Sound-Event Recognition with a Companion Humanoid

Maxime Janvier[†], Xavier Alameda-Pineda[†♯], Laurent Girin[‡†], and Radu Horaud[†]
[†]INRIA Grenoble Rhône-Alpes, [‡]GIPSA-LAB and [♯]Université Joseph Fourier

***Abstract*—In this paper we address the problem of recognizing everyday sound events in indoor environments with a consumer robot. Sounds are represented in the spectro-temporal domain using the stabilized auditory image (SAI) representation. The SAI is well suited for representing pulse-resonance sounds and has the interesting property of mapping a time-varying signal into a fixed-dimension feature vector space. This allows us to map the sound recognition problem into a supervised classification problem and to adopt a variety of classifications schemes. We present a complete system that takes as input a continuous signal, splits it into significant isolated sounds and noise, and classifies the isolated sounds using a catalogue of learned sound-event classes. The method is validated with a large set of audio data recorded with a humanoid robot in a house. Extended experiments show that the proposed method achieves state-of-the-art recognition scores with a twelve-class problem, while requiring extremely limited memory space and moderate computing power. A first real-time embedded implementation in a consumer robot show its ability to work in real conditions.**

I. Introduction

For the last decade, robots have gradually moved from well-structured factory floors to unstructured populated spaces. There is an increasing need of robots interacting with humans in the most natural way. It is generally agreed that this human-robot interaction paradigm is conditioned by robust and efficient robot perception capabilities. There has been a tremendous amount of work towards endowing robots with *visual perception*. This allows robots to model their environment, to safely navigate, to detect people and to recognize their everyday actions, gestures and facial expressions. Nevertheless, vision has its own limitations, e.g., it cannot operate in bad (too dark or too bright) lighting conditions, and the interaction is inherently limited to objects and people that are within the visual field of view. The analysis and understanding of auditory information could help robots improve their understanding of various acoustic events, and hence both extend their range of perceptive modalities to hearing and improve their interaction capabilities with humans and with their environment. While speech is a fundamental way of communication, non-speech sounds also convey a lot of information about the ongoing situation and are equally important. By being able to exploit both verbal and non-verbal auditory information, robots may participate in ongoing activities, understand humans based on emitted auditory signals, and react to a wide variety of acoustic events. For example, a home robot that is able of identifying a temporal sequence of audio signals emitted by various domestic appliances may cooperate with the tenant. More generally, on top of speech processing, speech recognition, and language understanding, a domestic robot should be able to identify a wide spectrum of sounds emitted by people, by artifacts, and by interactions between people and artifacts, and to interpret them in terms of an extended catalogue of events, e.g., Fig. 1. This can be addressed within the emerging framework of *machine hearing* [7]. Nevertheless, there is an important difference between static arrays of microphones and handheld devices such as smart phones, on one hand and a robot equipped with microphones, on the other hand. A robot can implement *active listening* of an auditory scene, by simultaneously recognizing and localizing a particular sound source as well as moving towards the source in order to select an optimal location thus allowing *multimodal* perception and communication.

Fig. 1: This figure shows a robot companion, i.e., the humanoid robot NAO, listening to a person involved in an everyday kitchen activity. Using the continuous sound recognition method described in this paper, NAO is able to extract a discrete sequence of isolated sounds and to interpret them in terms of a pre-stored catalogue of auditory events (microwave alarm, open microwave door, close microwave door, toaster, etc.), all in the presence of background noise, competing sound sources, and reverberations. Remark how difficult would be to recognize this human activity using vision.

The focus of this paper is on the automatic recognition of non-verbal sounds. People are able to easily identify a large amount of sounds they hear in their everyday life, e.g., the closure of a door or the barking of a dog, as well as distinguish the sound of the microwave oven from the sound of the washing machine; These tasks are not trivial to achieve with an artificial agent such as a humanoid robot. For

Work supported by the EC project HUMAVIPS FP7-ICT-2009-247525

example, different objects can produce similar sounds that should, or should not, be classified together depending on the application at hand. On the opposite, the same physical object can produce different types of sounds that may not belong to the same sound event category. In addition to that, the sounds can come from different objects placed at different positions relatively to the robot head. The audio signals that are perceived by the robot's microphones are perturbed by various non-linear filtering effects (the robot's head-related transfer function is difficult to estimate), by sounds emitted by the robot itself including noise coming from the hardware located inside the robot head, by room reverberations and by competing sound sources.

The sound recognition problem is relatively new in both audio signal analysis and in robot hearing literatures when compared with the prolific field of automatic speech recognition (ASR), or with the sound source localization problem. It can be addressed in two manners. First, recognition can be carried out directly on a continuous audio stream, as for most ASR systems. Alternately, it can consist of two modules (Fig. 2): in a first stage, an audio event detection module isolates relevant sounds from silence or background noise; then a classification module identifies isolated sounds. A lot of work has been done on sound event detection. Standard approaches are based on thresholding the signal energy [24], [1], [11], [20]. More complex methods rely on wavelet coefficient trees [9], long-term signal variability [4], power envelopes [10], or support vector machines [16]. As for the sound recognition techniques, most of them are derived from state-of-the-art speech recognition, which consists of a supervised training of Hidden Markov Models (HMM) feeded with Mel-Frequency Cepstrum Coefficients (MFCC) [19], [17]. This approach has been recently applied to sound recognition [18]. Other supervised statistical techniques have been proposed, e.g., using Gaussian mixture models (GMM) with LFCC [2], MFCC and other spectral features [22], or wavelet transforms [5]. In opposition to the parametric approaches, a non-parametric learning method based on sparse coding of stabilized auditory images (SAI) has been proposed in [8]. Special purpose sound recognition methods addressed applications such as security surveillance [18], [22], detection of critical situations in health homes [5], [2], or footstep recognition [12].

In this paper we propose a complete sound detection-recognition system embedded in a consumer humanoid, and dedicated to indoor environments. Compared to the state-of-the-art methods just cited, the proposed method has the following novelties.

First, the input audio signal is recorded using low-quality microphones embedded in a robotic head. As already mentioned, there are both non-linear filtering effects and inside-head noise which alter the quality of the signal and challenge most of the existing recognition methods. Moreover and unlike the prevailing ASR paradigm which works well with clean speech signals gathered with a close-range microphone, or *tethered* interaction, the emitted-sound-to-perceiver distance is in the range of several meters, or *untethered* interaction. Hence, the perceived signal incorporates a fair amount of reverberations, which is particularly the case in an indoor environment. Second, the computing and power resources on board of a consumer robot are inherently limited. Therefore, the ratio between performance and computational cost of the implemented algorithms should be particularly high.

We adopt a two-step detection-then-classification approach and we focus our attention on the recognition of *auditory events*. This term includes kinds of sounds with a short duration (under 1s) and with well-defined start- and end-points. Most of these sounds have an impulsive nature. They are very often associated with sound-emitting events such as a person opening a door, walking, dropping an object, etc., as opposed to continuous sounds. e.g., washing machines, fans, music, etc., which are treated in this paper as background noise. Continuous sounds cannot be processed by our approach which need sounds with finite and sufficiently short duration.

The detection splits the (noisy) continuous audio stream into a discrete set of isolated sounds that are further analyzed, one-by-one, by the sound classification algorithm using the recently proposed *stationary auditory image* (SAI) representation [23]. The resulting auditory images are mapped into a vector space of reduced dimension, and a vector quantization (VQ) technique is used to efficiently map the SAI representation in a feature space. The quantized vectors are used to classify the isolated sounds based on an off-line supervised training stage then learns prototype vectors. Notice that, whereas the testing data acquired by the robot in real-world conditions is a continuous audio stream, isolated sounds can be used to train the classifier, thanks to the sound detection module. It is worthwhile to stress the importance of performing the training using the robot's microphones in real conditions, as opposed to using an existing dataset of sounds gathered in a *clean* environment, e.g, unechoic rooms.

We note that recent approaches to sound classification [18], [22], [8] do not deal with the problem of sound detection. Hence, they are not suitable for a real-time implementation as needed in robotics, since the direct continuous analysis of the audio stream is prohibitive in terms of computational complexity. Some other approaches merely oriented towards interaction [5], [2], rely on static microphones mounted in specially equipped spaces. However, in the present study the effect of the environment on the audio signal depends on the relative position of the audio sources and the robot, which are unknown and variable. Summarizing, up to the authors' knowledge, no work has been reported before on a complete sound recognition algorithm based on a cascaded detection and VQ-based learning method which has been implemented on humanoid robot, and both trained and tested in a natural indoor environment.

In order to prove the concept of sound event recognition with a consumer robot, we placed the robot in a kitchen.

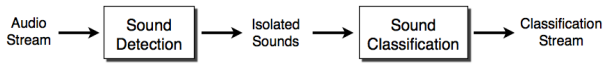


Fig. 2: The proposed machine hearing system consists of two modules. The sound detection module segments the input audio stream into isolated sounds which are categorized by the sound classification module

The ability to recognize everyday sounds that are typical of such a set-up is a prior to further understanding (high-level processing) of the scene. For instance, if some sounds are related to human actions (eat, cuttlery) or to human reactions (alarms, choking), the robot may expect a particular human behaviour accordingly. Hence, this may help the robot to detect anomalies in the ongoing human activity or in the environment.

The remainder of this paper is organized as follows. Section II describes the computation of SAIs using the auditory image model. The sound classification method is fully detailed in Section III. Experiments and results are presented in Sections IV and V respectively. Conclusions and future work are discussed in Section VI.

II. Sound representation: from waveform to SAI and Feature Vector

The sound recognition task can be seen as a matching task, in which sounds with similar characteristics are linked or related, hence belonging to the same class. For this aim, we need a sound representation that preserves and highlights the class membership. That is, a feature space in which sounds belonging to the same class are mapped together and far away from the features representing sounds of other classes. This is challenging for several reasons: (i) sounds can be of many different kinds (speech, music, prosody, interactions between objects, etc.), since they can be produced by many different devices (larynx and vocal tracks, fingers, feet, musical instruments, electronic and electric appliances, etc.), (ii) even inside the same class differences exist (voice timbre, instrument design, motor model, etc.) due to the variability of acoustic realizations, (iii) relevant information can be corrupted by irrelevant noise or competing sources, and (iv) sound waveforms (of the same or different classes) are generally of different lengths, hence requiring either normalization or temporal alignment prior to comparison. In the following, we look for an efficient representation of each sound as a feature vector of fixed size, in order to apply simple and efficient classification techniques based on vector quantization (see Section III). This way, each sound in the data set is represented as a point in the feature vector space, regardless of its class membership.

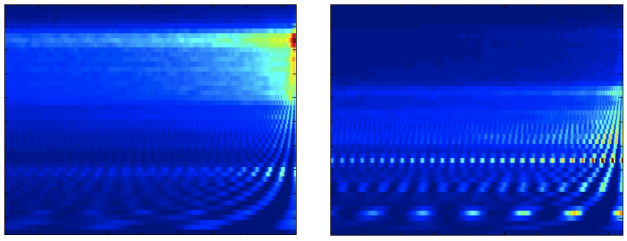
As the human auditory system is "designed" to ease the interaction with the environment, it is able to accurately represent communication sounds (speech and animal languages) as well as many other more or less natural sounds. Most of these sounds are pulse-resonance signals, i.e. they are generated by a series of pulses activating resonances in an emitting body, so that the traits of size and type of the emitting body are encoded in the resonances. Many models have been proposed to mimic the human/mammalian auditory system, explaining the way it reacts to such sounds. Based on Patterson's work on the human cochlea [14], [13], the auditory image model (AIM) presented in [23] produces stabilized auditory images (SAI), which are a time-frequency sound representation close to a correlogram. The AIM projects the main features of the pulse-resonance sounds into two different dimensions: on the one side, changes of pulse rate are equivalent to changes in frequency; on the other side, variations of resonance correspond to variations on the time scale.

The AIM is decomposed in three main stages. Roughly speaking, the first stage simulates the basilar membrane motion by means of a multi-channel gammatone filter bank [15]. Afterward, the neural activity pattern is computed using a half-wave rectification. Finally, the statistical distribution of the half-rectified signal peak's delay is estimated, leading to the SAI. Two examples of SAI resulting form those processes are shown in Figure 3, corresponding to *Water tap* and to *Eat*. We can observe the difference in pitch between the two sounds and there is a remarkable difference both in timbre (frequency dimension) and in resonance (time dimension).

The dimension of the SAI representation corresponds to the image size (the number of pixels), which leads to a representation with high dimensionality. A technique was proposed in [8] to reduce the dimensionality of the SAI features. This procedure consists of three steps: (i) create patches from the SAI, i.e., subsets of the set of pixels, (ii) compute a low-dimensional vectorial representation of each patch and (iii) concatenate these patch feature vectors to form the final feature vector. The D patches created from the SAI may vary in size and shape and may have some overlap. Changes of size and position modify the information contained in the patches. For instance, short patches have fine frequency resolution and capture local spectral shapes. Also, thin patches have fine temporal resolution, containing information about resonances. Step (ii) consists of the concatenation of the row-wise addition of the patch values. It has been found to be a good trade-off between reducing the dimensionality and keeping the temporal and spectral information. Given a sound x in the data set, the patch feature vectors are denoted by $u_1(x), \ldots, u_D(x)$ and are concatenated to form a Feature Vector (FV):

$$\mathrm{FV}(x) = (u_1(x), \ldots, u_D(x)) \tag{1}$$

To summarize, the SAI is computed and split in D patches. These patches are projected to a lower dimensional patch feature space to later concatenate them into a single feature vector. Note that for a given recognition system configuration, the size of the feature vector is fixed and independent of



(a) *Water tap* SAI. **(b)** *Eating* SAI.

Fig. 3: Two examples of SAI in the time delay-frequency representation: (a) *Running the tap* and (b) *Eating*. The color encodes the intensity: the lighter the more intense. The time axis and the frequency axis are respectively on the abscissa and on the ordinate. Notice how the difference in timbre and resonance are emphasized in the frequency and time dimensions respectively.

the length of the sound waveform (a very nice property that results form the fixed size of the SAI). Thereby, each sound is represented in the same vector space independently from its duration. The dimensionality of the space only depend on how is split the SAI.

III. Sound recognition based on Vector Quantization of Feature Vectors

Since the proposed method is a *supervised* method, we assume that we work with training data that are implicitely classified (the training phase basically consists of joint recording and annotation). The basic principle of recognition is thus to compare the unknown test data to the annotated training data, and find out the closest element so that the input data is associated to the corresponding class. When the number of sounds in the training set is large, two problems can appear: first, the memory requirement to store the training data can become too high; and second, it may become computationally too expensive to compare the incoming sound with the entire set of training sounds. Assuming that the sound representation is vectorial (see Section II), we use Vector Quantization (VQ)techniques to circumvent those problems. In the following, we first briefly describe the fundamentals of VQ, and then we present two variants of a classification method based on the split-Vector Quantization (split-VQ) algorithm [3].

A. Basics of VQ and split-VQ

Basically, VQ takes a set of Feature Vectors as input and quantize them, i.e. associate a vector prototype to each Feature Vector. Formally, a VQ quantizer is thus a function q that maps a feature vector space $\mathcal{U}$ to acodebook $\mathcal{C}$ according to:

$$q : \mathcal{U} \longrightarrow \mathcal{C}, q(u) = \underset{c \in \mathcal{C}}{\operatorname{argmin}}\, d(u, c), \tag{2}$$

where d stands here for the Euclidean distance. The codebook is a reduced set of feature vector prototypes in charge of representing the overall set of possible vectors. To design a codebook from training (Feature Vector) data, we use the classical K-means algorithm. Note that in the following we actually use a split-VQ [3], for instance a separate VQ for the different patch vectors that compose each Feature Vector (see SectionII). This is appropriate for the present problem since each patch represents different characteristics of the sound.

B. Class-Free Method

1) VQ codebooks design and Feature Vector coding: In the first implementation of our sound recognition system, the quantifier q is designed using the entire set of Feature Vectors of the training sound dataset independently of their class. That is why the method is called class-free. However, there is one codebook of size K ($K < S$ where S is the number of sounds in the data set) for each patch vector, therefore the K-means algorithm is run D times, one time for each patch vector training set, resulting in D general (i.e. class-independent) K-codebooks. After that, the Feature Vectors of the dataset are quantized using the resulting codebooks (as the concatenation of the quantized patch vectors) to form the Quantized Feature Vectors (QFV):

$$\text{QFV}(x) = (q_1(u_1(x)), \ldots, q_D(u_D(x)))\,. \tag{3}$$

2) Classification Phase: Once the codebooks are designed, the system is ready to classify any new isolated incoming sound, x^I. After being mapped onto the SAI-derived Feature space, the Feature Vector corresponding to the new sound is quantized to $\text{QFV}(x^I) = \big(q_1(u_1(x^I)), \ldots, q_D(u_D(x^I))\big)$. All prototypes $\text{QFV}(x)$ are compared to the incoming one. The training sound x^* with minimum Euclidian distance $d\left(\text{QFV}(x^I), \text{QFV}(x^*)\right)$ is selected, and x^I is classified to the class to which x^* belongs.

Note that in the Class-Free method, there are as much prototype vectors as original training vectors, and each prototype vector resulting from the quantization of a given data vector is assumed to belong to the same class. Thus, there is no reduction of the number of comparisons between a new vector to be classified and the prototype vectors. However, (i) the VQ is basically used as a coding technique to significantly reduce the memory requirement for the storage of training vectors, and (ii) since the comparison is made on a quantized patch basis, pre-calculated values of Euclidian distance between quantized patches can be stored and do not need to be recalculated during the recognition phase, saving significant computational resource. Split-VQ is an interesting strategy to perform nearest neighbor search as shown in [6]. Using a different subquantizer q_i for each subvector u_i (and therefore for each patch in the SAI) has two main advantages. First, if we use a simple VQ, the vector space is only discriminated on a codebook with K differents prototypes, while with a split-VQ with M subquantizers, you can generate $M \times K$ prototypes with the same memory usage and have a weaker distortion on the quantization step. Secondly, the energy dispersion is not uniform in the SAI,

so the subvector components are not in the same range of values depending from which patch it is extracted. Using independant subquantizers allows to reduce the distortion introduced by the quantization.

C. Class-Constrained Method

1) VQ codebooks design on a per-class basis: In this approach, the codebooks are not designed and used to quantize the complete feature space as a whole, but to quantize each class separately. However, the split-VQ principle is preserved, so that there are now D codebooks (with K prototypes) for each of the C classes. Each training vector is quantized using the patch quantizers of its class. If we denote by q_{ij} the quantizer designed for the i-th class and for the j-th patch, a Feature Vector x is here quantized by:

$$\text{QFV}_i(x) = (q_{i1}(u_1(x)), \ldots, q_{iD}(u_D(x))) \,. \tag{4}$$

2) Classification Phase: After computing the Feature Vector corresponding to a new incoming sound x^I, C different $\text{QFV}_i(x)$ are calculated, using the C groups of D quantizers. The sound x^I is classified following a basic nearest neighbor approach, i.e., assigned to the class i^*:

$$i^* = \underset{i \leq C}{\text{argmin}} \; d\left(\text{QFV}_i(x), x^I\right) \tag{5}$$

Compared to the Class-Free configuration, here we have a reduced set of K prototypes for each class of sounds, so that the patches of the incoming Feature Vector are compared to the prototypes instead of the complete set of quantized data. These comparisons are made for each class, thus we have $K \times C$ distances to calculate instead of S, and the Class-Constrained method is computationally efficient especially when $K \times C << S$.

IV. Experiments

The experiments have been done with the NAO humanoid robot in a natural indoor environment. Notice that the use of a humanoid robot implies several challenges. First of all, the sensorial capabilities of NAO are limited; since the use of high quality devices that take a lot of place and electric power is not possible, only low-quality microphones are available. Secondly, the computational power is low, which makes the robot unable to execute computationally expensive algorithms. Also the use of a natural indoor environment encloses challenges such as indoor noise and reverberations.

In the following we describe NAO's auditory architecture, the data set and the experiments we conducted.

A. NAO's Auditory Architecture

All the data was recorded with the auditory architecture of the NAO robot. This robot has four microphones placed on its head as shown in Figure 4. The frequency bandpass of these microphones is 300 Hz to 18 kHz. The input signals were sampled at 48 kHz and 16 bits per sample. The robot's fan produces frequencies between 0 and 4 kHz which shades weak sounds into the noise.

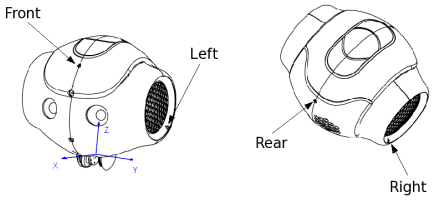


Fig. 4: NAO's microphones (front, rear, left and right).

B. A Data Set of Sound Events

The data set used to validate the proposed method was recorded following the rationale of a kitchen scenario. This scerario encloses a large range of everyday sounds of differents natures and is prone to reverberations. Also, the sounds in the kitchen scenario show a large variability both in terms of temporal and spectral characteristics. For example there are impulsives sounds (*Close the microwave*, *Choking*), harmonic sounds (*Microwave alarm*), and transients sounds (*Running the tap*, *Eating*). Table I gathers the 12 sound classes that we defined, organized in a taxonomy derived from the scenario. Furthermore and in order to evaluate the robustness to the change of position, we recorded sounds from three different positions (two on the floor, one in height) where NAO stays stationary. Each sound source kept its position in the room, so the distance between the robot and them depends on the position of NAO. The sound is picked from the right or the left microphone and can arrive from differents orientations to the head. At each position, 7 instances of each class were recorded and cut by hand, which makes 21 sounds per class, thus a total of 252 sounds.

Note that, aside from isolating the sound in the temporal domain, no post-processing such as noise removal or normalization has been applied to the recordings. Note also that we did not rely on existing data sets because of two reasons. Fist, our study concerns sounds in a indoor environment. Second, most of the existing data sets (BBC, freesounds.org, etc.) are recorded with sensors of significantly higher quality than NAO's ones, which make those recorded signals inappropriate for our purposes. Hence, up to our knowledge, there is no sound data set similar to ours[1].

[1]The kitchen dataset described in this paper will soon be made publicly available

Prosodic	Cooking	Moving	Alarms
Eating	*Cuttlery*	*Open/close a drawer*	*Microwave*
Choking	*Fill a glass*	*Move a chair*	*Fridge*
	Running the tap	*Open the microwave*	*Toaster*
		Close the microwave	

TABLE I: Taxonomy of the recorded dataset of the kitchen scenario. The twelve sound classes are organized into: **Prosodic**, **Cooking**, **Moving** and **Alarms**.

Method	Accuracy Mean	Accuracy Standard deviation
Class-Free	95.9%	3.9%
Class-constrained	91.9%	5%
HMM	96%	3.9%

TABLE II: Accuracy scores (mean and standard deviation) for the three different methods evaluated.

C. Experimental Settings

In order to validate and compare the proposed methods, we applied a formal evaluation strategy. The state-of-the-art audio recognition method presented in [18] was implemented and used as a baseline. In [18] sounds are represented as sequences of 12-dimensional MFCC. That is, classical MFCC features removing the energy coefficient, the delta MFCC as well as the delta-delta MFCC. The temporal model used is a five-state left-to-right HMM. The emission probability density function is a mixture of five gaussians. The training phase estimates the maximum likelihood parameters for each class. When an incoming sound has to be classified, the learned models are use to compute the per-class likelihood. The sound is assigned to the class with the highest likelihood.

Concerning the implementation of the proposed methods, we use SAI features which are produced by the C++ implementation developed by Walter and van Engen[2]. Several parameters have been experimentally set. The SAIs have 48 pixels on the frequency axis and 1680 on the temporal one. Each SAI is divided in 12×12 patches. As for the value of K, it has to be large enough to discriminate between different sound classes, and small enough not to overfit the model. Since the number of instances used to compute K-means is 252 in the class-free method and 21 in the class-constrained method, we set $K = 25$ for the first and $K = 6$ for the second.

V. Results

A. Recognition Scores

In order to be able to statistically compare the different sound recognition methods, we perform k-fold cross-validation. Since the number of sounds per class is limited, we run the k-fold cross-validation n times. By setting $k = 10$ and $n = 50$, the results are an average of 500 experiments. Due to this evaluation, the dataset recorded with NAO is used for both training and testing. We recall that the dataset is built without using the detection module. Table II shows the accuracy scores (mean and standard deviation) for the three methods. These scores are quite high: we obtain respectively 95.9% and 91.9% for the Class-Free and for the Class-Constrained methods. The HMM reference method is only 0.1% above the Class-Free method, at 96%, although HMM modeling is much more computationally demanding than our VQ-based methods. (Actually there is no statistical difference, in terms of accuracy, between the class-free and the HMM-based methods). However, both methods perform statistically better than the class-constrained method. An interesting fact is that without any quantification step and keeping the features vectors (FV) for the classification, the accuracy score drops to 88.9%. It shows that quantification step do not have necessarily a destructive effect and can even adds robustness to the system.

In order to have a more in-depth understanding, we also provide the mean confusion matrix obtained by the class-free, the class-constrained and the HMM-based methods in Tables III, IV and V respectively. While in the class-free method the errors are spread all along the confusion matrix, the other two methods have a few, clearly defined, systematic errors. For instance, the HMM-based method classifies some occurrences of *Close microwave* as *Drawer* and some occurrences of *Eat* as *Glass*. Besides that, the class constrained method classifies some occurrences of *Chair* as *Drawer*, and some occurrences of *Choke* and *Toaster* as *Close microwave*. Notice also that the class-free method missclassifies sounds that have similar spectro-temporal features. However, the class-constrained method seems to be less adapted to the SAI features, since the amount of miss-classifications is higher.

In order to test the Class-Free method in a continuous scenario (real conditions), we recorded a continuous audio stream consisting of a concatenations of several sounds. We recorded this scenario with NAO in a regular kitchen (as the presented data set), figure 1. Since the sounds were enclosed in a continuous stream, we need to detect the beginning and the end of each sound event in order to isolate it. We chose to use the state-of-the-art cross-correlation-based detection method described in [21]. After automatically isolating these "sound events", they are processed by the classification system. The experiments showed that the proposed Class-Free method is able to correctly classify sounds being the output of an automatic detection module. Hence, this proves the appropriatenes of the proposed method to be incorporated in a two-step detection-classification algorithm. Thus, demonstrating that the proposed approach is suitable to be used in a consumer robot.

B. Computational Considerations

We provide some comments regarding both the memory and computational powerrequirements of the proposed method.

[2]Available at `http://code.google.com/p/aimc/`

		Found class											
		Alarm f.	Alarm m.	Chair	Close	Cuttlery	Drawer	Eat	Open	Choke	Tap	Toaster	Glass
True class	Alarm fridge	**100**	0	0	0	0	0	0	0	0	0	0	0
	Alarm microwave	0	**100**	0	0	0	0	0	0	0	0	0	0
	Chair	0	0	**98**	0	0	2	0	0	0	0	0	0
	Close microwave	0	0	0	**97**	0	0	0	0	1	2	0	0
	Cuttlery	4	0	0	0	**86**	0	4	1	0	0	4	0
	Drawer	0	0	5	1	0	**94**	0	0	0	0	0	0
	Eat	0	0	0	0	0	0	**100**	0	0	0	0	0
	Open microwave	0	0	0	0	0	0	0	**100**	0	0	0	0
	Choke	0	0	0	9	0	0	0	0	**91**	0	0	0
	Tap	0	0	0	5	0	0	0	0	0	**90**	5	0
	Toaster	0	0	0	0	0	0	0	0	0	4	**96**	0
	Glass	0	0	0	0	0	0	0	0	0	0	0	**100**

TABLE III: Confusion matrix for the class-free method. The element ij corresponds to the percentage of sounds of the i-th class assigned to class j.

		Found class											
		Alarm f.	Alarm m.	Chair	Close	Cuttlery	Drawer	Eat	Open	Choke	Tap	Toaster	Glass
True class	Alarm fridge	**100**	0	0	0	0	0	0	0	0	0	0	0
	Alarm microwave	0	**100**	0	0	0	0	0	0	0	0	0	0
	Chair	0	0	**79**	0	0	**16**	0	0	5	0	0	0
	Close microwave	0	0	0	**93**	0	1	0	0	0	1	5	0
	Cuttlery	0	0	0	5	**90**	0	0	5	0	0	0	0
	Drawer	0	0	3	0	0	**95**	0	0	2	0	0	0
	Eat	0	0	0	0	0	0	**95**	5	0	0	0	0
	Open microwave	0	0	0	0	0	0	0	**100**	0	0	0	0
	Choke	0	0	0	8	0	0	0	0	**90**	0	2	0
	Tap	0	0	0	**16**	0	0	0	0	0	**75**	9	0
	Toaster	0	0	0	**13**	0	0	0	0	0	1	**86**	0
	Glass	0	0	0	0	0	0	0	0	0	0	0	**100**

TABLE IV: Confusion matrix for the class-constrained method.

		Found class											
		Alarm f.	Alarm m.	Chair	Close	Cuttlery	Drawer	Eat	Open	Choke	Tap	Toaster	Glass
True class	Alarm fridge	**100**	0	0	0	0	0	0	0	0	0	0	0
	Alarm microwave	0	**100**	0	0	0	0	0	0	0	0	0	0
	Chair	0	0	**97**	0	0	2	0	0	1	0	0	0
	Close microwave	0	0	0	**88**	0	**12**	0	0	0	0	0	0
	Cuttlery	0	0	0	0	**91**	3	0	0	2	0	4	0
	Drawer	0	0	0	0	0	**100**	0	0	0	0	0	0
	Eat	0	0	0	0	0	0	**80**	0	0	0	0	**20**
	Open microwave	0	0	0	0	0	1	0	**98**	0	0	1	0
	Choke	0	0	0	0	0	0	0	0	**100**	0	0	0
	Tap	0	0	0	0	0	0	0	0	0	**100**	0	0
	Toaster	0	0	0	0	0	1	0	0	0	0	**99**	0
	Glass	0	0	0	0	0	0	0	0	0	0	0	**100**

TABLE V: Confusion matrix for the HMM-based method.

Since the recognition methods rely on comparing quantized features vectors to the incoming feature vectors, all the QFV need to be stored. Regarding the class-free method, we need to store the K QFV (the centroids provided by the D K-means) plus all the codes (to which word of the codebook is assigned each patch of the training set). Indeed, since each patch of each training sound is mapped to one of the K centroids, we need to store the centroids and the result of this mapping. Each centroid of the codebook takes 16 bytes, and each reference takes 1 bytes. Thus, the total amount of required memory is: $M_{CF} = 16\,K\,D + S\,D$ bytes. In our case ($K = 25, D = 144, S = 252$) this sums up to 93.8 kB. Regarding the class-constrained method, the references are not needed any more, but we need to store C different codebooks, that is one per class. Thus, the amount of memory is $M_{CC} = 16\,K\,D\,C$ bytes, which makes a total of 165.6 kB in our set up ($K = 6, D = 144, C = 12$). Experiments with the class-free method show that the accuracy score does not improve significantly if $K > 20$. However, if the parameter K is too low, the performance drops exponentially (70% and 42% if $K = 8$ and $K = 4$ respectively). Empirically, The HMM-based method need 83 kB to store the models when $C = 12$. If it goes up to 42 classes (with 20 instances per class), the free-class method need 179 kB when the HMM-based method need 290 kB of data, which proves that the memory cost of the free-class method is less important when the number of classes increases.

As for the computational cost, the evaluation is not so informative, since the implementation is not optimized. With the current implementation, it takes 2.5ms to the Class-Free

algorithm to process the VQ-based recognition of this vector, whereas the class-constrained method needs 40ms, in our Matlab implementation. We need to take into account the time to compute the SAI (about 0,5 s). We used a CPU at 2,26 GHz. In comparaison, the HMM-based method (with Matlab) need 100ms to compute a complete classification task.

As a last remark on complexity, let us mention that the training phase takes about 4-6 s. This means that we can largely increase the number of sound classes and sound instances without leading to a prohibitive training time. Experiments with a higher number of classes (always with 21 sound instances per class) and $K \in [20; 60]$ show that the offline training takes about 8-10 s for 20 classes and 35-42 s for 42 classes.

To prove the viability of our method, we developed a real-time application (written in C++) using the middleware of NAO. Nao succeeded to perform classification on a 4-class problem (*hand clap*, *fingerclap*, *tongue clic*, and the spoken word *NAO*) in various and difficult situations (different rooms and positions, people *not in the training set*, from various ranges , etc.). We used the class-free method for a complete detection and classification task under 300ms. The main limiting factor remains the generation of the SAI. The process only use 7% of the CPU making our classification system suitable to run continuously.

VI. Conclusions & Future Work

In this paper, we address the problem of recognizing everyday sound events in indoor environments for home robots. Since the aim is to enhance the robot's understanding of its environment, we needed both: (i) a data set acquired with a real robot in a real environment and (ii) a method with a good trade-off between low computational complexity and good accuracy. The proposed sound recognition method is intuitive, principled and robust, and hence suitable for platforms with limited computational resources. We demonstrated that our two approaches using SAI features are suitable to perform sound recognition in indoor environnements. The accuracy results are good (greater than 91% in the worst case). The complexity of the proposed algorithms is also experimentally evaluated. We have shown that the system can work in real-time for a low computational cost and is suitable to help audio-visual systems to focus on new audio events.

This work can be extended in several directions. First, by improving the way we perform the classification phase in the class-free method. Somes techniques in data mining could enhance the response time, which will allow us to increase the number of training sounds. Second, by merging this module with a sound source localization and separation algorithm, to extend the proposed technique to complex, multi-source and noisier auditory scenes. Third, by adding an outlier class to account for sounds belonging to categories for which the system has not been previously trained. Finally, by testing the method on other robots to check that the methodology does not depend on the characteristics of the NAO robot.

References

[1] A. Dufaux, "Detection and recognition of impulsive sounds signals," Ph.D. dissertation, University of Neuchatel, 2001.

[2] A. Fleury, N. Noury, M. Vacher, H. Glasson, and J.-F. Serignat, "Sound and Speech Detection and Classification in a Health Smart Home," in *EMBS*. IEEE, nov 2008, pp. 4644–4647.

[3] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[4] P. K. Ghosh, A. Tsiartas, and S. S. Narayanan, "Robust voice activity detection using long-term signal variability," *IEEE Trans. on Speech, Audio and Lang. Proc.*, vol. 19, no. 3, pp. 600–613, 2011.

[5] D. Istrate, M. Binet, and S. Cheng, "Real time sound analysis for medical remote monitoring." in *EMBS*, 2008, pp. 4640–4643.

[6] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search." *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–28, Jan. 2011.

[7] R. Lyon, "Machine hearing: An emerging field," *Signal Processing Magazine, IEEE*, vol. 27, no. 5, pp. 131 –139, sept 2010.

[8] R. F. Lyon, M. Rehn, S. Bengio, T. C. Walters, and G. Chechik, "Sound retrieval and ranking using sparse auditory representations," *Neural Computation*, vol. 22, no. 9, pp. 2390–2416, 2010.

[9] D. I. M. Vacher and J. F. Serignat, "Sound detection through transient models using wavelet coefficient trees," in *CSIMTA*, sept 2004, pp. 367 – 372.

[10] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics," *IEEE Trans. on Speech and Audio Proc.*, vol. 10, no. 2, pp. 109 –118, feb 2002.

[11] M. Moattar and M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm," in *EUSIPCO*. EURASIP, 2009, pp. 2549–2553.

[12] A. Pakhomov, A. Sicignano, M. Sandy, and T. Goldburt, "A novel method for footstep detection with an extremely low false alarm rate," in *SPIE*, 2003.

[13] R. D. Patterson, "Auditory images : How complex sounds are represented in the auditory system," *J. Acoust. Soc. Japan E*, vol. 21, no. 4, pp. 183–190, 2000.

[14] R. D. Patterson and J. Holdsworth, "A functional model of neural activity patterns and auditory images," *Adv. in Speech, Hear. and Lang. Proc.*, vol. 3, pp. 547–563, 1995.

[15] R. D. Patterson and B. Moore, "Auditory filters and excitation patterns as representations of frequency resolution," in *Frequency Selectivity in Hearing*, B. Moore, Ed. Academic Press Ltd., 1986, pp. 123–177.

[16] A. Rabaoui, H. Kadri, Z. Lachiri, and N. Ellouze, "One-class SVMs challenges in audio detection and classification applications," *Journal on Advances in Signal Processing*, 2008.

[17] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*. Pearson, 2011.

[18] V. Ramasubramanian, R. Karthik, S. Thiyagarajan, and S. Cherla, "Continuous audio analytics by HMM and viterbi decoding." in *ICASSP*. IEEE, 2011, pp. 2396–2399.

[19] D. Reddy, "Speech recognition by machine: A review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 501 – 531, apr 1976.

[20] S. Tanyer and H. Ozer, "Voice activity detection in nonstationary noise," *IEEE Trans. on Speech and Audio Proc.*, vol. 8, no. 4, pp. 478 –482, jul 2000.

[21] M. Vacher, D. Istrate, L. Besacier, J. F. Serignat, and E. Castelli, "Life sounds extraction and classification in noisy environment," *in Proc IASTED International Conference on Signal Image Processing*, 2003.

[22] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *CAVSBS*. IEEE Computer Society, 2007, pp. 21–26.

[23] T. C. Walter, "Auditory-based processing of communication sounds," Ph.D. dissertation, University of Cambridge, 2011.

[24] K.-H. Woo, T.-Y. Yang, K.-J. Park, and C. Lee, "Robust voice activity detection algorithm for estimating noise spectrum," *Electronics Letters*, vol. 36, no. 2, pp. 180 –181, jan 2000.