

Research Article

Density-Based Statistical Clustering: Enabling Sidefire Ultrasonic Traffic Sensing in Smart Cities

Volker Lücke*n*, Nils Voss, Julien Schreier, Thomas Baag, Michael Gehring, Matthias Raschen, Christian Lanius, Rainer Leupers, and Gerd Ascheid

Institute for Communication Technologies and Embedded Systems (ICE), RWTH Aachen University, Aachen, Germany

Correspondence should be addressed to Volker Lücke; luecken@ice.rwth-aachen.de

Received 18 August 2017; Accepted 6 December 2017; Published 4 January 2018

Academic Editor: Mehmet Yildirimoglu

Copyright © 2018 Volker Lücke et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic routing is a central challenge in the context of urban areas, with a direct impact on personal mobility, traffic congestion, and air pollution. In the last decade, the possibilities for traffic flow control have improved together with the corresponding management systems. However, the lack of real-time traffic flow information with a city-wide coverage is a major limiting factor for an optimum operation. Smart City concepts seek to tackle these challenges in the future by combining sensing, communications, distributed information, and actuation. This paper presents an integrated approach that combines smart street lamps with traffic sensing technology. More specifically, infrastructure-based ultrasonic sensors, which are deployed together with a street light system, are used for multilane traffic participant detection and classification. Application of these sensors in time-varying reflective environments posed an unresolved problem for many ultrasonic sensing solutions in the past and therefore widely limited the dissemination of this technology. We present a solution using an algorithmic approach that combines statistical standardization with clustering techniques from the field of unsupervised learning. By using a multilevel communication concept, centralized and decentralized traffic information fusion is possible. The evaluation is based on results from automotive test track measurements and several European real-world installations.

1. Introduction

In the context of the Internet of Things (IoT), the integral transformation process of urban infrastructure into intelligent and connected devices plays an important role for future mobility. Cities experience an increasingly high level of road congestion due to the focused traffic coming along with progressing urbanization. This congestion induces a high social, economical, and environmental cost. From a European Union (EU) perspective, it is amounted to about one percent of the GDP [1] and predicted to increase by about 50 percent by the year 2050 [2]. From an environmental point of view, carbon dioxide emissions and air pollution, especially locally in cities, are a significant social and health aspect [3]. With this new unprecedented level of traffic density, offline and online traffic planning including live routing techniques becomes a requisite for larger cities, being useful both for end users and for city planners. Especially in the case of real-time traffic monitoring and the subsequent

information provision to drivers and public routing systems, proper conditioning of this traffic information in terms of accuracy, area coverage, and density is paramount. Traffic congestion can only be effectively reduced if this information base, which is used for the routing decisions, is delivered with a sufficient level of quality in these dimensions. Then, with a city-wide traffic status, large-scale end-to-end routing optimizations can be performed. The results can be deployed to the vehicle driver with provision methods such as in-car information systems or centralized traffic guidance for triggering the individual's routing reaction. If this provision is not performed properly either due to an insufficient traffic information quality or improper consideration of group effects in the optimization, even detrimental effects can occur [4]. In this article, these requirements on traffic information quality and density are in focus, building a base for future routing decisions. Many currently used traffic sensing techniques for creating this live data basis will only have a negligible positive impact on the routing decisions

if they are distributed sparsely and do not achieve sufficient coverage.

In general, two main types of techniques achieving more comprehensive traffic information have been established in the last years. A first category is the fixed or mobile infrastructural distribution of sensors, like inductive loops and traffic cameras. The second type is represented by systems relying on end-user based distributed information, for example, movement data from smartphone users or information gathered using V2X communication approaches. Details and characteristics of several techniques are discussed in this paper, considering not only metrics like traffic information quality and performance in different scenarios but also possible privacy issues which may arise with certain techniques.

The novel ultrasound-based traffic sensing technique, which is the focus of this paper, belongs to the category of infrastructural sensing. A significant difference to existing ultrasonic sensor systems is the placement of the sensors in a so-called sidefire setup. This means that the sensors are mounted in a height of at least three meters and face the street sideways with downtilted sensors, while being positioned on the side of the street. Previously existing systems were mainly single-lane ultrasonic sensors measuring the height profile top-down, for example, from a special sensor gantry above the street, with a simple first-reflection processing. The monitoring of multiple lanes is possible with our setup, while, at the same time, the requirements in terms of algorithmic evaluation and processing are increased in comparison to these previously existing systems. However, with these improvements, our approach allows operation in a highly reflective acoustic environment, which is present in urban areas and permits simply mounting the sensor on the side of the street. This is enabled by a new approach with a combination of statistical signal processing, clustering, and inference algorithms for traffic participant object detection.

This sidefire ultrasonic traffic sensing technique is part of a development which focuses on intelligent infrastructure solutions, more specifically on intelligent street lights. An exemplary integrated setup is shown in Figure 1. These light modules can be used to replace inner components of street lamps, for example, by retrofitting the widespread type of high-pressure sodium-vapor lamps. Then, they can act as a flexible platform for many applications by employing the processing, communication, and light control actuation capabilities. With the transition to LED-based lighting, which is triggered by high energy costs and several EU directives [5], a large proportion of the existing street lamps will have to be replaced in the next decade, offering potential for an integration of Smart City solutions. The seamless integration into infrastructure allows an almost complete coverage of relevant urban traffic to provide the required coverage density of traffic information. Furthermore, the major problem of power supply for sensing devices is alleviated, as the lamps are connected to the mains power, which eliminates the need for energy-constrained and expensive battery-powered devices.

The rest of this article is organized as follows. Section 2 provides the related work of urban traffic sensing techniques in general and ultrasonic sensing in specific. Section 3 describes the system architecture together with the specific



FIGURE 1: Integrated street lamp head with a single ultrasonic sensor head.

scenario and typical sensor setup. Section 4 provides the algorithmic approach to the sensor signal processing system and object detection inference. In Section 5, the evaluation methodology and framework for reference data are presented. Section 6 gives results of real-world measurements and provides an evaluation from the perspective of detection and classification together with a state-of-the-art technology comparison. Section 7 then concludes the article.

2. Related Work

Ultrasonic traffic participant detection falls into the field of traffic sensing with a large number of techniques available in research and in practical application. In this section, we therefore provide a state-of-the-art overview from two perspectives. First, we give a comprehensive overview of traffic monitoring techniques in general with a discussion of their performance, strengths, and weaknesses in order to show the technological gaps and potential. In the second part, specific technical approaches in ultrasonic traffic sensing, which are rare both in concepts and in practical implementations, are given. This application-centric perspective shows the drawbacks and unresolved problems in current state of the art.

2.1. Traffic Monitoring Techniques from a Broad View. In the introductory section, two main categories of traffic information acquisition systems were introduced. These are on the one hand infrastructure-based sensing systems, working in a temporarily or permanently fixed location. On the other hand, the type of end-user assisted sensing systems incorporates distributed information acquired from the drivers, their technical devices, and the vehicle's internal sensors itself. In the following, a state-of-the-art overview together with a technological comparison is given for nowadays most relevant traffic monitoring techniques of both categories. A condensed overview of the properties is also given in Table 1.

The most widely used approach to traffic sensing is the use of infrastructure-based solutions, which represents a mainly centralized approach. In general, authority of the data is kept with the operator and, thus, several of the problems coming with user-centric techniques do not arise. However, the sensing systems have to be deployed over the whole city or at least at critical locations in terms of traffic. Achieving the required density for an extensive urban traffic monitoring is therefore paramount, which brings along high effort and cost as a centralized solution.

TABLE 1: Overview of state-of-the-art traffic monitoring techniques.

Sensing technology	Characteristics (\oplus/\ominus)	References
<i>Infrastructural sensing techniques</i>		
Infrared sensors (passive and active)	<ul style="list-style-type: none"> \oplus Vehicle class and speed information acquisition is possible \oplus Multilane operation possible (active sensors) \ominus Susceptible to bad weather conditions \ominus Regular cleaning required (active sensors) 	[10, 11]
Camera-based systems	<ul style="list-style-type: none"> \oplus High detection and classification accuracy \ominus Privacy and authorization issues \ominus Susceptible to bad weather and light conditions 	[6, 7]
In-pavement sensors (inductive, piezoelectric, and strain)	<ul style="list-style-type: none"> \oplus High detection accuracy with axle counting \ominus Invasive and costly installation due to in-pavement integration 	[8, 9]
Directional high-end radar sensors	<ul style="list-style-type: none"> \oplus Multilane multiobject traffic participant detection \oplus Directional and velocity information \ominus Cost and complexity of solutions 	[10, 12, 13]
Top-down ultrasonic and radar sensors	<ul style="list-style-type: none"> \oplus Low cost and simple mode of operation \ominus Only single-lane operation \ominus Mounting above lane required, for example, on sensor gantry 	[10, 14–18]
Proposed sidefire ultrasonic sensing	<ul style="list-style-type: none"> \oplus Multilane operation possible \oplus Sidefire mounting allows seamless integration into environment \ominus No directional information available in sidefire configuration 	
<i>User-based distributed traffic monitoring techniques</i>		
Crowdsourcing systems (smartphones and navigation devices)	<ul style="list-style-type: none"> \oplus High number of devices \ominus Privacy concerns \ominus Susceptibility to attacks and data manipulation 	[21–23]
V2X-based cooperative systems	<ul style="list-style-type: none"> \oplus Combination with in-vehicle information and decisions \ominus Low support and prevalence 	[24]

Camera-Based Systems. Frequently, camera-based systems are used for traffic monitoring. Here, we focus on systems with an automated analysis that is based on Computer Vision algorithms. With camera-based systems, a high detection accuracy plus a vehicle classification can be achieved. During nighttime, further solutions are required, for example, artificial illumination of the road and infrared or thermal imaging. Only in bad visibility conditions as fog, snow, or smog, the performance is limited [6]. The main concern with these systems, however, is the issue of privacy arising with video cameras in the public domain. On the one hand, the personal privacy of the citizens is affected; on the other hand, governmental regulations might even forbid the use of cameras in public areas, especially in this high-coverage case. Also, even with only a sparse coverage of video cameras in the public space, acceptance levels amongst citizens are low [7].

In-Pavement Sensors. Sensors in the urban area embedded into the pavement asphalt are often used in city intersection areas for traffic light control. The main in-pavement sensor technologies are inductive loops, piezoelectric sensors, and strain gauges [8]. They can also enable high-accuracy traffic measurements together with a classification by counting

the axles of vehicles [9]. As a downside, the installation of pavement-integrated sensors is highly invasive and costly and requires per-lane road work with a possible temporary disturbance of city traffic.

Infrared Sensors. Passive and active infrared sensors are nonintrusive sensor types which can be mounted on the side or directly above the street. They offer detection with additional speed and vehicle class information in some realizations. With active sensors, multilane operation is possible. However, infrared sensors are susceptible to bad weather conditions like rain, snow, or fog. Furthermore, active infrared sensors require regular cleaning which might lead to lane closure during maintenance [10, 11].

Radar Systems. From the basic object detection principle, radar sensors represent the technology most comparable to ultrasonic sensors in the field of traffic sensing. High pulse repetition rates and ranges together with the possibility of obtaining directional information together with velocity data allow the monitoring of multiple traffic lanes and objects independent of the weather conditions [10, 12, 13]. As a major downside, the significant cost of the radar system, including front-end and signal processing, can compromise

a city-wide system deployment for a sufficient coverage with these sensors.

Top-Down Ultrasonic and Radar. In the field of radar and ultrasonic sensors, several top-down sensor solutions are available, which require the placement above a specific street lane for single-lane car profile measurements [10, 14–18]. These techniques allow obtaining the complete height profile of vehicles for detection and classification but require a gantry or bridge over the street for mounting the sensors, which is infeasible in many situations. Moreover, only a single lane can be monitored with the existing solutions [10]. For the ultrasonic sensors, the processing is strongly simplified, as with the available systems, only the distance to the first relevant reflection and not the complete impulse response is evaluated. The significantly lower propagation speed and resulting decay time of the impulse responses together with the requirements for a sufficient pulse repetition rate (typically ≥ 10 Hz) of ultrasonic sensors can lead to self-interference and also to issues with fast-moving traffic participants [18]. For ultrasonic sensors, previous studies have shown that the impact of weather effects such as wind, snow, and rain is only marginal [19, 20].

Crowdsourcing Systems. For the second category of end-user based distributed information systems for traffic monitoring, one of the most present techniques is the crowdsourcing of traffic monitoring to end-user smartphones and navigation devices in order to obtain real-time and high-density traffic flow information [21, 22]. This approach is used in several systems and solutions, for example, by “Google Maps” with the product “Google Traffic” or by “TomTom HD Traffic.” As a central advantage, no investments in additional sensors are required and the very high prevalence of these applications on end-user devices can be exploited by actively tracking the devices. However, the live tracking of the users is highly critical in terms of privacy aspects. Furthermore, these systems are susceptible to manipulative techniques and fake information due to trust issues with the user base, which has been demonstrated before [23].

V2X Communications. With the emerging field of V2X (vehicle-to-everything) communications, which is currently focused on V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) techniques, new possibilities for traffic information acquisition and exchange arise. Together with in-vehicle information for a direct delivery of inferred routing decisions to the driver and in a fusion with available travel plans on the car navigation device, an optimized information exchange and routing is enabled. Nevertheless, these types of systems are still to be realized in mass market together with suitable application scenarios and it will take time until they are sufficiently established amongst everyday vehicles. Moreover, reservations with regard to privacy and trust are also an important aspect for these systems. In literature, it is shown that, with a sufficient prevalence of cooperative V2V techniques in cars together with a high relaying communication range, dense traffic information can be obtained [24]. Therefore, vehicular communication

techniques are a promising approach for the future. Certainly, as long as the density and proportion of vehicles supporting V2X communications is too low, a diverse traffic monitoring approach, which in parallel comprises infrastructural sensing techniques, has to be pursued for several decades.

2.2. Specific Advances in Ultrasonic Traffic Detection. In the previous introductory Section 2.1, the basic top-down ultrasonic sensing technology has already been introduced. However, the practical use is limited, as additional mounting efforts are required, if no infrastructure like bridges above the street or traffic light poles at intersections are available. Moreover, only single-lane detection is possible in this regard. Therefore, these solutions are limited in their practical applicability.

There also exist initial concepts that use ultrasonic sensors mounted on a lateral street position with a so-called sidefire orientation, facing the opposite street side either horizontally or in a downtilted position [19, 25, 26]. In the case of horizontal tilt, even a first simple multilane detection has been realized. However, all these concepts perform only an analysis of the first-reflection distance instead of the whole impulse response, which highly limits the practical use in a variety of scenarios. The urban scenario is a highly reflective time-varying environment, where first reflections of other objects such as moving trees, parking cars, and the scattering of the road surface are always present in the distance range with the relevant traffic participants for detection, the distance region-of-interest for measured signal reflections of objects. In addition, the horizontally facing type of sidefire sensors could be obstructed by other objects and traffic participants in urban scenarios due to their low mounting height.

3. System Concept and Architecture

Sensor systems have specific requirements regarding the setup, scenario, and working environment they are being operated in. In this chapter, the system setup and application concept, together with the generic requirements for our proposed ultrasonic traffic sensing system, are given. The scenario stays the same as initially proposed in Section 1: a Smart City street lamp system platform. Nevertheless, a wide variety of installation and application possibilities are given also without any ties to this integration concept.

3.1. Sensor System Setup. An exemplary setup of the smart street lamp platform was already shown with Figure 1. It features a single mounted ultrasonic sensor attached to the lamp head or the pole and an embedded system integrated into a customized LED-based lamp head. The digital core system contains modules for realizing processing and communication capabilities, enabling distributed processing in a local-level group of lamps in a street by building a dynamic mesh network and also allowing communication on the global and cloud level by employing mobile network machine-to-machine type communication (M2M) functionalities. The central component for the signal processing capabilities consists of a Texas Instruments Sitara system-on-chip (SoC) including an ARM Cortex-A8 core. Typical examples of

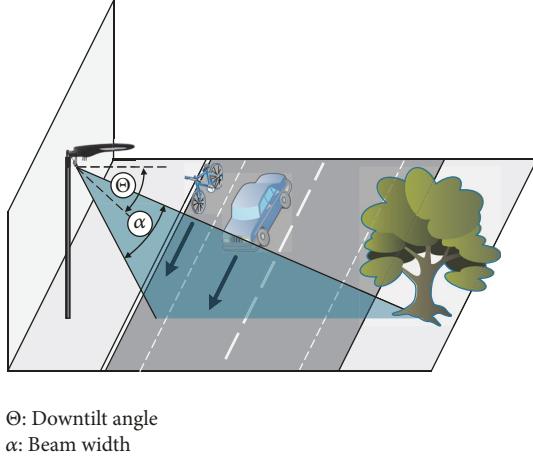


FIGURE 2: Downtilted sidefire sensor configuration in typical urban environment with two-lane street (with downtilt angle Θ and beam width α).

possible applications with this system are location-based content delivery, attachment of modules for environmental or traffic sensing applications, and energy efficiency optimizations of street lighting. Due to the combination of infrastructure with an extensible platform, the systems can be widely distributed amongst the city, as almost all central areas are covered by street lamps. Furthermore, many of the constraints on processing and communications, which are typically imposed for Smart City traffic monitoring systems in IoT context, can be relaxed, for example, data and refresh rate restrictions due to battery power [5].

The typical seamless integration of the ultrasonic sensing concept into an urban street area is shown in Figure 2. In the so-called downtilted sidefire setup, the sensor is mounted in a height between 3 and 8 m, which is above the typical height of vehicles in urban areas. The sensor head is then oriented diagonally downwards the street, with a downtilt angle Θ between 30° and 80° . The ultrasonic sensor transducer together with the casing has a cone-shaped beam characteristic, with an opening angle α typically between 20° and 80° depending on the scenario and lane coverage requirements. In the exemplary scenario shown in Figure 2, the multitude of reflective elements in the urban environment already becomes apparent. For example, the first reflections and scattering of the street surface can arrive earlier than reflections from relevant objects on distant lanes to the sensor. Additionally, reflective objects can exhibit short- or long-term time variation, like parked cars or trees moving in the wind. In these cases, simple distance-measurement based sensor techniques would already fail.

3.2. Ultrasonic Sensor Platform Module. In order to provide the capabilities for ultrasonic sensing, the core system is extended with a custom sensor platform module, which is shown in Figures 3 and 4. It provides four ultrasonic sensor channels for real-time arbitrary waveform generation and recording in half-duplex mode. The reception quality of the ultrasonic impulse response is with an input stage SNR of

above 105 dB in the ultrasonic band. The analog driving circuit for the transducer is integrated into the ultrasonic transducer head itself and is adapted to the capabilities and characteristics of the transducer.

Control of the attached components, signal generation, and preprocessing capabilities is realized with an integrated Xilinx Artix-7 series FPGA. It provides communication capabilities to the core processing system over an Ethernet link, which is also used for the power supply of the sensor system extension. Together with the core system, GPS including the high-precision PPS signal is used for a timing synchronization and exact coordination of multiple sensors in the vicinity of a street. Interference can therefore be coordinated by a global optimization of the sensor timing patterns if the sensor heads are transmitting in the same frequency band.

3.3. General Signal Structure. For simplicity, the signal structure in the single-sensor case is discussed. The transmit signal $s_T(t)$ consists of repeated sequences of the base pulse $h_i(t)$ with an overall duration T_{rep} , yielding a pulse repetition rate of $f_{\text{rep}} = T_{\text{rep}}^{-1}$. This is also shown in Figure 5. T_{rep} is typically chosen between 50 and 150 ms to ensure low self-interference levels between the different impulse response blocks. The repetition time should in general be larger than the RT60 reflections decay time of the acoustic channel [27, p. 98]. The RT60 time is a common measure in indoor acoustics but can also be applied in outdoor scenarios, where it typically yields much shorter values. The general pulse sequence is then defined as follows:

$$s_T(t) = \sum_{n=0}^{\infty} h_n(t - n \cdot T_{\text{rep}}). \quad (1)$$

The base pulses $h_i(t)$ itself consist of two specific duplexer phases. At the beginning, the transmit duplex mode is used and a transmit pulse part of length T_{wnd} is emitted. Then, the duplex mode is switched to receive mode and records the channel impulse response for the rest of the period $T_{\text{rep}} - T_{\text{wnd}}$ without further transmission. For the investigations in this article, the transmit pulse is chosen to be a windowed signal with a carrier frequency of $f_c = 40$ kHz and an envelope window $h_{\text{wnd},i}(t)$ with length $T_{\text{wnd}} = 2$ ms as a tradeoff between time and frequency resolution for the narrowband signal. The i -th base pulse is then given by the following:

$$h_i(t) = \begin{cases} h_{\text{wnd},i}(t) \cdot \text{Re}\{e^{j2\pi f_c t}\} & \text{if } 0 \leq t \leq T_{\text{wnd}}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

With the use of identical repeated transmit pulses in our case, $h_i(t) = h(t) \forall i \in \mathbb{N}$. A Blackman window [28] function is chosen for $h_{\text{wnd},i}(t)$ because it does not exhibit discontinuities at the beginning and end of the time domain window, while exhibiting a high stop-band attenuation [29].

4. Processing Concept and Algorithms

The ultrasonic traffic sensing technique presented in this paper is based on a combination of statistical analysis

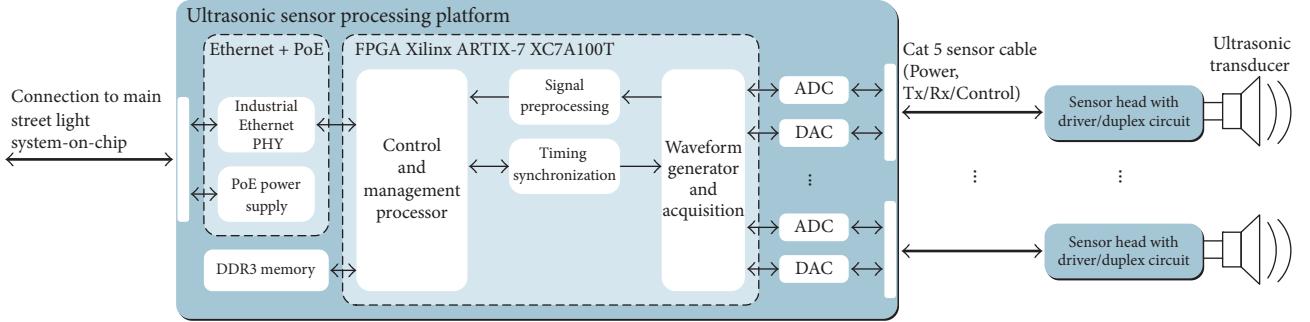


FIGURE 3: General structure of sensor platform including the FPGA and attached ultrasonic sensors with driving circuits.



FIGURE 4: Custom sensor platform which is integrated as an extension module into the intelligent street light.

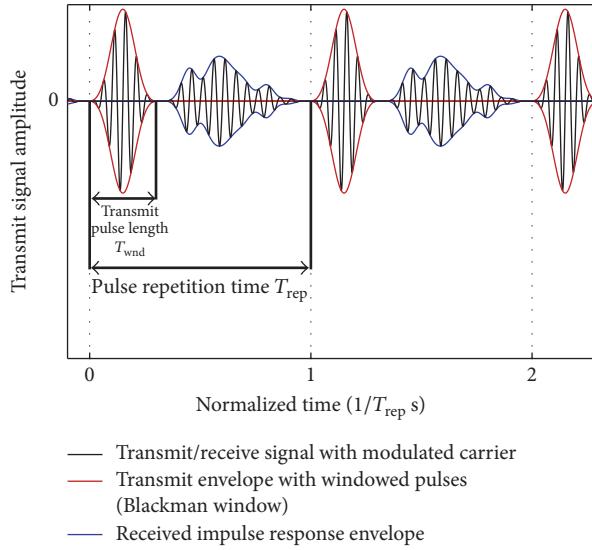


FIGURE 5: Generalized signal structure with transmit/receive duplexing mode, corresponding envelopes, and modulated signals.

and clustering techniques for object candidate detection. Together with a special signal chain structure, objects representing traffic participants can be detected, together with an extraction of further characteristics and object properties. In this section, the model for the received signal is given first. Then, the characteristics of the reflective environment are modeled and empirically analyzed. Based on these findings, statistical hypothesis testing is performed during operation and the resulting statistical information is combined with a modified clustering algorithm for object boundary detection.

Then, the system concept for the extraction of further characteristics is given and the reduced-complexity embedded implementation for real-time operation is discussed.

4.1. Received Signal Model and Preprocessing. With the general signal structure described in Section 3.3, repeated measurements of the acoustic channel are performed with the ultrasonic sensor system. Based on the impulse responses as the one-dimensional received signals, which consist of a large number of reflections in the acoustic environment, object detection and further analyses can be performed together with the time-of-flight distance information.

The received signal $s(t)$ from the sensor head is recorded continuously and then sliced into blocks of length T_{rep} , yielding a series of impulse responses $s_i(t_D)$, with i being the i th measurement interval of length T_{rep} . The parameter t_D is the time position of the specific impulse response, which is mapped to an equivalent time-of-flight distance $d = t_D \cdot c$ given the speed of sound $c = 343 \text{ m/s}$ in the air medium at normal conditions. Due to the round-trip, the real distance to the reflective object is half the equivalent time-of-flight distance. By acquiring information on the current air temperature via sensor measurements or weather information, c can be compensated for the current conditions. During the initial period of $s_i(t)$ with $0 \leq t_D \leq T_{\text{wnd}}$, which is the transmission duplex mode, no useful signal is acquired, while the region $T_{\text{wnd}} \leq t_D \leq T_{\text{rep}}$ represents the main signal of interest for further analysis. The single impulse response slices are then given by the following:

$$s_i(t_D) = s(i \cdot T_{\text{rep}} + t_D). \quad (3)$$

The equivalent time-discrete representation is

$$x_i(n_D) = s(i \cdot N_{\text{rep}} T + n_D \cdot T). \quad (4)$$

with the sampling time T and the discrete-time repetition interval length $N_{\text{rep}} = T_{\text{rep}}/T$.

As a first step of preprocessing of the sliced receive signal on the FPGA, each impulse response is convolved with a bandpass filter $h_{\text{BP}}(t)$ centered at the transmit carrier frequency f_C with a bandwidth of 6 kHz for allowing the analysis of Doppler-shifted signals with typical urban vehicle velocities. Then, a time-discrete Hilbert transform \mathcal{H} [30, 31]

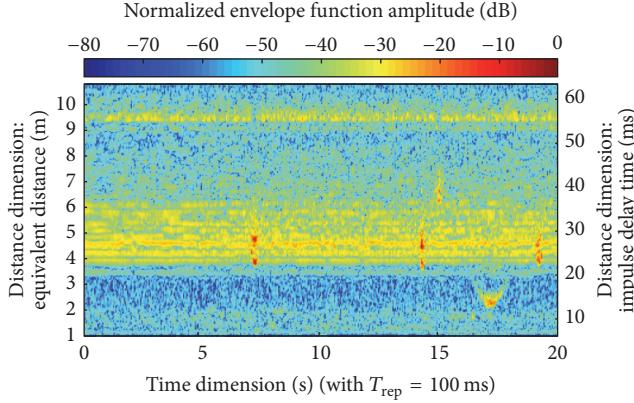


FIGURE 6: Two-dimensional envelope function mapping of received signal $e_{R,i}(n_D)$ (amplitude normalized and in logarithmic scale).

is performed in order to acquire the complex-valued analytic signal:

$$x_i(n_D) = (s_i * h_{BP})(n_D) + j \cdot \mathcal{H}\{(s_i * h_{BP})(n_D)\} \quad (5)$$

together with the instantaneous amplitude or real-valued envelope signal

$$e_i''(n_D) = |x_i(n_D)|. \quad (6)$$

This envelope signal $e_i''(n_D)$ is further processed to facilitate object detection in the following. Only in case of Doppler-based motion and velocity estimation, time-frequency information such as the instantaneous phase and frequency has to be incorporated. Both its two discrete dimensions, i being the index of the impulse response (from now on called time dimension, with an equivalent sampling rate of f_{rep}) and n_D being the distance-equivalent impulse response time (from now on called distance dimension, with an equivalent sampling rate $f_s = 1/T$), are derived from a single original time dimension in the received signal by slicing. Therefore, the distance dimension is fixed and limited to $0 \leq n_D \leq N_{rep} - 1$, while the causal time dimension in real-world application is extending with f_{rep} . With this two-dimensional mapping, object detection algorithms become applicable which consider a vicinity in both dimensions. This means that effects of an object creating a peak at several neighboring distance and time points are now properly represented as neighbors in the 2D space, in contrast to the original one-dimensional recorded sensor signal.

An example for the envelope signal $e_{R,i}''(n_D)$ is given in Figure 6, with a normalized logarithmic color mapping of the amplitude for improved visualization. The equivalent distance and impulse response time mappings are shown on the ordinate for reference purposes. In this plot, the scattering and reflections from the street surface can already be seen in a distance range of 4 to 6 m in a comb-like pattern. Also, around the time positions at 7 s, 14 s, 15 s, 17 s, and 19 s, strong reflections from objects passing the sensor are visible.

4.2. Signal Statistics and Standardization. Given the high complexity of the reflective patterns in the sensor data, the

statistics of the signal need to be described properly in order to perform an outlier or anomaly detection, which would indicate the presence of objects in the sensor field. The goal is to incorporate all noise components (measurement noise and other acoustic emissions in the ultrasonic band) and also the typical static and time-varying reflections (e.g., moving leaves of a tree caused by wind and reflection patterns of the street) for every specific distance point of the signal into the statistics. This allows the classification of segments of newly acquired impulse responses in terms of the data following the a priori distribution, called base distribution, which was estimated in the past, or whether an outlier is present.

In contrast to typical binary decision problems, here, only this base distribution without any presence of an object can be estimated, representing the null hypothesis. The alternative hypothesis of object presence is however different for every object. This problem can be solved using parametric or nonparametric hypothesis testing techniques for properly representing the underlying base distributions of the signal points and testing the outlier probabilities. However, for the real-time implementation on the embedded system of the sensor setup, we present a simpler and less computationally complex approach that is based on the distance-wise standardization of the signal based on the calculated statistics in a predefined time window in the past. These standardization techniques are often used in the field of data science as a preprocessing step for feature scaling. The distance-wise standardized envelope signal $e_i'(n_D)$ is calculated based on the information from the statistics memory with a window length of $N_{T,w}$ time dimension steps in the past (equivalent to a time range of $N_{T,w} \cdot T_{rep}$ in continuous time), yielding $e_i'(n_D)$:

$$e_i'(n_D) = \frac{e_i''(n_D) - \mu_{n_D}(i)}{\sigma_{n_D}(i)} \quad (7)$$

$$= \frac{e_i''(n_D) - \overline{N_{T,w}^{-1} \sum_{k=i-N_{T,w}}^i e_k''(n_D)}}{\sqrt{(N_{T,w} - 1)^{-1} \sum_{m=i-N_{T,w}}^i [e_m''(n_D) - N_{T,w}^{-1} \sum_{k=i-N_{T,w}}^i e_k''(n_D)]^2}}, \quad (8)$$

Windowed mean $\mu_{n_D}(i)$ for distance n_D

Windowed standard deviation $\sigma_{n_D}(i)$ for distance n_D

$$0 \leq n_D \leq N_{rep} - 1, i \geq N_{T,w}.$$

Under the simplified assumption that the underlying process of $e_i''(n_D)$ (with no objects present) for a given distance n_D is an i.i.d. Gaussian process $X_{n_D} \sim \mathcal{N}(\mu_{n_D}, \sigma_{n_D}^2)$, the specific distance ranges of $e_i'(n_D)$ will follow the standard normal distribution $Y_{n_D} \sim \mathcal{N}(0, 1)$. The assumption is especially found to be valid for direct reflections in the closer range, shown in studies from other acoustic environments like underwater [32]. As a general approximation, it will later be used for approximatively calculating the initial detection thresholds for a given false alarm rate.

For the object detection, only the right tail of the distribution is of interest for outlier detection, as shadowing effects are not considered. Therefore, the complete standardized distribution data $e_i'(n_D)$ is clipped for values below zero,

resulting in the final envelope $e_i(n_D)$ with statistical standardization:

$$e_i(n_D) = \max(e'_i(n_D), 0). \quad (9)$$

This yields a variable with a standard normal distribution left-censored at 0. The resulting probability distribution function (PDF) $f(x)$ can be written as follows:

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} + \frac{1}{2} \delta(x) & x \geq 0, \\ 0 & \text{otherwise;} \end{cases} \quad (10)$$

and the cumulative distribution function (CDF) $F(x)$ is

$$F(x) = \begin{cases} \frac{1}{2} \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + \frac{1}{2} & x \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where $\operatorname{erf}(x)$ is the Gauss error function. The first terms of $f(x)$ and $F(x)$ are similar to a scaled half-normal distribution [33], with the addition of the clipped values being statistically censored and not truncated.

To summarize, $e_i(n_D)$ is now used for all further object detection and processing together with the clustering techniques in the following. Simply put, it yields the positive outlier level of a newly acquired data point by the distance to the calculated mean in a measure of the multiple of standard deviations. This is based on the statistical history in a specific time interval and calculated separately for each discrete distance point. By having a limited time window for the statistics, the sensor system can adapt to changing environments without needs for recalibration. Optionally, for achieving a better robustness and generalization, these distance-wise statistics can be blurred with neighboring distance points.

4.3. Object Boundary Detection with Density-Based Statistical Clustering (DBStaC). As a next step, the standardized data given in the previous section is used for object detection. The two-dimensional data $e_i(n_D)$ is passed through a modified density-based clustering algorithm based on the original DBSCAN (Density-Based Spatial Clustering of Applications with Noise [34]), which is able to perform clustering on noisy data by aggregating core points of data peaks. In our case, these data peaks can be the result of statistical outliers due to physical objects passing by the sensor, in comparison with the base distribution of the signal when no objects are present. Our proposed modified algorithm in combination with the input preprocessing steps from (7) and (9), called *density-based statistical clustering* (DBStaC), can be used on both statistical hypothesis test results, and in our case of a simplified algorithm, data standardized based on the a priori statistics.

4.3.1. Choice of Clustering Algorithm. The choice of a suitable clustering algorithm from the field of unsupervised learning techniques for our application was subject to several requirements. While many clustering algorithms can aggregate nearby points in a space with arbitrary dimensionality

and sparsely distributed data points, it is required here to incorporate the weight of data points, coming from our the nonsparse standardized and clipped data field $e_i(n_D)$ described before in Section 4.2.

Furthermore, the algorithm should be able to detect important data peaks based on a local density, while, for the whole cluster, no penalty for an infinite extension in the time dimension should be given. This is necessary because the dwell time of an object in the sensor range is arbitrary, especially due to different movement speeds of traffic participants or even in a traffic jam situation.

As a third requirement, an anisotropy of the cluster properties and shape is desired, as we have two dimensions that are time (index of different impulse responses) and distance (specific point in the impulse response itself), which have highly different characteristics. A typical scenario where anisotropy would not be required is 2D image processing, where both image dimensions have similar properties.

4.3.2. Original Definition of DBSCAN. First, we want to provide a basic understanding of the original DBSCAN algorithm in general and the specific terms coming with the algorithmic definition. Then, we can introduce the modifications for our specific application. Readers interested in the complete formal description are referred to [34, 35], on which the following original definitions are based.

Core Point. The general principle of DBSCAN is the analysis of a set D of points in a k -dimensional space. Then, for any given point $\mathbf{p} \in D$, the ε -neighborhood of this point is evaluated, meaning that all points $\mathbf{q} \in D$ with a distance $\operatorname{dist}(\mathbf{p}, \mathbf{q}) \leq \varepsilon$ belong to the neighborhood of \mathbf{p} , denoted \bar{N} . The distance metric $\operatorname{dist}(\mathbf{p}, \mathbf{q})$ is typically chosen to be the L_1 or L_2 norm, and ε is preset. This results in the score \bar{n}_ε of a point \mathbf{p} being the number of other points \mathbf{q} falling into the ε -neighborhood. Then, the point \mathbf{p} is called core point if $\bar{n}_\varepsilon \geq \text{MinPts}$, with MinPts being a preset threshold for core point detection. All points in the ε -neighborhood of a core point are part of a cluster set C , and they are called border points if they are not core points themselves.

Direct Density-Reachability. A point \mathbf{p} is directly density-reachable from a point \mathbf{q} if \mathbf{q} is a core point and \mathbf{p} is in the ε -neighborhood of \mathbf{q} . This property is symmetric if \mathbf{p} and \mathbf{q} are a pair of core points [34].

Density-Reachability. A point \mathbf{p} is density-reachable from a point \mathbf{q} if a chain of n points $\mathbf{p}_1, \dots, \mathbf{p}_n$ with $\mathbf{p}_1 = \mathbf{q}$, $\mathbf{p}_n = \mathbf{p}$ exists where every point \mathbf{p}_{i+1} is directly density-reachable from \mathbf{p}_i , $\forall i \in [1 \dots n - 1]$ [34].

Density-Connectivity. A point \mathbf{p} is density-connected to another point \mathbf{q} if there is a point \mathbf{o} such that both \mathbf{p} and \mathbf{q} are density-reachable from \mathbf{o} [34].

Cluster Definition. Given the set of all points D , a cluster C is a nonempty subset of D which satisfies the following conditions [34]:

- (1) $\forall \mathbf{p}, \mathbf{q}$: if $\mathbf{p} \in C$ and \mathbf{q} is density-reachable from \mathbf{p} , then $\mathbf{q} \in C$ (maximality).
- (2) $\forall \mathbf{p}, \mathbf{q} \in C$: \mathbf{p} is density-connected to \mathbf{q} (connectivity).

Any cluster C is then uniquely determined by its core points.

4.3.3. Modified Core Point Definition for DBStaC. In our special case, the algorithm is modified to suit the needs for clustering our two-dimensional space; therefore $k = 2$ in order to represent the time and distance dimension of our acquired data. Furthermore, our data points are weighted by their standardized value. Therefore, for a point $\mathbf{p} = (i, n_D)$, the weight $e_i(n_D)$ is assigned, which is not an option in the classic DBSCAN algorithm. Instead of using the dist metric for determining the ε -neighborhood, anisotropy is introduced by choosing the ε -neighborhood as a rectangular area with dimension $(2\varepsilon_T + 1) \times (2\varepsilon_D + 1)$, symmetrically placed around the position (i, n_D) of the core point, with ε_T and ε_D given. Then, all points $\mathbf{q}(i', n'_D)$ with $|i' - i| \leq \varepsilon_T \wedge |n'_D - n_D| \leq \varepsilon_D$ belong to the ε -neighborhood N of $\mathbf{p} = (i, n_D)$ and \mathbf{p} is a core point if

$$\sum_{t=i-\varepsilon_T}^{i+\varepsilon_T} \sum_{d=n_D-\varepsilon_D}^{n_D+\varepsilon_D} e_t(d) \geq \text{MinSum}. \quad (12)$$

With this definition, all data points of our two-dimensional data set are now taken into account, not just by their count, but by the sum of their assigned weights in the specific ε -neighborhood. The implementation of the final clustering is omitted at this point and widely described in literature [36]. With the original algorithm implemented, adaptions only have to be made in the core point definition and neighborhood metrics.

4.3.4. False Alarm Rate. In the clustering process, every element of $e_i(n_D)$ is tested for the core point property, requiring both compensation for multiple comparisons and an approximation of the false alarm rate for a single point itself. For the latter, we will provide measures for determining the false alarm rate under the approximative distribution assumptions from Section 4.2. With the threshold decision from (12) for a core point, given MinSum and the ε -neighborhood with

$$A = (2\varepsilon_T + 1)(2\varepsilon_D + 1) \quad (13)$$

points in the rectangular region, we want to calculate the false alarm probability:

$$P_f = P(S(i, n_D) \geq \text{MinSum} \mid \bar{O}), \quad (14)$$

where \bar{O} is the event that no relevant object reflection yielding a core point is present, meaning that all summands follow the base distribution without outliers. $S(i, n_D)$ is the sum of weights in the ε -neighborhood:

$$S(i, n_D) = \sum_{t=i-\varepsilon_T}^{i+\varepsilon_T} \sum_{d=n_D-\varepsilon_D}^{n_D+\varepsilon_D} e_t(d). \quad (15)$$

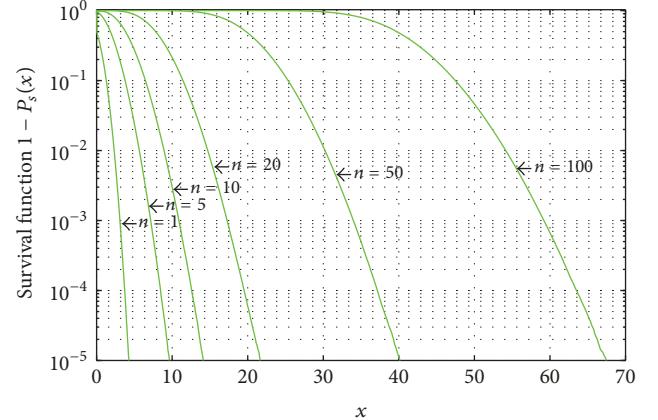


FIGURE 7: Survival function $1 - P_s(x)$ for distribution $p_s(x)$ with n -fold convolution of $f(x)$.

In Section 4.2, we showed that under the assumption that the original envelope signal $e_i''(n_D)$ is coming from a process with normal distribution, $e_i'(n_D)$ is standard-normally distributed after standardization, and $e_i(n_D)$ follows a left-censored standard normal distribution. If the data points $e_i(n_D) \forall i, n_D$ and the resulting $S(i, n_D)$ are also hypothesized to be statistically independent and exhibit the identical PDF $f(x)$ from (10), we can write for the resulting PDF $p_S(x)$ of the summation of $n = A$ points in (15):

$$p_S(x) = \underbrace{(f * \dots * f)}_{n \text{ times}}(x) = f^{*n}(x). \quad (16)$$

Please note that the statistical independence of all $S(i, n_D)$ is not given in the first place because the same data points in the two-dimensional field are affected by the sliding window in several i and n_D positions. Then, adjacent $S(i, n_D)$ are correlated and a single peak in $e_i(n_D)$ could yield multiple peaks in $S(i, n_D)$ resulting in core points. However, as we seek to calculate the false alarm rate in general for this specific algorithm, adjacent core points would be clustered into a single cluster, raising only false alarm for a single object if $P_f \ll 1$.

Finally, neither for the left-censored standard normal distribution $f(x)$ nor for the related half-normal distribution in general, the result of k -fold convolution is known or given in literature in closed form for $k > 2$. Therefore, the final calculation is based on Monte Carlo simulations of these random processes. For the false alarm probability analysis, we are interested in the survival function (SF) $1 - P_s(x)$, where $P_s(x)$ is the CDF of $p_s(x)$. With $x = \text{MinSum}$, the survival function yields the false alarm probability per data point. The results from the Monte Carlo simulations for the survival function are shown in Figure 7 and were also verified with an additional false alarm Monte Carlo simulation.

4.3.5. Choice of Clustering Neighborhood. The neighborhood choice in the time direction, ε_T , affects the detection robustness and separability of vehicles with length l_{Veh} following each other in a gap distance l_{Gap} , with speed v and passing the sensor lobe which approximately covers a section of length

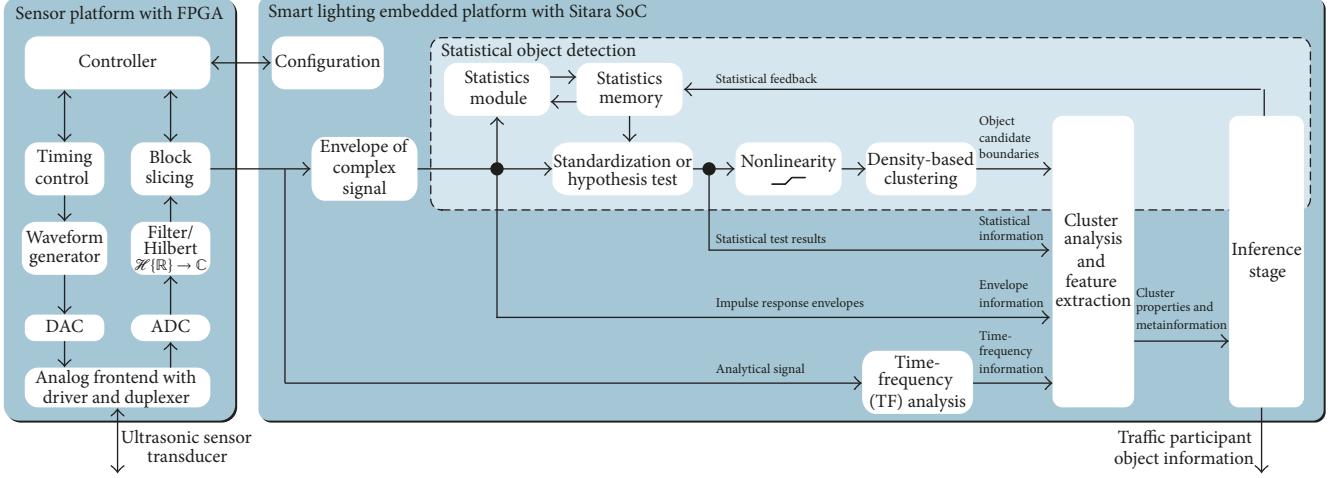


FIGURE 8: Signal chain with acquisition, analysis, and inference stages on both platforms.

l_{Lobe} on the specific lane. Given the pulse repetition rate T_{rep} , no vehicle is covered by the sensors during the gap for n_{Gap} pulse times:

$$n_{\text{Gap}} = \frac{(l_{\text{Gap}} - l_{\text{Lobe}})}{(\nu \cdot T_{\text{rep}})}. \quad (17)$$

Also, considering the detection of a single vehicle, the vehicle is sampled in the lobe for n_{Veh} pulses:

$$n_{\text{Veh}} = \frac{(l_{\text{Veh}} + l_{\text{Lobe}})}{(\nu \cdot T_{\text{rep}})}. \quad (18)$$

Given ε_T , the full neighborhood extension in time direction is $2\varepsilon_T + 1$ pulses. Therefore, $n_{\text{Gap}} \geq 2\varepsilon_T + 1$ is recommended for a good separation of the vehicle clusters in time direction in order to have no overlapping core points. Typical values of the time neighborhood are $0 \leq \varepsilon_T \leq 2$. With an urban area example of $\varepsilon_T = 1$, $l_{\text{Veh}} = 4.5$ m, $\nu = 40$ km/h, and $l_{\text{Lobe}} = 1.5$ m, the recommended gap length becomes $l_{\text{Gap}} \geq 4.83$ m and a vehicle is covered by $n_{\text{Veh}} = 5.4$ pulse times. The second neighborhood parameter ε_D in distance direction is mostly relevant when it comes to multilane separation of multiple vehicles. In general, our approach in this paper is to learn the typical vehicle speed and distance profile without a direct calculation, but by learning the scenario for a short training period (e.g., one hour) and then setting the neighborhood parameters in a parameter optimization, which is described in the following section.

4.4. Signal Processing Concept. In Figure 8, the previously described statistical object detection is integrated into the complete object detection concept with feature extraction and inference. It allows detecting candidates for detected objects representing traffic participants, which are then used to gather further information from several signal processing paths. The preprocessing and main processing stages are also divided in hardware into the sensor platform FPGA and the Sitara SoC, respectively.

4.4.1. Preprocessing. In contrast to the sensor platform structure described previously in Section 3, here, the signal flow and processing are in focus. As can be seen in the left part of Figure 8, the sensor platform is configured by the main system on the Sitara SoC and performs the whole signal transmission and reception. The preprocessing steps of filtering, Hilbert transformation, and possible sample rate adjustments are performed using a dedicated architecture on the FPGA, before slicing the blocks from the different sensors into single impulse responses, which are then transmitted to the main processing system with the Sitara SoC.

4.4.2. Object Detection and Extraction. On the right side of Figure 8, the complete object detection and feature extraction concept is shown. It is based on four main information paths going into the cluster analysis and feature extraction block at the end, whose output is then used for inference of detected traffic participants. By performing a fusion of these four paths, all necessary information for the analysis can be provided. The different components of the processing system are described in the following.

Statistical Object Detection with Resulting Object Candidate Boundaries. The basic principle of object clustering on the standardized data for boundary detection has already been described before in Sections 4.2 and 4.3. The only addition is an optional nonlinearity after standardization or the optional hypothesis testing (which is not considered here). The nonlinearity can be modified to perform further transformations on the standardized data. With the DBStaC algorithm, the envelope signal at the beginning of the statistical object detection block is analyzed by the statistics module together with its statistics memory. Only a predetermined amount of past data can be stored in this fixed-size memory for calculating the base statistics. The boundaries of resulting clusters of the algorithm are taken as boundaries for further analyses in the time-distance-field of the different data paths. Inside each cluster's area which was determined by the object boundary information path, the cluster analysis stage

performs an analysis of the statistical, envelope, and time-frequency-information path.

Statistical Feedback. The statistical feedback path coming from the inference stage is important to keep the base distribution in the statistics memory accurate to be mostly restricted to data without objects present. This is an optional feature and especially helpful with high fluctuations of traffic density in front of the sensor, as the detected objects are removed from the statistics memory and the outlier analysis of the DBStaC-based object detection is more accurate and robust against drift.

Statistical Information and Envelope Information. The standardized data and the original envelope signal are directly fed into the cluster analysis stage. For feature extraction and accurate information on the object reflection characteristics, properties like the accurate position of the object's first reflection and the mathematical two-dimensional center of mass of the signal or the geometry are analyzed. Especially with a priori geometric information on the system setup in a specific scenario, effects caused by the beam width of the transducer can be exploited. Even if a car moves perfectly sideways through the cone-shaped beam of the sensor, it forms a typical arch-shaped pattern, which could already be seen in the initial example signal, Figure 6. This arc-shaped reflection pattern is also known from marine sonar systems as "fish arc" and caused by the fact that at the point in time when a vehicle first enters the beam, the reflection path is diagonal and is not fully straight until the car is directly in front of the sensor. This can be used to support the information on vehicle size and type.

Time-Frequency Information. The time-frequency information is mostly irrelevant as long as the sensor is oriented perpendicular to the vehicle movement on the street. However, in case that the sensor is oriented partially sideways or a second sensor with such orientation is available, velocity information can be gathered by analyzing the instantaneous frequency of the signal for Doppler shift estimation.

Inference Stage. The inference stage performs the final decision whether the cluster detected in the DBStaC stage originated from a traffic participant in the sensor range and allows the estimation of further object parameters and a simple classification of the vehicle type using a supervised learning stage with a Support Vector Machine. As our analyses in this article are focused on the DBStaC object candidate detection algorithm itself, no further rejection of objects is performed by the inference stage in this case. Only objects detected outside of the specific lanes, for example, on the sidewalk, are rejected.

4.5. Implementation Details. The main parts of the processing chain shown on the right side of Figure 8 were implemented in C++, including optimized versions for embedded system operations, while parts of the postprocessing and learning techniques as part of the cluster analysis and inference stages were implemented in Python. The core processing module

can be used for both algorithmic and performance analysis purposes on general Linux/BSD systems and the embedded Linux system running on the ARM Cortex A8 as part of the Sitara SoC. For algorithmic evaluation and parameter optimization in the next section, capabilities of the core module will also be used as part of an evaluation system. The numerical precision requirements for embedded system operation were relaxed from 64-bit floating point to 32-bit floating point, largely without compromising performance, in order to exploit the NEON floating point accelerator of the Sitara SoC. For the processing chain with the DBStaC algorithm, real-time operation is then possible for two attached ultrasonic sensors. This is the case for typical parameter sets, as the algorithmic runtime of DBSCAN is not fixed and is largely dependent on the choice of parameters (size of ϵ -neighborhood and decision thresholds) and traffic situation yielding different cluster sizes. Edge cases like extremely large clusters, for example, due to a traffic jam situation with very long dwell times of an object in the sensor range, have to be taken care of in the practical implementations.

For the sensor platform with the FPGA shown on the left side of Figure 8, dedicated accelerators for filtering and processing are used together with a MicroBlaze soft core for the coordination of waveform transmission and data acquisition. The configuration is controlled by the main processing system side with the Sitara SoC. A timing reference signal is acquired using GPS together with the high-precision PPS reference, which allows global synchronization of multiple sensors attached to multiple platforms, with an optimization of the transmission and interference patterns. Exchange of resulting information, for example, for sensor fusion and distributed processing for trajectory calculation of objects passing by multiple systems, is possible on both the level of a local wireless meshing between the systems and a global communication using cellular networks.

5. Evaluation Scenarios and Methodology

Evaluation of the complete system concept for traffic participant detection requires analyses both in real-world scenarios and in isolated conditions with synthetic scenarios, such as on automotive test tracks. A variety of European-wide test measurements have been performed with the system, described in Section 5.2, which will be analyzed and discussed in Section 6. For the reference data acquisition, a video camera is running in parallel which is afterwards used to label, annotate, and classify traffic participants on the time line. These labels can then be utilized to evaluate the performance in terms of several metrics for quantifying detection performance, parameter estimation, and classification. In general, it is desirable to have specific correctness information for every object instead of just taking the overall numbers of detected objects for the analysis, which is supported by an exact pairing of detected objects and reference data in the following. Then, parameter sets for the system are optimized with regard to different performance metrics using a central (hyper)parameter optimizer as part of an evaluation framework, which is described in the following. This process can also be deployed to a high-performance cluster (HPC).

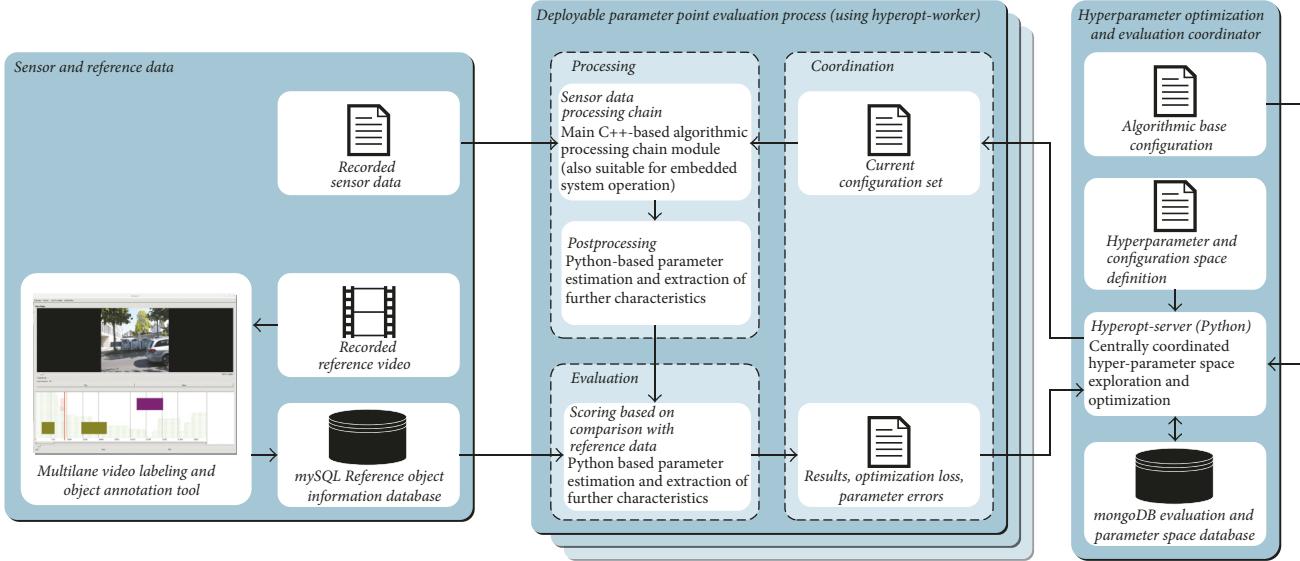


FIGURE 9: Evaluation framework and methodology for parameter space exploration.

5.1. Parameter Space Exploration and Optimization Methodology. In the following, the framework for parameter space exploration of the whole system with its corresponding algorithms is specified. The basic structure is shown in Figure 9.

Parameter Set Evaluation Concept. As a main principle, (hyper)parameter sets are given to a worker shown in the central block in the diagram. Each worker then evaluates a single parameter set (deployed with the *Coordination* subsystem) for real-world recorded sensor data using the C++-based main processing chain and Python-based postprocessing as part of the *Processing* subsystem. The processing principles were described in the implementation description of this article in Section 4.5. Then, in the *Evaluation* subsystem, the objects resulting from the processing stage are taken together with data from an object reference database. This allows the analysis of several performance metrics, such as the detection F_1 score and the correct lane assignment, which yields a performance score or optimization loss as a quality indicator of the parameter set. This is then reported back to the *Coordination* subsystem of the worker.

Reference Database. On the left side of the diagram, the sensor and reference data are shown. Reference object data is stored in a MySQL database. A reference video is recorded in parallel to a sensor data recording. This video is then manually processed using a labeling tool which is also shown in the diagram. The tool allows labeling traffic participant objects on multiple lanes in a timeline view and adding annotations to these objects such as the vehicle type, vehicle size, movement direction, and velocity, which is stored in the database. By analyzing the raw video material itself, the tool also provides the activity level in the video material which supports finding the next vehicle passing by in less frequented time ranges.

Cluster-Label-Pairing for Scoring. Both the resulting clusters from the processing of the raw sensor data and the objects in the reference database are events extended in the time dimension, with a specific start and end time. In the detection performance analysis of the system, each cluster (in case of perfect detection) would be paired with the corresponding reference label object. As the exact cluster-label-pair assignment is not the case in a real scenario, there can be ambiguities or false-positive/false-negative cases. Therefore, a bipartite cluster-label-graph is built up from the cluster and reference data, with edges between the nodes of specific cluster-label candidate pairs. Then, a maximum cardinality bipartite matching is performed. With the resulting cluster-label-pairs and the known sets of all reference and cluster objects, all scorings such as precision, recall, accuracy, and F_1 -score can be calculated and reported back for the specific parameter set.

Coordination of (Hyper)parameter Sets. Given the possibility of using a worker process to evaluate parameter sets with the described procedure, the exploration of the (hyper)parameter space needs to be coordinated and optimized. This is achieved by using the hyperopt Python package [37], which allows the definition of several (hyper)parameters with their according distributions and properties and then to automatically explore this space using simulated annealing, random search, and tree of Parzen estimator techniques for optimization. The parameter sets and their according loss results are stored in a MongoDB database. New parameter sets are integrated into a base configuration file which is then assigned to the worker process. In order to speed up the optimization and training procedure, a large number of workers are deployed on an HPC. Typically, good convergence is achieved after 10,000 optimization iterations. If a faster convergence is desired, starting points for the parameters can be calculated with the investigations done in Section 4.3. Typical

TABLE 2: Overview of test scenarios and their characteristics.

Scenario name	Number of lanes	Sensor distance range ¹	Sensor mounting height	Typical speed range	Pulse interval T_{rep}	Description
Urban1	2	7.5 m	3.5 m	20–50 km/h	100 ms	Urban alley street with trees on both sides and parked cars on the adjacent side, mixed use by cars, buses and bicyclists, sensor placement in a horizontal distance of 1 m to curb
Urban2	2	8 m	3.5 m	20–40 km/h	100 ms	Sensor placement behind sidewalk, distance to curb 2 m, mixed use by cars, buses and bicyclists, reflective trash containers on adjacent side
Urban3	2	7 m	5 m	20–50 km/h	100 ms	Typical “urban canyon” with large buildings on both side close to the street and narrow sidewalks, distance to curb 20 cm
TestTrack	-	>15 m	3.5 m	5–100 km/h	50–100 ms	Automotive test track without reflective objects, only asphalted road surface in sensor range, low rate of vehicles passing by with highly different speeds

¹The distance range is the distance from the ground projection point of the sensor position to the curb side or delimiter of the adjacent street side. For the calculation of the time-of-flight equivalent distance, both this sensor distance range and the mounting height have to be incorporated.

(hyper)parameters, which are part of the optimization process, are as follows:

- (i) Dimensions of ε -environment ($\varepsilon_T, \varepsilon_D$) for the DBStaC algorithm and the core point threshold level MinSum (see Section 4.3.3)
- (ii) Properties of the statistics memory, such as the statistics memory size (past information, see Section 4.4.2) and a storage subsampling factor
- (iii) Optional use of statistical feedback (see Section 4.4.2) in the scenario for base distribution stabilization
- (iv) Optional clipping threshold for calculated statistical norms in standardization to stabilize against outliers
- (v) Optional use of matched filtering for the received envelope signal

5.2. Test Scenarios. In Table 2, the different test scenarios are given with their characteristics and description, covering a variety of urban environments and synthetic measurements on an automotive test track. The scenarios are identified by a specific name and will be referenced in the following, when performance results are given.

6. Results and Discussion

Results of the field tests and evaluations are given in this chapter. First, the according performance metrics are introduced and specified, followed by the results for the different scenarios and finally a discussion of the results.

6.1. Evaluation Metrics. In this paper, the main focus lies on the detection of traffic participants as objects and their correct assignment to the lanes of the street. For describing performance in our object detection scenario, no calculation

of the true-negative count is possible, as we can have an arbitrary number of objects in our data timeline. Therefore, the widely used F_1 score is chosen as the main performance metric for detection, being the harmonic mean of precision and recall. The precision P is defined as the number of true positives (T_P) divided by all positive outcomes:

$$P = \frac{T_P}{T_P + F_P}. \quad (19)$$

In contrast, the recall R is defined as the number of true positives divided by the sum of true positive and false-negative (F_N) outcomes:

$$P = \frac{T_P}{T_P + F_N}. \quad (20)$$

Then, the F_1 score is defined as follows:

$$F_1 = 2 \frac{P \cdot R}{P + R} = \frac{2 \cdot T_P}{2 \cdot T_P + F_N + F_P}. \quad (21)$$

A further advantage is that the combination of precision and recall into the F_1 score allows being independent of requirements biases, which can be a focus on either precision or recall performance, for example, when more false positives (F_P) or negatives are acceptable in a specific scenario. F_1 score calculation only requires the information of true positives, false positives, and false negatives. These numbers are calculated based on the bipartite cluster-label-graph assignment introduced in Section 5.1.

The second important metric is the assignment of objects to the correct lanes for determining the direction of the traffic flow. With multiple lanes as a multiclass problem, the F_1 score cannot be used directly. As an alternative, the weighted F_1 score is used, which first calculates the F_1 score for every lane (correct or incorrect assignment to specific lane) and then calculates a weighted mean of all lane scores, with the number of true reference object instances as the weight.

TABLE 3: Field test evaluation results for different scenarios.

Scenario name	Object count	F_1 score detection	Weighted F_1 score lane assignment
Urban1	224	0.98	0.91
Urban2	133	0.92	0.95
Urban3	305	0.97	0.98
TestTrack	17	0.94	—

6.2. Discussion of Test Results. The performance evaluation results for the different scenario with optimized parameters are shown in Table 3. Both detection and lane assignment scores are always larger than 0.9, being a very good performance level in real scenarios and satisfying the initially stated high requirements on traffic information quality. The single-sensor ultrasonic traffic participant detection is therefore possible with high performance without exploiting any directional information of the signal. This is facilitated by the algorithms proposed in this paper, which solved several prevailing problems in ultrasonic sensing techniques.

The multilane assignment performance is also very high and of special importance. As no directional or velocity information is available with these sensors, the known information of a fixed lane's driving direction yields the final traffic flow information with direction. A possible problem is showing up in the scenario *Urban1* with the lowest weighted F_1 score for assignment of 0.91: people were driving in the middle of the two-lane street several times, which compromised lane assignment performance.

Further future long-term analyses are required to analyze the long-term stability in scenarios with a highly fluctuating traffic flow and day-and-night cycle. However, with the system structure which relies on the standardized data together with the statistics memory, the sensitivity to long-term sensor drift and also production variations is limited. Further investigation is also required on the impact of the statistical feedback for stabilization of the statistical base information. As the parameter optimization is based on only a few parameters, the susceptibility to overfitting effects is limited in the shown results. Still, for future investigations, also the test performance with regard to future measurements in the same scenario is of high interest.

7. Conclusion and Future Work

In this paper, it was shown that sidefire ultrasonic sensing is a viable option for multilane traffic participant sensing, giving very good performance results in real-world urban scenarios with a single sensor. The functionality is enabled by a novel combination of standardization techniques based on windowed statistics and a modified density-based clustering algorithm, together called DBStaC. Several evaluations were performed, both in real urban environments and for special cases on an automotive test track. The proposed system is integrated into an evaluation and parameter optimization methodology, whose reference system is flexible to be extended with further object characteristics in future.

The ultrasonic sensor system deployed together with the street lamp platform was presented to be a Smart City technology suitable for high-coverage deployment in cities, enabling traffic monitoring and further applications. With this platform, distributed processing and sensor fusion can expand the possibilities of using available traffic participant information even more, for example, for future applications like trajectory prediction. Further future work can be the use of ultrasonic sensor arrays for acquiring directional and velocity information. In the field of algorithmic improvements, the investigation of more advanced hypothesis testing techniques and channel statistics analyses could yield further performance improvements. Also, the comparison of the presented algorithmic concept with state-of-the art supervised machine learning algorithms such as recurrent neural networks is planned.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] H. van Essen and A. van Grinsven, “Final report appendix 9: Interaction of GHG policy for transport with congestion and accessibility policies,” Project Report, EU Transport GHG 2050, 2012.
- [2] “Roadmap to a single european transport area - towards a competitive and resource efficient transport system,” White Paper, European Commission, 2011.
- [3] H. Mayer, “Air pollution in cities,” *Atmospheric Environment*, vol. 33, no. 24–25, pp. 4029–4037, 1999.
- [4] R. Arnott, A. de Palma, and R. Lindsey, “Does providing information to drivers reduce traffic congestion?” *Transportation Research Part A: General*, vol. 25, no. 5, pp. 309–318, 1991.
- [5] A. Zanella, N. Bui, A. P. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [6] N. Buch, S. A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.
- [7] S. Himmel, M. Zieffle, and K. Arning, *From Living Space to Urban Quarter: Acceptance of ICT Monitoring Solutions in an Ageing Society*, Springer, Berlin, Germany, 2013.
- [8] W. Xue, L. Wang, and D. Wang, “A prototype integrated monitoring system for pavement and traffic based on an embedded sensing network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1380–1390, 2015.
- [9] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, “A vehicle classification based on inductive loop detectors,” in *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics*, pp. 460–464, May 2001.
- [10] L. Klein, D. Gibson, and M. Mills, *Traffic Detector Handbook*, vol. 1, no. FHWA-HRT-06-108, Federal Highway Administration, Turner-Fairbank Highway Research Center, 3rd edition, 2006.

- [11] M. Hallenbeck and H. Weinblatt, *Equipment for Collecting Traffic Load Data*, Transportation Research Board, NCHRP Report 509, 2004.
- [12] N. Garber and L. Hoel, *Traffic and Highway Engineering, SI Edition*, Cengage Learning, 2014.
- [13] R. Möbus and U. Kolbe, "Multi-target multi-object tracking, sensor fusion of radar and infrared," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 732–737, IEEE, June 2004.
- [14] I. Urazghildiiev, R. Ragnarsson, P. Ridderström et al., "Vehicle classification based on the radar measurement of height profiles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 245–253, 2007.
- [15] I. Urazghildiiev, R. Ragnarsson, K. Wallin, A. Rydberg, P. Ridderstrom, and E. Ojefors, "A vehicle classification system based on microwave radar measurement of height profiles," in *Proceedings of the 2002 International Radar Conference*, pp. 409–413, Edinburgh, UK, October 2002.
- [16] J. Tao, S. Chen, L. Yang, and Y. Hu, "Ultrasonic technique based on neural networks in vehicle modulation recognition," in *Proceedings of the The 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 201–204, October 2004.
- [17] E. U. Warriach and C. Claudel, "Poster abstract: A machine learning approach for vehicle classification using passive infrared and ultrasonic sensors," in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks, IPSN 2013 - Part of CPSWeek 2013*, pp. 333–334, April 2013.
- [18] T. Matsuo, Y. Kaneko, and M. Matano, "Introduction of intelligent vehicle detection sensors," in *Proceedings of the 1999 Intelligent Transportation Systems Conference*, pp. 709–713, Tokyo, Japan.
- [19] H. Kim, J.-H. Lee, S.-W. Kim, J.-I. Ko, and D. Cho, "Ultrasonic Vehicle Detector for Side-Fire Implementation and Extensive Results Including Harsh Conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 3, pp. 127–134, 2001.
- [20] V. Agarwal, N. V. Murali, and C. Chandramouli, "A cost-effective ultrasonic sensor-based driver-assistance system for congested traffic conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 486–498, 2009.
- [21] C. Heipke, "Crowdsourcing geospatial data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 6, pp. 550–557, 2010.
- [22] S. Hind and A. Gekker, "Outsmarting traffic, together: Driving as social navigation," *Exchanges: the Warwick Research Journal*, vol. 1, no. 2, pp. 165–180, 2014.
- [23] M. B. Sinai, N. Partush, S. Yadid, and E. Yahav, "Exploiting social navigation," <https://arxiv.org/abs/1410.0151v1>.
- [24] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative Vehicle-to-Vehicle communications," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks, LCN 2010*, pp. 606–612, October 2010.
- [25] S. Jeon, E. Kwon, and I. Jung, "Traffic measurement on multiple drive lanes with wireless ultrasonic sensors," *Sensors*, vol. 14, no. 12, pp. 22891–22906, 2014.
- [26] Y. Jo and I. Jung, "Analysis of vehicle detection with WSN-based ultrasonic sensors," *Sensors*, vol. 14, no. 8, pp. 14050–14069, 2014.
- [27] H. Kuttruff, *Room Acoustics*, Spon Press, New York, NY, USA, 5th edition, 2009.
- [28] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2nd edition, 1999.
- [29] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.
- [30] B. Gold, A. V. Oppenheim, and C. M. Rader, "Theory and implementation of the discrete Hilbert transform," *Symposium on Computer Processing in Communications*, vol. 19, 1970.
- [31] M. Meissner, "Accuracy issues of discrete hubert transform in identification of instantaneous parameters of vibration signals," *Acta Physica Polonica A*, vol. 121, no. 1A, pp. A164–A167, 2012.
- [32] L. Xu and T. Xu, *Digital Underwater Acoustic Communications*, Elsevier Science, 2016.
- [33] S. C. Kumbhakar and C. A. K. Lovell, *Stochastic Frontier Analysis*, Cambridge University Press, Cambridge, UK, 2003.
- [34] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, pp. 226–231, AAAI Press, 1996.
- [35] J. Gan and Y. Tao, "DBSCAN revisited: Mis-claim, un-fixability, and approximation," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD'15, pp. 519–530, ACM, New York, NY, USA, 2015.
- [36] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.
- [37] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*, S. van der Walt, J. Millman, and K. Huff, Eds., 2013.

