fisica, Università Degli Studi Della Campania "L. Vanvitelli", Viale Lincoln 5, 81100 Caserta, Italy
{antonio.balzanella,salvatore.dangelo,mauro.iacono,stefania.nacchia, rosanna.verde}@unicampania.it

Abstract. Low cost monitoring of road traffic can bring a significant contribution to use the smart cities perspective for safety. The possibility of sensing and classifying vehicles and march conditions by means of simple physical sensors may support both real time applications and studies on traffic dynamics, e.g. support and assistance for car crashes and prevention of accidents, and maintenance planning or support to trials in case of litigation.

Optical fibers technology is well known for its wide adoption in data transmissions as a commodity component of computer networks: its popularity led to large availability on the market of high quality fiber at affordable price. As a purely physical application, its optical properties may be exploited to monitor in real time mechanical solicitations the fiber undergoes. In this paper we present a novel approach to using optical fibers as road sensors. As quite popular in literature, fiber is used to sense the vibrations caused by vehicles on the road: in our case, signals are processed by functional classification techniques to obtain a higher quality and a larger flexibility for the reuse of results. Classification aims at enabling profiling of road traffic. Moreover in our approach we would like to optimise the analysis and classification computations by splitting the process among edge nodes and cloud nodes according to the available computation capacity.

Our solution has been tested by an experimental campaign to show the suitability of the approach.

Keywords: Optical fiber · Sensing · Automatic classification · Smart cities

1 Introduction

The smart cities paradigm promises the provisioning of a vast basket of advanced services for citizens, society and managers. Achieving this goal requires a number

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12252, pp. 31–46, 2020.
https://doi.org/10.1007/978-3-030-58811-3_3

of factors to enable success: as first, a proper vision over all the city framework and needs; planning capacity and skills; a holistic maintenance policy both for preventive and corrective interventions; an ubiquitous communication infrastructure; the availability of affordable, dependable, available, reliable and secure computing resources to run supervision, management, monitoring, control and service applications; proper scalable and reconfigurable software solutions; and, finally, heterogeneous sensing hardware capable of covering all the needs of the applications all over the city. Out of these factors, the ones related to software and computing capabilities may be easily allocated in one or more datacenters, while the communication infrastructures are in general already available or easily deployable in terms of a (usually existing) wired backbone and a last mile based on wireless technologies: installation or integration of this infrastructure may be done at affordable costs that may be shared with other uses, and rarely, in the urban tissue, pose significant challenges. The deployment of sensors may instead be problematic, because of the needed capillarity of coverage to guarantee sufficient information, of the installation costs, of the needed protection apparels against vandalism or environmental weariness, and connected costs are non negligible both for the possible recurrence and the number of installation needed.

Alternative solutions to the same sensing need may be available, and fit for different scenarios: this is the case of interest of this paper, in which the aim of sensing activities is road traffic monitoring. Two possible solutions are the use of cameras, with subsequent image processing, and the use of optical fiber as vibration sensors, with subsequent signal processing. Cameras may cover large spaces and are a traditional solution, allowing both automatic detection and human surveillance: image processing is a well established approach, but it is easily affected by errors when monitoring is not related to small areas and does not provide direct knowledge on dynamics and details of traffic, vehicles and road. Cameras have the significant advantage of being installed easily if there is room, but optics and in general the whole device may be easily damaged or stolen, or disturbed by dirt or other obstructions, and may pose problems in terms of privacy, or may be hacked. Fiber has a very modest acquisition cost but a high installation cost, as it is generally deployed inside the top level of the road structure to catch vibrations when vehicles pass by, and, additionally, to statically diagnose road deformations or problems, as every alteration of its condition influences light transmission and is measurable: fiber is only affordable when building or rebuilding road infrastructures, or for highways, but also provides remote diagnostic capabilities and detailed position and movement information independently in each point of their length, pose no privacy issue and are protected against weariness or vandalism by the road itself.

In this paper we focus on the use of optical fiber as a road sensor. The system we propose follows the most shared choices of solutions existing in literature, with some variations in the setup, in the clustering approach and in the testing campaign: the original contribution of this paper is the introduction of a pre-processing step for extracting appropriate features from DAS signals

and an online clustering algorithm which discriminates the type of vehicles. The proposed algorithm is also able to evaluate the evolution of the clustering model itself over time through a continuous stream analysis of the data coming from the fiber road sensors. The use of on-line algorithms allows to analyze real time data and to cope with devices pervasively deployed in a smart city scenario which might not be performing enough to carry put the entire execution.

The need to analyze the data collected from the distributed sensors to make real-time intelligent decisions to control the system is nothing new. In fact, these decisions are often performed on data that is continuously streaming from the edge devices at high input rates [19]. Since not every analysis can be performed directly by or nearby the sensors, as they may be too computing-intensive, some of the computation, according to the problem that must be solved, can be executed in a more performing data center. Splitting the computation among edge devices and datacenter infrastructure is more and more common in data-stream analysis [8].

For this reason, our approach aims at following the evolution of the model over time by splitting the overall analysis and classification computation among the edge nodes, thus nearby the optical fiber sensors and the datacenter, according to the needs. This may also be of fundamental importance if we consider the extensive use that can be done of optical fiber for road sensing and control, that requires the system to correctly scale and distribute the computation, in order to grant a certain level of performance and results quality.

The paper is organized as follows: in Sect. 2 related works from literature are presented and examined; in Sect. 3 the approach followed in this paper for the definition of the system and the analysis of data is presented; in Sect. 4 the feasibility and the applicability of the approach are demonstrated; finally, conclusions close the paper.

2 Related Works

Considering the physical and financial growing of modern smart cities, researchers must look out for new strategies and technologies to improve traffic congestion, as it is known as the main cause of air pollution in urbanized areas. In recent years, technological developments have enabled collection and transition of real-time traffic information, and this, coupled with the increase of traffic trips and congestion, has increased the interest in traffic modeling, not only to improve traffic conditions but also to monitor traffic flows and detect potentially dangerous situations.

Advanced traffic management systems (ATMSs) [20], such as adaptive traffic control (ATC), are enabling a higher efficiency of the traffic management ecosystem and can help integrate the expected growth in vehicle numerousness without overwhelming existing infrastructures.

One specific area, which is also the focus of this work, is the *automatic vehicle identification (AVI)*, that refers to the technology used to identify a particular vehicle when it passes a particular point. Early development of AVI occurred in

the United States, beginning with an optical scanning system in the 1960s to identify railroad boxcars [12]. AVI systems can be of various types, such as automatic toll collection systems [4], vehicle-mounted transponders of different types and roadside beacons [18], video cameras and license plate matching techniques [15,28], and the more recent Bluetooth and WIFI based detection systems [1,3]. However, these systems are partly subjected to some inefficiencies:

- video based solutions and technologies have high costs of installation and maintenance;
- the detection effect is also highly sensitive to environmental and weather conditions;
- wireless sensor network technologies are affected by the communication environment and nodes performances.

In contrast, the distributed acoustic sensing technology, that is the use of distributed optical fiber acoustic sensing (DAS), is being used more and more for traffic detection, discrimination and counting [16]. The solutions based on the DAS technology are more sensitive and offer lower costs and higher resistance to temperature, corrosion and interference phenomena. The first experiments involving the DAS technology revolved around the vertical seismic profiling [14] and surface seismic surveys with active sources [5,24]. However, due to their low cost and low power consumption, a large number of these sensors can be deployed in a certain area for purpose of detection, classification, identification and tracking of approaching targets; thus they have been lately used, also quite extensively, to detect and classify vehicles in order to improve traffic flow management.

In [17] the authors studied traffic-related seismic vibrations measured by an urban geophone network with a spacing slightly shorter than a typical city block dimension, a vibration source with a speed of $25 \pm 3\,\mathrm{m/s}$ was detected on the expressway. However, since there were no traffic flow data available, it was ultimately impossible to verify whether the vibration source was the vibration caused by the vehicle.

On the other hand [13] describe the deployment of a shallow gully (DAS) array on a highway where cars on the road parallel to the array direction were the main source of noise. Analysis of the source of the beam-form confirmed that most of the noise came from cars driving on adjacent roads.

Important results have also been reported in [11]; here the authors use a wavelet-denoising algorithm and a dual-threshold algorithm to reconstruct the signal for feature extraction, and the vehicle count and speed are obtained. When all features have been extracted, the classification of vehicle types is implemented by a support vector machine (SVM) classifier.

As explained in the previous section, another focal point to ensure a performing streaming analysis as well as a scalable solution. We would like to distribute the computation among computational nodes and sensors close to the fiber and the datacenter. In the current state of the art, this strategies are referred to as "cloud/edge analytics" and there are many research activities that withness the

successful aspects of deploying the analysis on both cloud and edge according to the computational needs of the tasks to be performed on the data.

Authors of [22] analyze performance trade-offs of hybrid cloud/edge-based in the data analysis scenarios using multiple workloads, reporting how hybrid cloud/edge processing speeds-up stateless and simple stateful operations: specifically, edge computing helps reduce the amount of data being uploaded to the cloud and, consequently, improves the overall performance, in terms of end-to-end throughput, and better supports real-time requirements. In [23] the authors propose a novel distributed deep neural network architecture (DDNN) that is distributed across computing hierarchies, consisting of the cloud, the edge and end devices to improve object recognition accuracy. In [25] the authors present a new computing framework, namely Firework, that facilitates distributed data processing and sharing for IoE applications via a virtual shared data view and service composition, specifically crafted for real-time video analytics. Vigil [26] is a distributed wireless surveillance system that partitions video processing (e.g., object/face recognition or trajectory synthesis) between edge nodes and the cloud with a fixed configuration.

Focusing on the challenge of data clustering, the algorithms which can be run on the edge of the overall system architecture have to fulfill some stringent restrictions: data arrive continuously; single instances have to be processed as soon as they arrive; the size of a stream is potentially unbounded; data objects are discarded after they have been processed; the monitored phenomenon evolves over time.

A recent survey [21], reviews the methods addressing the data stream clustering issue posing the accent on the data structure used for summarizing data, on the capability to discover outliers and to deal with data evolution, on the required input parameters, on the shape of the discovered clusters. Some of the most effective methods, also emerging from such review, are [2,7,27].

3 Architecture and Processing Strategy for DAS Data

Since the traffic monitoring and management has become more and more important and challenging in the current smart cities environment, as stated in the previous section, our main objective is to recognise and classify the signals coming from the deployed optical fiber used as road sensor. In order to achieve this we propose a scalable and distributed architecture and a specific process for sensing and processing the signals coming and for performing the online classifications of the vehicles.

3.1 Overall System Architecture

The general structure of the proposed system is composed of 3 main functional blocks (see Fig. 1) that satisfy the two main duties of the system, sensing and processing, in the most flexible way, to reduce costs and facilitate management,

expansion, upgrading and maintenance while lowering the costs. The blocks are: the sensing infrastructure, the interconnection network and the processing facility.

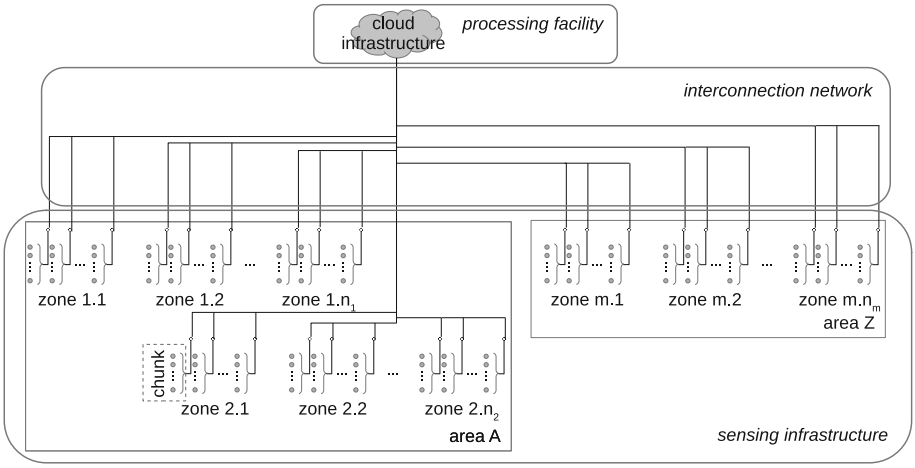

Fig. 1. Logical organization of the system

The *sensing infrastructure* consists in a number of fibers, distributed along the road network that has to be monitored and hierarchically organized into areas, zones and chunks. A chunk is a linear section of a road covered by a number of fibers, that is considered as the elementary component of the sensing infrastructure; a zone is a contiguous portion of the road system (e.g. the roads that embrace a number of blocks or a portion of a highway) the chunks of which are managed as a unit by the interconnection network[1]; an area is a logical portion of the road network, including many zones, that is considered a management unit, e.g. a district, a town, a highway.

The *interconnection network* is the communication infrastructure that collects data from the sensing infrastructure and vehicles it towards the processing facility. Standard networking technologies, possibly exploiting an existing network, are used, and a sensing hub exists per zone, with the task of collecting data from fibers and send them to the computing facility by standard TCP/IP based messaging.

The *processing facility* is implemented in the cloud[2], to minimize costs and exploit advantages such as elasticity and centralized management and

[1] This is similar to what happens in the organization of sensing and control in other transportation networks, e.g. the high speed trains standard ERTMS/ETCS [6].

[2] An alternative solution, that offers the possibility of having zone services enabled and enacted locally, is to design, at a higher cost, the software infrastructure as a edge computing infrastructure, e.g.. to implement real time services computed zone by zone locally in the sensing hub.

maintenance, and to integrate data processing with other functionalities provided by other software services in the smart city solution.

In the rest of the paper, we focus on the end-to-end computing process for data from a single fiber, for the sake of simplicity.

3.2 Workflow

The general workflow applied by the system is divided into 4 phases: sensing, processing, de-noising, classification (see Fig. 2).

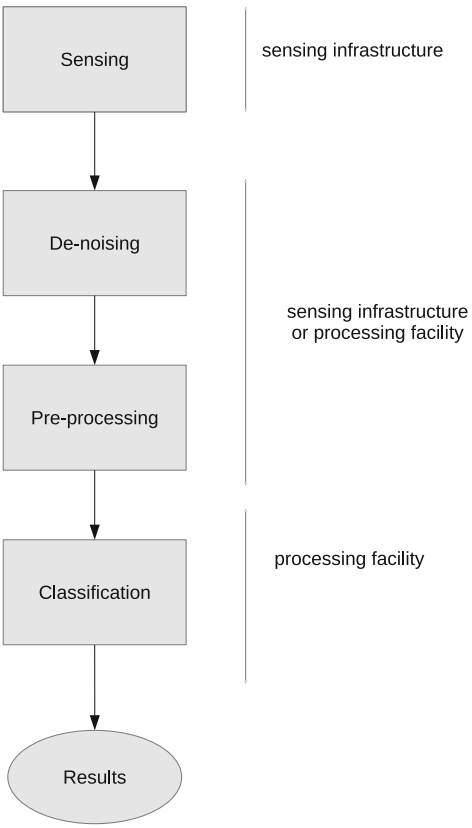

Fig. 2. The workflow and its mapping on the organizational blocks of the system

Sensing - In the *sensing* phase, that is performed by a fiber, the rough signal is measured in space, along the length of the fiber, and time, in terms of variation of the sensed vibration in each point of the fiber in each instant, actually organizing the fiber in sections to compensate the non-ideal conditions in which signal is generated and propagated. The sensing phase is divided into two sub-phases:

– The pose of the fiber
– Events detection

The laying of the fiber can be external or internal, the latter is in general the best configuration to avoid noise as it is blocked by immersing it in a material or fixing it on supports. Another challenging aspect of the sensing phase is the right sampling phase and acquisition rate that must be chosen correctly to not lose valuable information.

Pre-processing and De-noising - In the *pre-processing phase*, the physical signal is measured and converted to be manipulated as a sequence of images that represent the state of the fiber and its evolution over time. However to better apply the clustering algorithm or, more generally, to perform data analysis algorithms on the signal, we have defined a sequence of steps to clean up or better to *de-noise* the signal as it can be highly subjected to noise in the sensing phase but also because the final signal can have a high dimensionality that can badly affect the final clustering results. In the *de-noising* phase, standard de-noising numerical techniques are applied to enhance data readability and improve the effectiveness of the classification and to better identify the sections where we can clearly see the transit of the vehicle. Specifically, in the de-noising phase we have opted for the wavelet transform, [10], a powerful tool for the analysis and processing of signals which is extremely efficient in various fields such as compression and de-noising, and in general when dealing with not stationary signals. The discrete wavelet transform (DWT) is a linear signal processing technique, it transforms a vector into a numerically different vector (D to D') of wavelet coefficients. The two vectors are of the same length, however it is useful for compression in the sense that wavelet-transformed data can be truncated; a small compressed approximation of the data can be retained by storing only a small fraction of the strongest wavelet coefficient e.g., retain all wavelet coefficients larger than some particular threshold and the remaining coefficients are set to zero. One interesting aspect is that the resulting data representation is sparse, thus the computations that can take advantage of sparsity are very fit if performed in wavelet space. Given a set of coefficients, an approximation of the original data con be got by applying the inverse DWT.

Once we have obtained a better signal, both in terms of nodes and in terms of dimensionality reduction, we use the Hough's transform, [9], which is a feature extraction technique used in image analysis, computer vision, and digital image processing. This technique aims at isolating the features that characterize a specific shape in an image, and proved to be specially useful in obtaining an overall description of one or more features from noisy local measures when there is no previous knowledge about the number of solution classes. Each point, expressing a measure, contributes to a globally consistent solution, e.g. a line we want to recognize as such. The classical Hough's transform is concerned with the identification of lines in the image, and use it to find sheaf of parallel lines that identify the vehicle transit and, ideally, the inclination of such lines should suggest the vehicle speed.

In *clustering* phase, that is described and discussed in details in the rest of the Section, images are used to extract knowledge from the available signals and to recognize the type of Tracy Lorraine the vehicles that transit along the fiber.

3.3 Data Definition and Clustering Approach

Hough's transform provides one or more lines for each image. Among the lines, we select one or more sets of parallel ones, which provide information about behaviors of vehicles in transit, anomalies, new emerging trends.

Considering the parametric equation of a straight line $x \cdot cos(\theta) + x \cdot sin(\theta) = r$, the set of J parallel lines ℓ_j $(j = 1 \dots J)$ is described by the orientation θ and by r_j with respect to the X-Axis. Thus, for each image we get a set of tuples $\{(\theta, r_j); j = 1 \dots J\}$, with constant θ.

We consider as input of the clustering procedure a set of on-line arriving events E_i which represent the set of tuples of each image, as: $E_i = (\theta_i, a_i)_{i=1,\dots,\infty}$, where $a_i = [max(r_{i,j}) - min(r_{i,j})]$ $(j = 1 \dots J_i, i = 1 \dots M)$. That is, we focus on the orientation and on the distance between the external parallel lines of the sheaf to capture information about vehicle transit.

The clustering process aims at discovering groups of similar vehicles according the two mentioned features. It is made by three main phases: 1) Initial typologies construction; 2) On-line clustering; 3) Typologies updating and outlier detection. These steps are based on micro-cluster algorithm.

In our implementation of the micro-cluster, we assume that a set of similar events can be summarized by the following set of statistics:

- $\overline{G}$: Micro-cluster centroid;
- MD: Micro-cluster boundary;
- n: Number of allocated items;
- ST: Sum of times;
- SST: Sum of squared times.

The centroid G is the representative of the events summarized by the micro-cluster. Since we assume that each sheaf of parallel lines can be described by two quantities (θ_i, a_i), G is the tuple $[\overline{\theta}, \overline{a}]$ such that a set of events is summarized by the average $\overline{\theta}$ of the θ_i's and by the average $\overline{a}$ of the a_i's.

The field MD records the maximum of the distances between the allocated events and the micro-cluster centroid. This allows to get a measure of the micro-cluster radius.

The parameter n stores the number of allocated events E_i.

The parameters ST and SST record the sum ST and the sum of squares SST of the time instants of each allocation to the micro-cluster. Indeed, by means of ST, SST and n we can get the average and variance of the allocation times which give, for instance, a measure of the obsolescence of a typology or information about if a typology is very recurrent only in recent or old time instants.

To define the initial set of event typologies, we cluster, off-line, an initial set $\mathcal{E}$ of M training events $E_i = (\theta_i, a_i)$ by using a standard k-means clustering

algorithm. The algorithm provides a set of k clusters $C_1, \ldots, C_k, \ldots, C_K$ and the corresponding set of centroids.

For each cluster C_k (with $k = 1, \ldots, K$) we compute the sum of distances to the centroid and the number of allocated items in order to record them in a micro-cluster. The parameters ST and SST are set to 0 since the micro-cluster initialization is performed on the training set.

The second phase analyzes the on-line arriving events. The first step on each event E_i is to evaluate if it can be allocated to an existing micro-cluster or a new micro-cluster has to be added. In the first case, the event E_i can be traced back to one of the existing typologies; in the second case, it can be an outlier or the first one of a new emerging typology.

The choice is performed by evaluating the Euclidean distance between the event and the nearest micro-cluster. If the distance is lower than the boundary MD of the micro-cluster, E_i can be allocated to it by increasing the number n of allocated events and updating the parameters ST and SST summing to their value the time stamp and the square of the time stamp of the event.

Alternatively, the E_i is used as seed of a new micro-cluster having the event as centroid, $n = 1$, ST and SST set to the value the time stamp and the square of the time stamp of the event. Finally, the parameter MD is set in heuristic way to the MD value of the nearest micro-cluster.

The third phase updates the set of typologies in order to reflect the current behavior of the monitored environment and provides a set of outliers.

With the flowing of data it is possible that too many micro-clusters have been generated due to emerging typologies or outliers. We select initially the outliers by bringing out the micro-clusters for which the parameter n is set to 1. Then we run a new k-means algorithm on the micro-cluster centroids to get a new reduced set of typologies.

Different typologies can provide a detection of several directions, number and speed of vehicles.

4 A Case Study

In this section we present the results obtained by applying the workflow steps in a real environment. In our case the fiber has been placed outside the road surface blocked to the ground, as the environment chosen for the experiments has posed a series of technical limitations. Four sections of fiber have been laid down from about 75 m for a total of about 300 m. The four sections are parallel to each other and the light travels in both directions twice. In this way the data has been detected 4 times in order to have a double validation of the detected vibration.

During the detection of test events, the data has been collected. Three different sets of data, depending on the acquisition signal frequency (152 Hz, 19 Hz and 9 Hz), have been collected. The measurement unit returns frames containing the measured values of the 512 "distributed microphones" in 128 s, which at 152 Hz corresponds to a total of 832 ms, at a frequency of 19 Hz the acquisition interval

is about 7 s and the last test at a frequency of 9 Hz the acquisition interval is about 12 s. We have carried out from 5 to 9 car passages for each frequency with two different car models, a B-segment car (approx. 1.105 kg in running order) and a minivan (approx. 1460 kg in running order).

The collected data has been processed in MATLAB by another team that provided us with the raw data. Each file consists of a three-dimensional matrix $sd = z \times tfast_ms \times time$ where:

- z are the "microphone" along the fiber;
- $tfast_ms$ the instants of time in which the intensity of the vibration from the "microphones" is measured;
- $time$ the number of measurement windows elapsed during the experiments.

Every matrix is a frame that represent the acquisition interval (832 ms, 7 s, 12 s). The section of fiber laid in a straight line was about 75 m, so two lines close to each other, in addition to having opposite slope, must be between 0 and 150 m apart (depending on the point where the car was at the time that frame was acquired). In the classic approach, on each frame we look for oblique uniform color bands that indicate a moving object, using Hough's transform to identify the external parallel lines and the inclination of the bundle in order to deduce the speed of the moving object Fig. 3.

For data analysis and cleaning we use python scripts and the numpy, scipy, pandas and pywt (for wavelet decomposition and reconstruction) libraries. These libraries allow us to import matlab data and to define the necessary steps to obtain a clean signal without loss of information.

We first rearrange the matrix by combining the measurement windows into one window. It should be noted that the windows have a delay of approximately 100 ms between them due to the processing time of the windows by the signal acquisition control unit.

We then clean up the matrix of the peaks present at the ends of the fiber. Before proceeding with the signal cleaning steps, it is good to remember that after the unification of the windows the matrix is a two-dimensional matrix. On the row there are 512 "microphones" distributed along the fiber, on the columns the instants of time of the unified windows. Each single line represents the signal measured by a single "microphone", and it is precisely on each of these that we are going to carry out the cleaning of the signal.

The steps we perform for de-noising are the following:

- each of the 512 signals is decomposed with a wavelet transform up to the fifth level using the "db1" wavelet;
- the coefficients of the five levels are then selected using customized thresholds for each level;
- the signal is then reconstructed with the filtered levels.

Once the signal has been cleaned up, we can proceed with the functional analysis and with Hough's transform for the identification of the test steps.

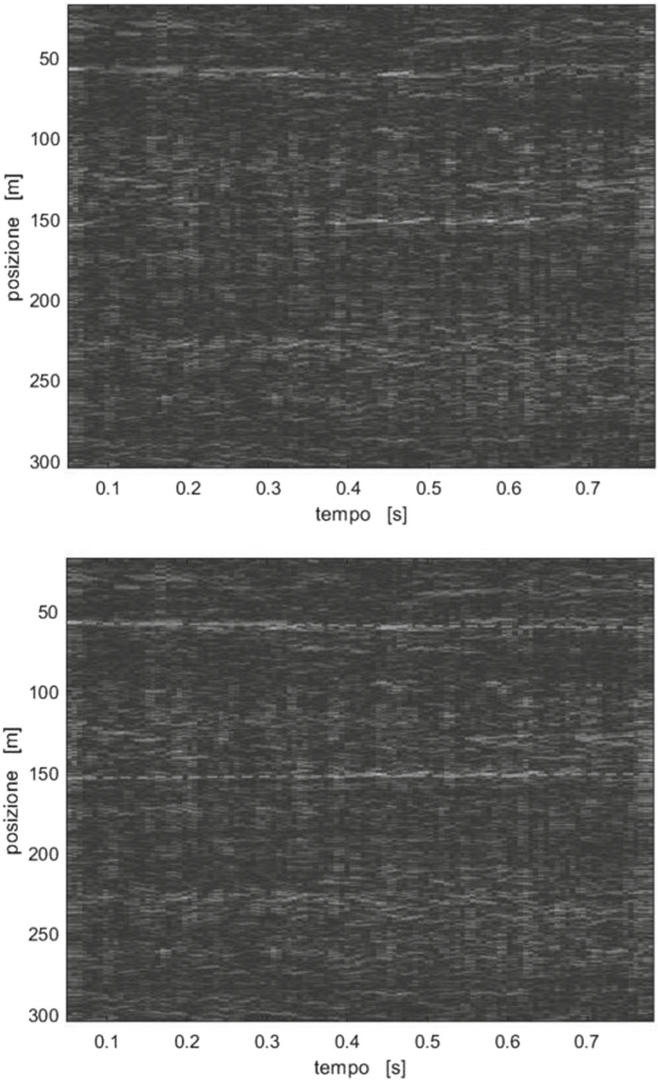

Fig. 3. A frame at 152 Hz with a car passage (top) and the same frame processed with Hough's trasform (bottom)

To apply Hough's transform, the windows relating to the test steps carried out and the windows in which we are sure of the absence of steps are selected. We only translate the windows into space (therefore along the microphones), translated only in time or translated both in time and in space. The first case allows us to verify the presence or absence of a vehicle in space, in the second case we can verify the mobility of the vehicle (if the vehicle moves it is no longer

detected by the selected "microphones"), in the last case we are able to follow the vehicle along its movement.

Below are some images of events allocated to micro-clusters describing vehicle behaviors lines identified with Hough's transform. The data analyzed were those relating to the frequency of 152 Hz, in which the test steps were carried out on the windows:

- 130–151 passage at a speed of 10 km/h (Fig. 4)
- 170–195 passage at a speed of 10 km/h
- 205–216 passage at a speed of 20 km/h (Fig. 5)
- 236–249 passage at a speed of 20 km/h
- 265–272 passage at a speed of 30 km/h (Fig. 6)

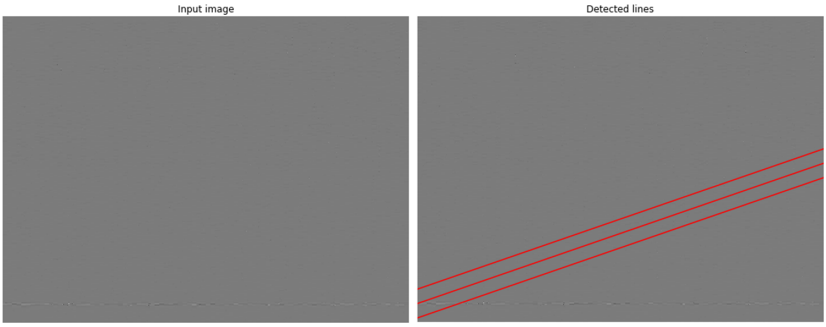

Fig. 4. The fiber section from 130 to 151 at 152 Hz processed with Hough's transform)

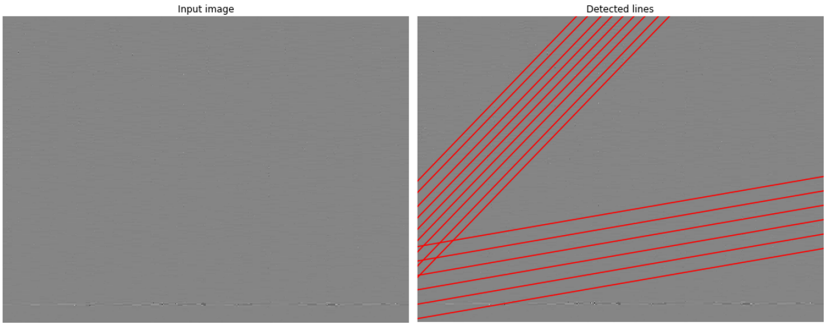

Fig. 5. The fiber section from 205 to216 at 152 Hz processed with Hough's transform)

Following the analysis of the results obtained and studies on the frequencies produced by moving vehicles, we had confirmation of the results obtained,

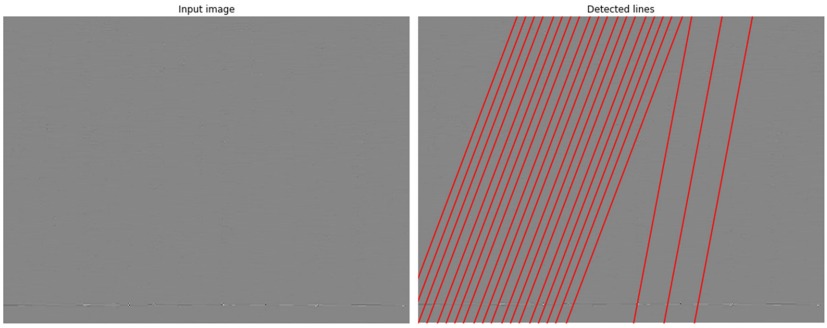

Fig. 6. The fiber section from 265 to 272 at 152 Hz processed with Hough's transform)

as the sound waves produced by vehicles have frequencies ranging from about 150 Hz to rise. For this reason we are already proceeding with further tests at higher frequencies and different setups to get better results and validate the methodology.

5 Conclusions and Future Works

In this article we have shown a new method for traffic monitoring using distributed fiber optic sensors. Some tests have been carried out to validate the presented method, even though they are quite preliminary the results show the possibility to use DAS sensors both to classify the vehicle flow and to detect heavy traffic situations and take actions based on the flow velocity.

Jointly with another team we are working on improving the measuring instruments to get cleaner and better results, specifically in the next experiments we are going to use a better fiber to improve the signal noise ratio of almost 15dB and we would like to experiment more vehicle configurations: not to track just one vehicle movements, but to also track more vehicles driving simultaneously that go for example in opposite direction, or forming queues, in order to have more data about real environmental situations and to improve the classification system to recognize more specific and articulate vehicles configurations without losing accuracy.

In the future, we will continue to work on the possibility to classify not only the vehicle flow but also to be able to classify types of vehicles or groups of vehicles. In addition, experiments may be conducted on road sections using the existing fibre-optic infrastructure provided by telecommunication companies.

Acknowledgements. This work is partially funded by the POR FESR 2014/20 - Project "Quick&Smart" of Università degli Studi della Campania "Luigi Vanvitelli". We are grateful to professor Aldo Minardo for his precious collaboration during the experiments and useful suggestions about the treatment of data.

References

1. Abbott-Jard, M., Shah, H., Bhaskar, A.: Empirical evaluation of Bluetooth and Wifi scanning for road transport (2013)
2. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams, June 2003
3. Barcelö, J., Montero, L., Marqués, L., Carmona, C.: Travel time forecasting and dynamic origin-destination estimation for freeways based on Bluetooth traffic monitoring. Transp. Res. Rec. **2175**(1), 19–27 (2010)
4. Bhavke, A., Pai, S.: Advance automatic toll collection & vehicle detection during collision using RFID. In: 2017 International Conference on Nascent Technologies in Engineering (ICNTE), pp. 1–5. IEEE (2017)
5. Daley, T.M., et al.: Field testing of fiber-optic distributed acoustic sensing (DAS) for subsurface seismic monitoring. Lead. Edge **32**(6), 699–706 (2013)
6. Flammini, F., Marrone, S., Iacono, M., Mazzocca, N., Vittorini, V.: A multiformalism modular approach to ERTMS/ETCS failure modeling. Int. J. Reliab. Qual. Saf. Eng. **21**(1), 1450001 (2014)
7. Gama, J., Rodrigues, P., Lopes, L.: Clustering distributed sensor data streams using local processing and reduced communication. Intell. Data Anal. **15**, 3–28 (2011). https://doi.org/10.3233/IDA-2010-0453
8. Gonzalez, N.M., et al.: Fog computing: data analytics and cloud distributed processing on the network edges. In: 2016 35th International Conference of the Chilean Computer Science Society (SCCC), pp. 1–9. IEEE (2016)
9. Illingworth, J., Kittler, J.: A survey of the Hough transform. Comput. Vis. Graph. Image Process. **44**(1), 87–116 (1988)
10. Lang, M., Guo, H., Odegard, J.E., Burrus, C.S., Wells, R.O.: Noise reduction using an undecimated discrete wavelet transform. IEEE Signal Process. Lett. **3**(1), 10–12 (1996)
11. Liu, H., Ma, J., Xu, T., Yan, W., Ma, L., Zhang, X.: Vehicle detection and classification using distributed fiber optic acoustic sensing. IEEE Trans. Veh. Technol. **69**, 1363–1374 (2019)
12. Martin, B., Scott, P.: Automatic vehicle identification: a test of theories of technology. Sci. Technol. Hum. Values **17**(4), 485–505 (1992)
13. Martin, E., et al.: Interferometry of a roadside DAS array in Fairbanks, AK. In: SEG Technical Program Expanded Abstracts 2016, pp. 2725–2729. Society of Exploration Geophysicists (2016)
14. Mateeva, A., et al.: Distributed acoustic sensing for reservoir monitoring with vertical seismic profiling. Geophys. Prospect. **62**(4), 679–692 (2014)
15. Oliveira-Neto, F.M., Han, L.D., Jeong, M.K.: An online self-learning algorithm for license plate matching. IEEE Trans. Intell. Transp. Syst. **14**(4), 1806–1816 (2013)
16. Parker, T., Shatalin, S., Farhadiroushan, M.: Distributed acoustic sensing-a new tool for seismic applications. First Break **32**(2), 61–69 (2014)
17. Riahi, N., Gerstoft, P.: The seismic traffic footprint: tracking trains, aircraft, and cars seismically. Geophys. Res. Lett. **42**(8), 2674–2681 (2015)
18. Rillings, J.H., Betsold, R.J.: Advanced driver information systems. IEEE Trans. Veh. Technol. **40**(1), 31–40 (1991)
19. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
20. Shahgholian, M., Gharavian, D.: Advanced traffic management systems: an overview and a development strategy. arXiv preprint arXiv:1810.02530 (2018)

21. Silva, J., Faria, E., Barros, R., Hruschka, E., de Carvalho, A., Gama, J.: Data stream clustering: a survey. ACM Comput. Surv. **46** (2014). https://doi.org/10.1145/2522968.2522981
22. Silva, P., Costan, A., Antoniu, G.: Investigating edge vs. cloud computing trade-offs for stream processing. In: IEEE International Conference on Big Data in 2019 (2019)
23. Teerapittayanon, S., McDanel, B., Kung, H.T.: Distributed deep neural networks over the cloud, the edge and end devices. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 328–339. IEEE (2017)
24. Wang, H., et al.: Field trial of distributed acoustic sensing using active sources at Garner Valley, California. In: AGU Fall Meeting Abstracts (2014)
25. Zhang, Q., Zhang, Q., Shi, W., Zhong, H.: Firework: data processing and sharing for hybrid cloud-edge analytics. IEEE Trans. Parallel Distrib. Syst. **29**(9), 2004–2017 (2018)
26. Zhang, T., Chowdhery, A., Bahl, P., Jamieson, K., Banerjee, S.: The design and implementation of a wireless video surveillance system. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp. 426–438 (2015)
27. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: a new data clustering algorithm and its applications. Data Min. Knowl. Disc. **1**(2), 141–182 (1997). https://doi.org/10.1023/A:1009783824328
28. Zhou, J., Gao, D., Zhang, D.: Moving vehicle detection for automatic traffic monitoring. IEEE Trans. Veh. Technol. **56**(1), 51–59 (2007)