Microphone array based classification for security monitoring in unstructured environments

Simone Scardapane, Michele Scarpiniti, Marta Bucciarelli, Fabiola Colone, Marcello Vincenzo Mansueto, Raffaele Parisi

Abstract

The aim of this paper is to describe a novel security system able to localize and classify audio sources in an outdoor environment. Its primary intended use is for security monitoring in severe scenarios, and it has been designed to cope with a large set of heterogeneous objects, including weapons, human speakers and vehicles. The system is the result of a research project sponsored by the Italian Ministry of Defense. It is composed of a large squared array of 864 microphones arranged in a rectangular lattice, whose input is processed using a classical delay-and-sum beamformer. The result of this localization process is elaborated by a complex multi-level classification system designed in a modular fashion. In this paper, after presenting the details of the system’s design, with a particular emphasis on the innovative aspects that are introduced with respect to the state-of-the-art, we provide an extensive set of simulations showing the effectiveness of the proposed architecture. We conclude by describing the current limits of the system, and the projected further developments.

© 2015 Elsevier GmbH. All rights reserved.

1. Introduction

In the last decade, the wide availability of cheap sensor instrumentation has made automatic surveillance an economical and technical possibility. A notable example in this sense is the Secure Perimeter Awareness Network (SPAN) in use at the J.F.K. International Airport [1], an integrated system of sensors which is used, between others, for automatic intrusion detection in the perimeter of the airport. Of particular interest for their flexibility and cheapness are the systems based on acoustic sensors [2]. When we consider generic outdoor scenarios, an equivalent automatic security monitoring system based on a microphone array would be an invaluable tool in assessing and controlling any type of situation occurring in them [3]. This includes, but is not limited to, handling large civil events, or increasing the awareness of a terrain in military contexts. Moreover, a sensor-based system possesses an intrinsic degree of security, being by design a completely passive device.

However, implementing automatic outdoor security systems able to work with noisy, realistic and diversified data in an unstructured environment is a challenging and largely unexplored area. This is particularly due to noise, air distortion, low signal-to-noise ratio and presence of multiple, possibly conflicting sources.

In this context, last year the Italian Ministry of Defense funded the SMART-OPTIGRID project, carried on by Intecs S.p.A. in collaboration with the DIET Dept. of “Sapienza” University of Rome. The SMART-OPTIGRID project is aimed at a feasibility study of a microphone array based acoustic antenna for the detection, localization and classification of source sounds in severe outdoor scenarios. In particular, we are interested in classifying a set of sources acquired from a large set of sensors, and locate the concurrent presence of weapons, vehicles and/or spoken sources. Therefore the conceived system should be highly reliable, reasonably limited in size so that it can be moved if necessary, and extremely adaptive to different operative conditions. To this purpose particular attention has been devoted to the design of the microphone array geometry, to the definition of proper detection and localization strategies, and to the development of innovative classification techniques to be effective in the considered scenarios.

From an algorithmic point of view, the first innovative aspect of our system is the investigation of an array composed of a large number of microphones, which are integrated into a small surface of limited size. The second innovative aspect, instead, is the...
introduction of a modular four-level classification stage, designed to cope with the large number of possible sources that can be present in an environment of interest.

Regarding the state-of-the-art, automatic security monitoring by means of collected audio signals falls under the broader field of Computational Auditory Scene Analysis (CASA) [4], whose aim is to successfully analyze a stream of continuous audio to identify and isolate the sources of interest contained in it. The audio can be acquired either (i) using large acoustic antennas [2,5,6] (which is our design choice), or (ii) using distributed sensors (e.g., [7]). The subsequent analysis is typically performed by applying state-of-the-art machine learning techniques [8] to recognize the presence of specific objects. This last problem is a notable example of Automatic Audio Classification (AAC) [9], the task of automatically labeling a given audio signal in a set of predefined classes. Generally an AAC system works by subdividing the audio signal in small, overlapping frames, extracting some statistical features, and finally classifying them using a standard machine learning tool.

Due to the reasons detailed above, AAC has been studied primarily in the context of single-level applications, where the classes are restricted to a very specific domain. For example, there exists a vast literature regarding speech discrimination [10–13], vehicle recognition [14–16] and weapon classification [17,18,7]. In addition, due to the maturity of the field there exist several commercial and open-source products that perform these tasks, such as the Halo system1 and the Sphinx toolkit.2

If we consider a system with the need of performing more than one of the aforementioned tasks, however, their combination is not as straightforward as it may appear. A complex, realistic classification system has the need of being highly modular, flexible, and hierarchical, topics that were only marginally considered in the learning literature until the last decade [19]. Regarding AAC for security monitoring, a small number of systems were proposed recently that undertake this direction, particularly by first separating the speech detection problem from the non-speech detection. Atrey et al. [20] presented a four-level system for event detection. However, they used a single sensor to gather information, and considered only binary classification tasks. In our work, instead, we consider an array composed of a large number of microphones, and are interested in classifying a wide range of possible sources of interest. Abu-El-Quran [21] and Zhao et al. [22] detail two systems that, starting from a microphone array, perform at the same time speech and non-speech recognition. Although their works bear some resemblance to the system we detail in this paper, they were primarily meant for use in an indoor application, and the non-speech classification was performed in a single step. An early application of the idea of multi-stage classification to an audio stream of data is described instead in [23].

In this paper, we detail the steps taken to design the various components and select the appropriate learning tools. Moreover, we show how we explicitly take into account the presence of air distortion by the use of virtual examples [24], to make our system robust to them. The result of this is a modular and flexible classification system that efficiently combines several small classifiers by virtue of its own structure. Some preliminary simulation results show the effectiveness of the proposed architecture.

The rest of the paper is organized as follows. In Section 2 we describe the general architecture of our system. Then, in Sections 3 and 4 we go into more detail with respect to the beamforming and classification operations, respectively. A brief analysis of the computational cost of the proposed architecture is given in Section 5. Some empirical evaluations are presented in Section 6 and finally Section 7 concludes the paper.

2. General system architecture

The surveillance system described in this paper is based on a microphone array acoustic antenna to be employed for detecting, localizing and classifying heterogeneous sources in severe outdoor scenarios.

The desired system specifications include the capability to operate in a wide search volume that spans the angular interval $[-45^\circ, 45^\circ]$ in both the azimuth and the elevation directions. Moreover, this should be accomplished by using narrow listening beams with $-3$ dB aperture of few degrees. Both impulsive and continuous acoustic emissions should be taken into account with a coverage that might reach $8\div10$ km for high power sources. In particular the considered sources should include vehicles, aircrafts, weapons and spoken sources.

The conceived system architecture is described schematically in Fig. 1. The input of the system is provided by a square array of 864 microphones mounted in a triangular lattice. The microphones are considered to be omnidirectional and have a flat frequency response in the acoustic band. The design choices pertaining the acoustic array sub-system are detailed in the following section.

A set of properly steered listening beams are extracted from the microphones’ raw output by performing a standard delay-and-sum beamforming operation [25,26]. At this stage, particular attention has been devoted to the design of appropriate strategies able to guarantee the required angular coverage even in the presence of sources with an impulsive nature (see Section 3). Once the active sources have been detected and localized at some listening beams, the corresponding signals are fed to the input of the processing stages responsible for the sources classification. This represents a very challenging task owing to the need to classify highly heterogeneous sources, possibly active at the same time in the environment.

To this purpose each of the designated beams’ outputs is split in small audio frames, from which a collection of 42 statistical descriptors are extracted. The features are then passed as input to a hierarchical classification system that categorize them on each of the classes that we considered. The overall classifier is composed of four, modular stages requiring smaller binary classifiers. This allows to subdivide the original learning tasks into a series of smaller and simpler tasks that are efficiently solved by each of the classifier’s modules. The detailed description of the innovative classification technique proposed in this paper are reported in Sections 4 and 6.

3. Beamforming

In its resting position, the microphone array consists of $N_{\text{mic}}$ microphones arranged on a planar surface (2D array). As it is well known, by jointly processing the signals collected at the available microphones it is possible to synthesize a listening beam steered towards the selected search angular quantity. The electronic steering is obtained by summing the received signals after proper compensation of the different delays induced at each microphone, so that the contributions from a given direction are coherently integrated (so-called delay-and-sum beamforming [27,26]). This results in a significant improvement with respect to the single microphone since the listening beam shows a much higher gain and increased angular discrimination capability due to the extremely narrower beam-width.

With reference to the SMART-OPTIGRID project, aiming at limiting the final array dimensions, the beamformer has been designed to guarantee a listening beam with $-3$ dB aperture of

---

1 http://www.roke.co.uk/halo/
2 http://cmusphinx.sourceforge.net/
3° in both azimuth and elevation at the operating frequency of 4 kHz when the beam is steered towards 0°. Consequently a squared shape has been selected where each side is about 1.5 m long.

The distance $d$ between microphones has been carefully designed to avoid ambiguities appearing in the array pattern shape. Specifically this condition should be preserved when varying the steering direction within the angular sector to be monitored. This has been obtained by arranging the microphones on an equilateral triangular lattice shown in Fig. 2. It is worth noticing that the implementation of this lattice might imply a significant saving in term of required microphones with respect to the standard rectangular lattice.

Based on this choice, the distance between adjacent microphones should be lower than 5.59 cm. By setting $d = 5.5$ cm, the number of required microphones is obtained equal to 864; these are arranged on 64 rows and 27 columns elements following the lattice depicted in Fig. 2.

Notice that the use of this huge number of microphones yields a theoretical gain of about 30 dB in terms of signal to noise ratio; moreover it makes the system robust to amplitude and phase noise affecting the microphones since this effect is compensated by averaging over the many microphones if the considered noise is random distributed [28].

Finally, the use of tapering functions has been considered to lower the pattern sidelobes typically appearing at 13 dB below the beam peak. To this purpose, among all solutions available in literature, due to its efficiency a Taylor weighting function has been employed, which allows to trade the sidelobe level with the beamwidth broadening [29].

An example of a 2D listening beam pattern is reported in Fig. 3 when it is steered at 0° in both azimuth and elevation without sidelobes control. In this configuration, the maximum SNR gain is equal to 29.36 dB and the highest sidelobe is 13.26 dB below the peak. The main lobe apertures at −3 dB in elevation and azimuth are $\delta \theta = 2.85°$ and $\delta \phi = 2.92°$, respectively.

Obviously the beam-width broadens as the listening beam is steered and when a tapering function is applied. As an example, Fig. 4 shows the pattern steered at $(30°, -30°)$ in elevation.
and azimuth, respectively, when using a Taylor amplitude tapering function that lowers the sidelobe level down to $-35$ dB with respect to the beam peak.

Steering the listening beam towards a given direction allows to search for acoustic sources included in the angular sector defined by the beam-width. In this regard we recall that the considered system should be able to detect and classify also acoustic sources emitting impulsive sounds. Therefore the search function cannot be performed by sequential scanning the listening beam within the search volume. In contrast, multiple beams should be contemporaneously formed with steering angles properly displaced so that the required angular coverage is instantaneously guaranteed. Obviously, the cluster of beams should be carefully designed aiming at limiting the maximum number of beams while yielding a partial overlapping among adjacent beams. The obtained cluster of beams is depicted in Fig. 5 where black perimeters represent the $-3$ dB contours of beam patterns at the operative frequency of 4 kHz without amplitude tapering. The source detection can be achieved by searching among all the listening beams for the ones yielding signals with power level exceeding a predetermined threshold. For example, in Fig. 5, we report the output for the case of two different sources from directions $(30^\circ, 30^\circ)$ and $(0^\circ, 0^\circ)$, respectively. The first source emits a continuous sound (like a car engine), while the second source is responsible for an impulsive sound (i.e., rifle shot). They are received such that the impulsive sound SNR is $3$ dB lower than the distributed noise SNR. Different colors are representative of different power levels (in dB) registered at the beams constituting the cluster. Once the sources have been detected, a rough localization is obtained by assigning to the detected source the steering angle of the beam in the cluster where the detection occurred. Finally the localization procedure can be refined by exploiting a denser cluster of beams formed around the first angle estimate.

4. Multi-level classifier

In this section we detail the development of the feature extraction and classification blocks of Fig. 1. With respect to the former, an important aspect is the use of a large, heterogeneous set of statistical descriptors, able to cope with a large range of possible sources. In the latter, instead, we introduce a modular four-level classification stage, and we show how to take into account the presence of air distortions by the use of so-called virtual examples.

4.1. Feature extraction

This module receives from the previous stages a set of reconstructed audio signals, one for each window in which a sound source has been localized. Each signal is initially segmented in small superframes of 200 ms, with an overlap of 50%. To detect the presence of silence, superframes are on turn segmented in smaller frames of 50 ms with no overlap. If the Root Mean Square (RMS) [9] of a frame is under $-30$ dB, the frame is discarded as silence. The superframe is then reconstructed from the remaining constituent frames.

Next, 42 features are extracted from each superframe, and passed as input to the classifier for the discrimination process. Each feature is obtained by segmenting again the superframe in smaller frames, computing a given descriptor for each subframe, and then retaining only some overall statistics on the computed values. Each computation uses a subframe of different length and overlap. In Table 1 we have collected the name of the features, and the specific window size and overlap used for each one. The descriptors have been chosen so as to efficiently represent different domains of the audio signal: characteristics such as the ZCC or the RMS are extracted in the time domain, whilst VSF/Lux or SBC pertain to the frequency domain. The reason is that the utility of each descriptor changes depending on the nature of the task. As an example, time-domain descriptors are particularly useful whenever we are interested in discriminating sounds of impulsive nature. Due to the heterogeneity of our classification task, a good amount of work has been spent in optimizing the set described in Table 1, without making the computational load on the system too expensive. For this purpose, we have decided to discard cepstral features since their computation is notoriously complex and time-consuming. For completeness, in the following we provide a brief description for each feature. For a comprehensive explanation of each feature, we refer to [9] and references therein.

The Zero Crossing Count (ZCC) is defined as the number of times the signal reverses its sign. The values of the ZCC over the sub-frames are subdivided into 10 bins, and the normalized frequencies of these bins are used as features for the superframe, together with the variance of the ZCC values (denoted as VZCC). Another feature extracted here is the High Order ZCC Ratio (HZCR), defined as the ratio of the number of frames whose ZCC is above 1.5 times the average ZCC of the superframe.

The RMS of each superframe is also computed. As before, the RMS values are subdivided into 10 bins, the bins are normalized, and their relative frequencies are used as features. Additionally, we compute the ratio of Low Energy Frames (LEF), i.e., the ratio of frames whose RMS is lower than 50% the average RMS for the superframe. The mean value of a similar feature, the Short Time Energy (STE) [9], is also computed. Then, we extract the Low Short Time Energy Ratio (LSTER), defined as the ratio of the number of frames whose STE is less than half the average STE of the superframe.

For the following features the superframe is convolved with a low-pass filter to retain only the frequencies between 1.5 kHz and 1.6 kHz. Both the low-band superframe and the original superframe are then segmented. The Low-Band Energy Ratio (VLER) is defined as the variance of the ratio of the low-band STE to the whole band STE. Next, the variance of the Spectral Flux (VSFLUX) is computed, where the spectral flux is defined as the 2-norm between the frequency spectra of the current frame and of the previous frame. An additional feature computed here is the Low Frequency RMS (LFRMS), where the LFRMS is defined as the median of the RMS of the low-band version of the frame. The Low Frequency ZCC (LFZCC) is computed as the 9-th moment of the ZCC values of the frames. Additionally, we compute the Sub-Band Correlation (SBC) [30] for 4 different filters of increasing frequencies, and the High-Order Crossing (HOC) [31] up to the 4-th order.

![Fig. 5. Strategy for the coverage of the search volume and detection/localization approach.](image-url)
Table 1
List of the features extracted from the reconstructed signal.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Window (ms)</th>
<th>Overlap (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCC</td>
<td>Zero Crossing Count</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VZCR</td>
<td>Variance of Zero Crossing Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HZCR</td>
<td>High Order Zero Crossing Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOC</td>
<td>High-Order Crossing</td>
<td>5</td>
<td>2.5</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LFRMS</td>
<td>Low-Frequency Root Mean Square</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTER</td>
<td>Low Short Time Energy Ratio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLER</td>
<td>Variance of Low-Band Energy Ratio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSLFLX</td>
<td>Variance of Spectrum Flux</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>LFZCC9</td>
<td>9th-order Moment of LFZCC</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>SBC</td>
<td>Sub-Band Correlation</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

For each feature, we provide the window length and overlap (in milliseconds) used for segmenting the original superframe.

4.2. Classifier architecture

The classification step is performed by a modular four-level architecture, whose components are summarized in Fig. 6. This has been developed so as to cope with a high level of noise, typically found in a severe outdoor scenario, and the large heterogeneity of the objects we intend to discriminate. The main advantage given by the multi-level design is that the initial task is subdivided into multiple subtasks which are simpler to solve and of rising difficulty (this last characteristic is needed since the error on one level may propagate to the following). Hence, each classifier can be customized in more detail with respect to a single problem. Moreover, the architecture is highly flexible with respect to adding or deleting new categories of objects. Finally, each level brings a confidence score in its own prediction. The analysis of these values allows for a more informed decision in real-time, depending on the specific requirements of the system in deployment phase, and on the required granularity level.

Next, we discuss briefly the choice of each layer. The first level is devoted to distinguishing speech from non-speech sound. This task, known under the name of Automatic Speech Detection [10–12], has been extensively studied in the literature, since it is basilar to any system requiring speech enhancement, speech recognition and (as in our case) speech classification. Today, it is known that the frequency components extracted from the signal are sufficient to perform this task with a high precision, even in the presence of noise and of talking sources with different characteristics (sex, accent, verbal dysfunctions and so on).

The next level is a coarse-grained classifier, able to distinguish between a restricted set of two broad categories: vehicles and weapons. The idea of coarse-to-grained classification is discussed in [32], and can easily be justified by some informal judgments. In particular, the average frequency spectrum of a weapon presents ample differences from the average spectrum of a vehicle [3], and these differences are easier to learn with respect to the specific differences in spectrum inside each category. This same reasoning applies also to the third level of the architecture, whose task is to distinguish between aerial and ground vehicles. Finally, the system has three sets of fine-grained classifiers that are trained on specific classes of each group. In our implementation, the weapon classifier separates guns from cannons; the aerial vehicle classifier separates jets from helicopters; and finally the ground vehicle classifier separates tanks from trucks.

Note that the classification step is performed frame-wise. We discard a frame if the confidence score at any level of the classifier is below 40%. Then, the overall class is chosen by performing a majority vote between the remaining frames. If all frames are discarded, the object is labeled with an additional class denoting an unknown object. In the current design, all features are fed to every stage of the architecture. Additionally, no feature selection strategy is used, since by some preliminary experiments it was not found to provide valuable improvement in performance.

4.3. Choice of the classifiers

Our last design choice involved finding a suitable learning model for each block in Fig. 6, and associated learning algorithms. For simplicity, we decided to make the same choice at every layer. After analyzing several possibilities, including a K-Nearest Neighbor (KNN) algorithm, and a Support Vector Machine (SVM), we decided to employ a Multilayer Perceptron (MLP) [8]. Particularly, this choice derives by the possibility of an eventual hardware implementation [33], although even more efficient architectures can in principle be considered [34]. In an MLP, the input \( x \) to the system is propagated to \( N \) hidden nodes, and the output of the \( i \)-th hidden node is computed as:

\[
h_i(x) = \sum_{j=1}^{d} w_{ij} x_j, \quad (1)
\]

where \( d \) is the dimensionality of the input \( d = 42 \) in our case), \( w_{ij} \) is the weight connecting the \( j \)-th input to the \( i \)-th input node, and \( f(.) \) is a non-linear function called the activation function of the hidden neuron. Denoting by \( h = [h_1, \ldots, h_N] \) the vector of outputs of the hidden neurons, the overall output of the network is then computed as:

\[
y(h) = g\left( \sum_{j=1}^{N} w_{ij}^o h_j \right), \quad (2)
\]

where \( w_{ij}^o \) is the weight connecting the \( j \)-th hidden node to the output node, and \( g(.) \) is the output activation function. In our implementation we used \( N = 50 \) hidden nodes, and all hidden and output neurons use hyperbolic tangent sigmoid activation function \( f(.) = \tanh(.) \). The weights of each network are trained using standard back-propagation with an \( L_2 \) regularization term [8]. All classifiers in Fig. 6 are binary, hence the network has only two possible outputs, in the set \( \{ -1, +1 \} \). In the testing phase, the sign of Eq. (2) defines the chosen class, i.e., the final output is defined as \( y' = \text{sign}(y) \), while the value of Eq. (2) is used as confidence value.
Table 2

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>Number of microphones</td>
<td>864</td>
</tr>
<tr>
<td>B</td>
<td>Number of beams</td>
<td>615</td>
</tr>
<tr>
<td>L</td>
<td>Number of samples</td>
<td>1024</td>
</tr>
<tr>
<td>f_s</td>
<td>Sampling frequency</td>
<td>16 kHz</td>
</tr>
<tr>
<td>T_f</td>
<td>Frame duration (beamforming)</td>
<td>64 ms</td>
</tr>
<tr>
<td>T_c</td>
<td>Frame duration (classification)</td>
<td>200 ms</td>
</tr>
<tr>
<td>d</td>
<td>Number of features</td>
<td>42</td>
</tr>
<tr>
<td>N</td>
<td>Hidden nodes for each MLP</td>
<td>50</td>
</tr>
<tr>
<td>S</td>
<td>Recovered sources</td>
<td>1–5</td>
</tr>
</tbody>
</table>

Values considered in the proposed system are provided in the third column.

5. Considerations on the computational cost

In this section we provide a brief analysis of the computational cost required by the proposed system. In particular, we discuss the cost required by the beamforming and classification operations, when performed on a standard CUDA architecture [35], in term of floating point operations (FLOPs) and FLOP per second (FLOPS) respectively. To provide a flexible analysis, we consider a general setting with a variable number of microphones, beams, etc. All the symbols that are used next are summarized in Table 2. For simplicity, we suppose that the GPU memory is sufficient to store all the necessary matrices and vectors, and we neglect the time required to transfer back and forth data from the CPU to the GPU and vice versa. A thorough analysis of these aspects goes beyond the scope of the current paper, and would require a deeper knowledge of the system’s components.

With respect to the beamforming operation, in our analysis we considered the possibility to guarantee a simultaneous coverage of the desired angular sector with a fixed number of beams $B$. The operations to steer a beam towards a desired direction are twofold: (1) the computation of the weights to be applied to the data collected by each microphone in the array and (2) the beamforming, that hereinafter is considered to operate the Delay and Sum operations in the frequency domain. The computation of the beamforming weights is performed just once for all the beams that we want to form. This means that the evaluated coefficients do not need to be computed for each frame, and they can be evaluated offline and stored in a memory to be accessed when needed by the digital beamformer. On the other end the application of these coefficients needs to be repeated for each frame to form the desired beams in the angular sector of interest. This goal is achieved in four steps: (1) amplitude weighting applied to the signal received by each microphone in the array to lower the sidelobes of the formed pattern, (2) FFT of the signal received by each array element, (3) weighted sum of the signal from all the microphones in each frequency bin for each beam to form, and finally, (4) IFFT of the output samples for each formed beam. It is possible to verify that the number of required operations is:

$$C_{\text{beam}} = L[m(1 + 2.5\log_2(L) + 4B) + B(\log_2(L) - 1)] \text{FLOPs.}$$  \hspace{1cm} (3)

Each recovered signal is then segmented in superframes of length $T_c$ as detailed in Section 4.1. The analysis of the feature extraction part is extremely hard to perform analytically, and it would depend strictly on the hardware components. However, previous works have shown it to be particularly efficient [36], and for this reason we consider here a fixed overhead of $C_{\text{feat}}$ for each frame. With respect to the classification part, training is performed offline and is not considered here. Additionally, we analyze the worst-case scenario, in which no silence is present in the recovered signals, and all superframes are analyzed. For a prediction, each MLP in Fig. 6 requires a vector–matrix multiplication followed by the application of a nonlinearity, and these operations are repeated twice for each layer. If we suppose that the nonlinearity is implemented via a standard lookup table, the classification cost is then:

$$C_{\text{class}} \approx P [dN + 2N + 1] \text{FLOPs,}$$  \hspace{1cm} (4)

where $P$ is the number of MLPs that are activated in the tree hierarchy of Fig. 6. Considering the values in Table 2, we find that the overall beamforming operation requires 34.85 GFLOPS, so that the overall cost is:

$$C_{\text{tot}} \approx 34.85 + \frac{S}{T_c} \left[ C_{\text{feat}} + P \left( 2 \times 10^{-6} \right) \right] \text{GFLOPS.}$$  \hspace{1cm} (5)
Clearly, even for a large number of recovered sources, the cost of beamforming is dominant on the overall architecture. However, this is easily affordable using standard GPUs, such as the NVIDIA QUADRO 1000M, which has a peak processing power of nearly 270 GFLOPS in single precision.

If there is an explicit need in keeping limited the number of floating point operation per second, a different approach can be considered. Differently from the previous case, a grid of $R$ angular sectors each one covered by a set of $B_0$ beams can be considered to provide the overall coverage not simultaneously but sequentially, such that signals from the same directions are acquired for $T_{oss}$ seconds with a certain revisit time $T_r$. The same expressions for the computational cost apply, but only $B_0$ beams must be simultaneously formed instead of $B$. As an example, considering an observation time $T_{oss}$ equal to three frames ($T_{oss} = 192$ ms), and $B_0 = 41$ we obtain $C_{beam} = 2.66$ GFLOPS. Similarly, for $B_0 = 215$ we obtain $C_{beam} = 12.42$ GFLOPS. Clearly, lowering the revisit time leads to a higher computational cost, and vice versa. This means that by decreasing $B_0$, a lower computational cost could be experienced, with respect to the case of the formation of $B$ simultaneous beams. At the same time, impulsive acoustic signatures with a duration smaller than the revisit time could be lost and therefore not correctly detected and classified.

6. Simulated results

To test the accuracy of the proposed architecture, the overall system has been simulated in a MATLAB environment. The results of the experiments are presented and analyzed in this section. In particular, in Section 6.1 we show the accuracy of the classifiers during the preliminary training phase. Then, in Section 6.2 the overall SMART-OPTIGRID architecture is simulated, and the results of the classification phase are analyzed for a large set of heterogeneous objects. Finally, in Section 6.3 we detail the current limits of the architecture, and the successive development steps that need to be undertaken.

6.1. Classifier training

In order to train the classifiers, we used a collection of sound effects bought from the web-library Sounddogs. In particular, we used the libraries relative to Cannons, Helicopters, Jets, Trunks, Tanks and Guns, for a total of 103 sound effects. To these, we supplemented a set of 8 recordings of male and female speeches registered in a controlled environment. All the sounds were resampled at 16 kHz, and features were extracted according to the algorithm described in Section 4.1. However, one tenth of the set of audio files for each class are removed from the dataset, to be used in the testing phase of Section 6.2. Each classifier is trained independently of the others. Here, a problem we encountered is given by the unbalance of the classes. For example, in the speech detection layer sounds corresponding to speech comprises only 1% of the overall dataset, and this can potentially hinder the training phase. Hence, we decided to rebalance the classes by randomly sampling the dataset prior to training. Classifiers are then tested using a 5-fold cross-validation [8].

In the first column of Table 3 we show the average classification accuracy for each layer. We see that speech detection works almost perfectly, with an average of 4 erroneous frames every 100 frames. Accuracies of 90% or higher are also obtained for the coarse classifier and the weapon classifier. The following layers have accuracies ranging between 80% for the aerial vehicle classifier to 83% for the vehicle type classifier. Since in reality decisions are averaged over a large set of frames, these accuracies seem largely sufficient to ensure that, in an ideal situation, objects are correctly classified.

However, in a realistic outdoor scenario sounds are typically distorted by the presence of air [37]. To simulate this, we convoluted a randomly chosen subset of our testing data with a low-pass filter, and tested the previously trained classifiers over the distorted data. Accuracies of this phase are shown in the second column of Table 3. It can be seen that the presence of air distortion is dramatic in terms of classification accuracy. Even the best performing classifier, the speech one, has a sudden decrease in accuracy of 16% points. Two classifiers, the one for weapons and the one for ground vehicles, perform practically as random guessing in the presence of distortions, and the classifiers for the vehicle type and aerial vehicles also obtain rather useless results, ranging around 60% of accuracy.

Hence, we had to search for an efficient way to make the system invariant to air distortion. In the end, we adopted the approach of using virtual examples [24], i.e., adding to the training set another set of distorted sounds. In particular, we generated the following distortions:

- Distorted audio at 50 m, 100 m and 250 m for the speech files.
- Distorted audio at 750 m, 1000 m and 2000 m for the other classes.

Classifiers are again trained using a 5-fold cross-validation, this time on the extended dataset. Results of this phase are shown in the third column of Table 3. In most cases, accuracies are restored to the original levels. We only witness a decrease with respect to the first column for the weapon classifier, and a lower decrease for the aerial vehicle and vehicle type classifiers. This is balanced by the fact that in two cases (speech classifier and coarse classifier) the presence of virtual examples actually increases the overall classification accuracy. Classifiers trained in this last phase are then used for the simulations of the next section.

6.2. Classifiers testing

In this phase, the classifiers obtained in the previous training phase are combined with a full simulation of the beamforming operation, including air distortion and attenuation of the signals due to distance. As detailed previously, we use a set of audio files that were not considered in the training phase, so as not to incur in overfitting. For each sound in our testing dataset, the following procedure is adopted:

1. We place the sound on the simulated terrain, with horizontal orientation extracted randomly from $[-45^\circ, 45^\circ]$, vertical orientation extracted randomly from $[-5^\circ, 45^\circ]$, and distance extracted randomly from [500, 3000] m. This last constraint is relaxed for the speech files, which are placed at a random distance in [50, 500] m.

### Table 3

Accuracy of the training phase for the proposed classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech/non-speech</td>
<td>96%</td>
<td>80%</td>
<td>98%</td>
</tr>
<tr>
<td>Coarse classifier</td>
<td>93%</td>
<td>79%</td>
<td>94%</td>
</tr>
<tr>
<td>Weapon classifier</td>
<td>90%</td>
<td>50%</td>
<td>82%</td>
</tr>
<tr>
<td>Vehicle type classifier</td>
<td>83%</td>
<td>57%</td>
<td>81%</td>
</tr>
<tr>
<td>Aerial vehicle classifier</td>
<td>80%</td>
<td>62%</td>
<td>76%</td>
</tr>
<tr>
<td>Ground vehicle classifier</td>
<td>82%</td>
<td>50%</td>
<td>82%</td>
</tr>
</tbody>
</table>

In Phase I, classifiers are trained and tested on the original sounds. In Phase II, classifiers of Phase I are tested on distorted versions of the files. Finally, in Phase III classifiers are trained again by supplementing the dataset with virtual examples.
2. We run a full simulation of the system, and classify the recovered audio signal. Beamforming is executed on 12 windows of 256 ms each.
3. We repeat steps (1)–(2) 10 times by varying the location of the object.

The results of this experiment, averaged over the multiple runs, are presented in Table 4. In order to show the effectiveness of the proposed system over a wide range of heterogeneous sources, results for each class are broken down into their constituent categories.

As we expected, speech classification works practically flawless, both in the presence of male speakers and in the presence of female speakers. The system is also able to classify with a very high accuracy a large set of diverse weapons, including semi-automatic assault rifles (the AK-47), pistols (e.g., the Beretta M1) and submachine guns (e.g., the Uzi weapon). Optimal performance is obtained for all the families of jet airplanes that we considered, either standard bombers (e.g., the B-1-B) and fighters (e.g., the F-16 or the MIG). Only slightly smaller is the performance for tank recognition, particularly in the presence of the M1 Abrams battle tank.

Inferior testing accuracies are obtained for the remaining classes, although always ranging ahead of 70%. In particular, the system has the most trouble with some specific subcategories of each class, including anti-aircraft cannons (accuracy of 50%), the Robinson helicopters (possibly due to their size with respect to the others helicopters that are considered), and finally the weapon carrier trucks. In fact, trucks are the most troublesome category here, with also the Ford SVT obtaining a 70% accuracy rate. This is mostly due to the fact that, over the 200 ms frames that we use as input to the classification system, the frequency spectrum of a truck has large similarities with comparable spectra of some tanks.

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Accuracy</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>Male Talker</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Female Talker</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Cannon</td>
<td>Anti-Aircraft</td>
<td>50%</td>
<td>75.66%</td>
</tr>
<tr>
<td></td>
<td>Mortar</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Howitzer</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Weapon</td>
<td>AK-47</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Giat</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remington</td>
<td>90%</td>
<td>98.33%</td>
</tr>
<tr>
<td></td>
<td>Beretta</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uzi</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Colt</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Helo</td>
<td>Robinson</td>
<td>90%</td>
<td>76.66%</td>
</tr>
<tr>
<td></td>
<td>Bell</td>
<td>80%</td>
<td></td>
</tr>
<tr>
<td>Jet</td>
<td>Harrier</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B1-B</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>F-16</td>
<td>100%</td>
<td>97.5%</td>
</tr>
<tr>
<td></td>
<td>MIG</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Tank</td>
<td>M1</td>
<td>80%</td>
<td>93.33%</td>
</tr>
<tr>
<td></td>
<td>Sherman</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Truck</td>
<td>Ford SVT</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Humvee</td>
<td>80%</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>Weapon Carrier</td>
<td>60%</td>
<td></td>
</tr>
</tbody>
</table>

Results for each class are further subdivided into multiple families to show the heterogeneity of the considered sounds.

6.3. Limits of the current architecture

Despite the system shows a remarkable degree of accuracy over a large set of possible objects, it can be seen that performance of Section 6.2, particularly in the case of trucks, is inferior to what could be expected from the training phase of Section 6.1. In addition, in some preliminary experiments with scenarios involving multiple sources active at the same time, performance of the SMART-OPTIGRID system was found consistent when sources are separated by at least a few degrees, but it deteriorates when sources are very near to each other, due to the overlap of the recovered signals. Hence, the next development phase will involve the design of specific sound enhancement algorithms, including a noise cancels [38] and a source separation method [39,40].

7. Conclusion

In this paper we have detailed the design of a security monitoring system, called SMART-OPTIGRID, for the automatic classification of audio sources in an unstructured and severe outdoor scenario. The system involves an array of a large number of microphones, whose input is processed by a delay-and-sum beamformer followed by a modular classification architecture. Our simulations show the effectiveness of the proposed model, achieving an high degree of classification accuracy over a wide range of possible sources.

Future works will involve the inclusion of a set of audio enhancement strategies, described in Section 6.3, to overcome the current limits of the architecture, particularly in term of overlapping signals.

Acknowledgment

The authors are very grateful to IntecS S.p.A for its support.

References


