

# Sound Representation and Classification Benchmark for Domestic Robots

Maxime Janvier, Xavier Alameda-Pineda, Laurent Girin, Radu Horaud

► **To cite this version:**

Maxime Janvier, Xavier Alameda-Pineda, Laurent Girin, Radu Horaud. Sound Representation and Classification Benchmark for Domestic Robots. 2014 IEEE International Conference on Robotics and Automation (ICRA 2014), May 2014, Hong-Kong, China. IEEE, pp.6285-6292, 2014, <10.1109/ICRA.2014.6907786>. <hal-00952092>

**HAL Id: hal-00952092**

**<https://hal.inria.fr/hal-00952092>**

Submitted on 26 Feb 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Sound Representation and Classification Benchmark for Domestic Robots

Janvier Maxime<sup>†</sup>, Xavier Alameda-Pineda<sup>†</sup>, Laurent Girin<sup>†,‡,#</sup> and Radu Horaud<sup>†</sup>  
<sup>†</sup>INRIA Grenoble Rhone-Alpes, <sup>‡</sup>GIPSA-LAB and <sup>#</sup>Université Grenoble Alpes

**Abstract**—We address the problem of sound representation and classification and present results of a comparative study in the context of a domestic robotic scenario. A dataset of sounds was recorded in realistic conditions (background noise, presence of several sound sources, reverberations, etc.) using the humanoid robot NAO. An extended benchmark is carried out to test a variety of representations combined with several classifiers. We provide results obtained with the annotated dataset and we assess the methods quantitatively on the basis of their classification scores, computation times and memory requirements. The annotated dataset is publicly available at <https://team.inria.fr/perception/nard/>.

## I. INTRODUCTION

In order to naturally interact with objects and people, robots need robust and efficient perception capabilities. For example, human-robot interaction requires the recognition of gestures, actions, and facial expressions. There has been tremendous progress towards endowing robots with visual perception. Nevertheless, the visual modality 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. In parallel to visual information, *sounds* produced by objects, by humans, or human-object interactions convey rich cognitive information about the ongoing context, events, and communicative behaviors.

Compared to visual analysis, audio analysis is complementary but it also has its own advantages. Visual data are huge, visual information is complex to extract, and hence efficient visual routines may be difficult to embed into the robot’s onboard hardware/software resources. In contrast, acoustic signal processing may be quite efficient, because the lower amount of data to be analyzed (depending however on the complexity of the acoustic scene). By using hearing, a robot may be able to recognize the ongoing events, estimate their relevance, and take appropriate decisions, even if they are not within the range of the visual sensors. Moreover, proper recognition and localization of sound events may be used to trigger visual attention mechanisms.

Therefore, audition is considered with increasing attention by robotic practitioners since hearing capabilities are likely to considerably improve the overall “cognitive understanding” of a scene as an extended catalogue of events, and

This work is financially supported by the “Direction Générale de l’Armement” (DGA), The French Government Defense

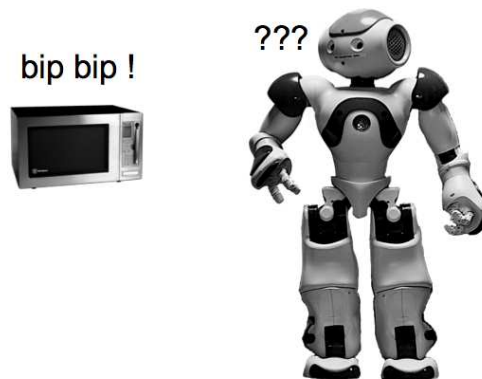


Fig. 1. Domestic robots, such as NAO, should be able to robustly recognize sounds in the presence of room reverberations, background noise and competing sound sources.

improve the interactive capabilities of robots with humans, as well as with animals and objects, including other robots. This is related to computational auditory scene analysis (CASA) which attempts to model the abilities of human audition, notably to segregate coherent auditory streams [1]. It covers a set of challenging problems some of which have already been successfully investigated in robotics: multiple-source localization [2] and separation [3], speech recognition [4], speech/non-speech/music classification, detection/segmentation and recognition of elementary sounds (possibly in background signal/noise), etc. Some of these modules were successfully integrated in robotic platforms, e.g., HARK [5] and ASIMO [6], to cite just a few.

In the framework of robot audition, this paper addresses isolated recognition of “domestic sounds”. We address both audio-signal representation and classification. The audio recordings are collected with a NAO robot manufactured by Aldebaran-Robotics<sup>1</sup>. Similar benchmarks can be found for example in [7] (for scene recognition), [8], [9] and [10]. This setup implies notable difficulties, the most notable one being the low microphone quality currently available with NAO. The collected sounds are from a real-world scenario, e.g., fig. 1: there are different types of sound sources, located at different (more or less distant) positions relatively to the robot head. The recorded audio signals are perturbed by room reverberations and by various linear or non-linear filtering effects (notably the robot’s head-related transfer function

<sup>1</sup><http://www.aldebaran-robotics.com>

which is difficult to estimate). The sounds are corrupted by the internal noise coming from the hardware inside the robot head. Also, the robot has limited computational capabilities, and this is expected to have a strong influence on the choice of signal representation and classification algorithms, as detailed below.

The experimental data and used in this paper stays in contrast with clean sound databases recorded with high-quality microphones in specially equipped rooms. Moreover, automatic speech recognition (ASR) techniques often use close-range microphones which is not the case here since the robot is at some distance from the audio sources.

We consider short sounds, typically in the range 0.1 to 1.0 seconds, that result from such events as the opening/closing of a door, people dropping an object or clapping hands, as opposed to continuous sounds or continuous sound streams. Many of these short sounds have an impulsive nature, and they are assumed to have well-defined start- and end-points. Therefore, basic detection techniques based on signal energy or other statistics can be used to pre-segment the signal before classification [11], and we do not address the detection/segmentation problem in the present paper: we assume that a correct segmentation of these short sounds is available. We also assume that the sounds do not overlap in time. Non-stationary sound streams such as continuous speech or music signals are not considered in the present study (although our dataset contains isolated spoken words, see Section II). Continuous speech is usually processed with specific classifiers, e.g., hidden Markov models (HMMs), that model the dynamic evolution of the spectral patterns corresponding to the successive phonemes [12]. Music signals are particularly tricky to process because of the richness of their content. More stationary sound streams such as the flow of tap water, washing machine, fans, etc., are not considered as well. The latter category can be considered either as long sound events or as background noise/context for overlapping short sound events. All these problems will be considered in future extensions of the present work which focuses on implementation of short sounds recognition in a robotic context. Note that this task is not trivial in itself, even without the limitations of the robotic context, depending on the number and complexity of the sound categories. For example, different objects can produce similar sounds that should, or should not, be classified together depending on the application. On the opposite, the same physical object can produce different types of sounds that may not belong to the same category. Our dataset contains 42 sound categories, which is a quite substantial number of sound types, as compared to previous studies, e.g., 10 as in [8], [13], [14], [9], 15-16 as in [15], [16] or 22 categories [10].

Our main goal is to carry out a benchmark assessing different signal representations (audio features) and different classifiers, in the spirit of what was done in, e.g., [17] for environmental sound recognition. We selected several feature spaces to represent sounds, as well as a number of classification techniques. Many possible combinations of

features and classifiers were tested, possibly to reveal general trends and propose an optimal solution.

Obviously, the accuracy score is the most important gauge for a classifier. The tested techniques are dedicated to be embedded in autonomous robots, hence other important indicators are analyzed and reported. First, robots have to work in (quasi) real-time, therefore execution has to be as fast as possible. Three time statistics are provided: the *feature computation time* (time to compute features from a raw signal of a given length), *training time* (time to train all the models for classification), and *recognition time* (time to classify a new incoming sound of a given length). Secondly, memory requirement is also a valuable resource in an embedded system, and we estimate the *training memory* (memory used to store the trained models). Getting the accuracy score, the computation times and memory costs for each feature/classification method will allow us to find optimal solution(s) or good trade-offs for reliable sound recognition with a consumer robot.

The remainder of this paper is organized as follows. Section II describes in detail the dataset recorded and used in this study. Sections III and IV present respectively the different features and classifiers that were used. Experiments and results are presented in Section V. Conclusions and the future work are expanded in Section VI.

## II. THE DATA

The dataset must have the following characteristics: (i) to be recorded with low-quality sensors, (ii) to suffer from typical internal robot noise, (iii) to be recorded in realistic domestic environments, i.e., in rooms with no special acoustic characteristics, presence of reverberations and of multiple sound-source randomly distributed across the room, and (iv) containing a substantial number of real-world sound types with only a few samples per class. Up to our knowledge, no existing database that fulfills these requirements is available. Therefore, we recorded a database by placing NAO in both a home and an office, and by using its frontal 300Hz – 18kHz bandpass microphone. The collected signals are sampled at 48kHz and quantized at 16 bits per sample. The robot-head fan produces noise within the band from 0 to 4 kHz, shading weak sounds. During recording, the robot stands still and hence is not affected by noise generated by its motion. The dataset is available online <sup>2</sup>. Four scenarios and 42 sound classes were considered, as summarized in Table I.

- **Kitchen:** The first part contains a large variety of everyday sounds collected in a home kitchen. We recorded 12 sound categories with different temporal and spectral characteristics: impulsive sounds (*Close the microwave*, *Choking*), harmonic sounds (*Microwave alarm*) and transient sounds (*Running the tap*, *Eating*). The sounds were recorded from three different positions, 1 to 5 meters range and at various angles from the sound

<sup>2</sup><https://team.inria.fr/perception/nard/>

source. At each position, seven instances of each class were recorded, which sums up to 21 examples per class.

- **Office:** The second part is related to an office environment. We acquired seven sounds: *Door close, Door open, Door key, Door knock, Ripped Paper, Zip, (another) Zip*. They were randomly recorded from 0.3 to 5 meters range and from various angles. All the sound related to door actions were recorded using different doors.
- **Non-verbal:** The third part of the data contains non-verbal sounds, which are produced by humans, and can be seen as communication signals, but typically not taken into account in ASR systems. There are three classes (*Fingerclap, Handclap, Tongue clic*) recorded from 0.3 to 5 meters range and from various angles, with four different people.
- **Speech:** The fourth part of the dataset contains occurrences of isolated words. Even if speech recognition is not in the scope of the present work, we judged of great interest to test methods designed for short sounds recognition on such speech samples. Hence, we recorded twenty word classes from four different people placed in front of NAO, roughly one meter away.

Except for the *Kitchen* classes, each class has 20 instances which made a total number of 852 sounds recorded for the whole dataset. Considering that detection step is not addressed in this study, each sound has been manually segmented using an audio editor. As an illustration of the signals “quality”, Fig. 2 shows the signal-to-noise ratio (SNR) statistics for each class, the noise being here the internal noise, measured during absence of any external sound.

TABLE I  
TAXONOMY OF THE RECORDED DATA SET CLASSES.

Scenarios	Taxonomy	Classes
Kitchen	“Mouth” sound	<i>Eating, Choking</i>
	Cooking	<i>Cutlery, Fill a glass, Running the tap</i>
	Moving	<i>Open/close a drawer, Move a chair</i>
	Alarms	<i>Open microwave, Close microwave</i>
Office	Door	<i>Close, Open, Key, Knock</i>
	Others	<i>Ripped Paper, Zip, (another) Zip</i>
Nonverbal		<i>Fingerclap, Handclap, Tongue Clic</i>
Speech	Numbers	<i>1,2,3,4,5,6,7,8,9,10</i>
	Orders	<i>Hello, Left, Right, Turn, Move</i> <i>Stop, Nao, Yes, No, What</i>

### III. AUDIO FEATURE REPRESENTATIONS

In this section, we present the different signal representations that were tested in our classification benchmark. Although quite short (see introduction), the considered signals are generally non-stationary, hence most of the features are actually *time sequences* of *feature vectors* computed using the very usual short-term sliding window approach widely

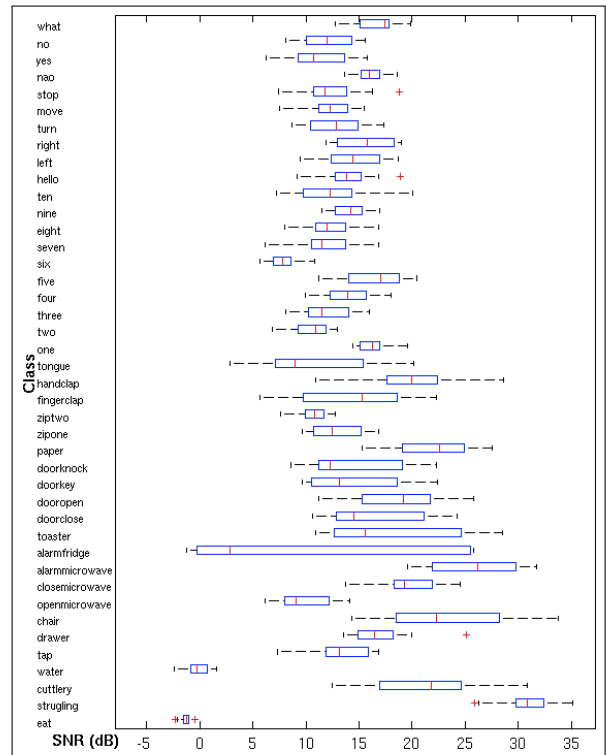


Fig. 2. Signal-to-Noise Ratio (SNR) per class. For each box, the central red mark denotes the median, the edges the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted with a cross.

used in audio processing. Except when specified, the window analysis is a 30ms Hamming window with 50% overlap. All the features introduced in this section have been proposed in the audio processing literature [18].

#### A. Time-Domain Features

1) *Energy*: We compute the energy as the root mean square of the samples in an audio frame (the rectangular window is used here). It can be seen a measure of amplitude variation over time.

2) *Zero Crossing Rate (ZCR)*: defined as the number of zero crossings in an audio frame. It can be used to classify voiced and unvoiced speech sounds, and it has also been used to differentiate speech, music and background noise [19].

3) *Sound Duration*: This feature is the total duration of the detected sound expressed in seconds. Therefore, in addition to being a scalar value, it is the only feature that is not extracted on a short-time basis. It may help to distinguish short, e.g. percussive, sounds from longer ones.

#### B. Frequency-Domain Features

All these features are computed using the Short-Term Fourier Transform (STFT) of the signal.  $S(t, k)$  denotes the  $k$ -th magnitude coefficient of the  $N$ -point STFT frame at time  $t$ .

1) *Spectral Roll-Off*: The Spectral Roll-off is the cut-off frequency below which 99% the spectral energy is contained. It is used in speech recognition to classify voiced and unvoiced speech [20].

2) *Spectral Shape Statistics*: Those features characterize the overall shape of the spectrum using  $n$ -order moments of frequency bin weighted by spectral magnitude:  $\mu_n(t) = \sum_{k=0}^{N-1} k^n S(t, k) / \sum_{k=0}^{N-1} S(t, k)$ . The first moment, or spectral centroid or brightness, corresponds to the mean value of the weighted frequency. The second order moment measures the spread of the frequency distribution around the mean. The third order moment, or skewness, is a measure of the asymmetry of the distribution. The kurtosis (fourth order moment) is a measure of the “peakedness” of the distribution.

3) *Spectral Slope and Spectral Decrease*: The two features represents the global amount of decreasing of the spectral amplitude. The spectral slope is estimated by linear regression.

$$S_{slope}(t) = \frac{N \sum_k f_t(k) S(t, k) - \sum_k f_t(k) \sum_k S(t, k)}{N \sum_k f_t(k)^2 - (\sum_k S(t, k))^2},$$

where  $f_k(t)$  represents the value of the linear regression at bin  $k$  (and at time  $t$ ). The formulation of the spectral decrease comes from perceptual studies and tries to be coherent with human hearing [18].

$$S_{decrease}(t) = \frac{1}{\sum_{k=1}^{N-1} S(t, k)} \sum_{k=1}^{N-1} \frac{S(t, k) - S(t, 0)}{k}.$$

4) *Spectral Flatness*: An estimation of the flatness of the magnitude spectrum is obtained by the ratio between its arithmetic and geometric mean (flat if  $\approx 1$  or peaky if  $\approx 0$ ):

$$S_{flat}(t) = \exp\left(\frac{1}{N} \sum_{k=0}^{N-1} \log(S(t, k))\right) / \frac{1}{N} \sum_{k=0}^{N-1} S(t, k).$$

5) *Spectral Flux and Spectral Correlation*: The two features measure the average variation of spectral coefficients between two consecutive frames:

$$S_{flux}(t) = \frac{\sum_k (S(t, k) - S(t-1, k))^2}{\sqrt{\sum_k S(t, k)^2} \sqrt{\sum_k S(t-1, k)^2}}$$

$$S_{cor}(t) = \frac{\sum_k S(t, k) S(t-1, k)}{\sqrt{\sum_k S(t, k)^2} \sqrt{\sum_k S(t-1, k)^2}}.$$

### C. Mel-Frequency Cepstral Coefficients

Widely used in speech and speaker recognition [12], MFCCs are cepstral coefficients that represent the spectrum envelope on a perceptive mel-frequency scale. Those coefficients are computed as the discrete cosine transform (DCT) of the logarithm of FFT power coefficients passed through a mel-filter bank (40 log-spaced bands in the range 300Hz-10000Hz according to the following mel-scale  $1127 \log(1 + f/700)$ ). Usually, the first coefficient is omitted and the first and second derivatives of the remaining coefficients can be added.

### D. Wavelet Features

The wavelet transform [21] transpose a signal from time domain to time-frequency domain like the STFT although the different family of basis functions, allowing multi-resolution analysis to get a variable time and frequency resolution. The discrete version of the transform [22] uses a  $M$ -stage cascade of a downsampling by 2 and a high-pass and low-pass filter. Thus, a signal  $x(n)$  can be decomposed on  $a_i(k)$  and  $d_i(k)$  with  $i = 1, \dots, M$  called respectively the approximation coefficients and the details coefficients. Inspired from [22] and [23], the feature vector is the concatenation of the mean and the standard deviation of the coefficients  $a_M$  and  $d_i$  with  $i = 1, \dots, M$ . The experiments use an 8th order decomposition on a 8-coefficient Daubechies family.

### E. Stabilized Auditory Images

Based on modelling of the human cochlea, the auditory image model (AIM) of [24] produces stabilized auditory images (SAI), which are a time delay-frequency sound representation close to a correlogram. The process chains three main stages, multi-channel gammatone filter bank, half-wave rectification and triggered time integration, and leads to a representation with high dimensionality. A technique was proposed in [25] to reduce the dimensionality of the SAI features. This procedure consists of three steps: create patches from the SAI, compute a low-dimensional vector representation of each patch, and concatenate these patch feature vectors to form the final feature vector.

### F. Post-processing

Depending on the feature nature, the successive feature vectors  $\mathbf{x}^t$  of a given sound can be further processed to produce different final features, which will feed the classifiers:

- The *sequencing* i.e. simple concatenation, of the (original) successive vectors  $\mathbf{x} = [\mathbf{x}^1, \dots, \mathbf{x}^T]$ .
- The *mean* of the vectors over the entire acoustic event. The concatenation of the mean and standard deviation can also be used.
- The *bag-of-words* (BoW) approach. The features of all sounds are first clustered using the  $K$ -means algorithm. Then, each sound has its feature vectors quantized using the resulting centroids, and is then represented as the *normalized histogram* of centroid occurrences.
- The *interpolation* of the feature vector sequence to the mean duration  $\bar{T}$  of all vector sequences in the database. Each sound is thus represented by  $\bar{T}$  interpolated feature vectors sequenced into  $\mathbf{x}_I = [\mathbf{x}_I^1, \dots, \mathbf{x}_I^{\bar{T}}]$ .

The interpolation enables to normalize the vector sequence along the time axis, so that the new representation can be used by “fixed data length” classifiers. It amounts to a simplified Dynamic Time Warping (DTW) applied “blindly”, i.e. without inspecting the fine structural organization of the

sounds. The bag-of-words also intrinsically enables (temporal) normalization but without taking into account the timeline ordering of the vector sequence.

Finally, it can also be noted that the final feature representation may also consist of the (row-wise) concatenation of several different features. This is a particular (straightforward) case of information fusion for classification, a vast domain which deepened investigation in the context of sound recognition by a robot is out of the scope of the present paper.

### G. Implementations

The computation of the wavelets has been done using the Matlab Wavelet Toolbox. SAI features are available at [26]. All other features have been computed with the Python/C++ toolbox YAAFE [27].

## IV. ISOLATED SOUND CLASSIFICATION

In this section all the tested classifiers are described. A multiclass classifier consists of a mapping  $g : \mathcal{X} \times \mathcal{C} \rightarrow \mathbb{R}$ , where  $\mathcal{X}$  is the feature space,  $\mathcal{C} = \{1, \dots, C\}$  is the set of labels and  $C$  is the number of classes. The dimension of  $\mathcal{X}$  may be fixed or varying with the sound, depending on the feature used. Given a feature vector (or sequence of feature vectors)  $\mathbf{x} \in \mathcal{X}$ ,  $g(\mathbf{x}; c)$  is the score of classifying  $\mathbf{x}$  as  $c$ . The higher the score is, the more likely  $c$  is the class of  $\mathbf{x}$ . Hence, a new unlabelled observation  $\mathbf{x} \in \mathcal{X}$  is classified as:

$$c^*(\mathbf{x}) = \arg \max_{c \in \mathcal{C}} g(\mathbf{x}; c).$$

In the following,  $\mathbf{X}$  will denote the training set, i.e. a set of feature vectors  $\mathbf{X} = \{\mathbf{x}^n\}_{n=1}^N$  which class is known, and that is used to train the classifiers.

### A. $K$ -Nearest Neighbors

The  $k$ -nearest neighbors ( $k$ -NN) classifier is based on the well-known  $k$ -NN algorithm which returns the subset of  $S_k(\mathbf{x}) \subset \mathbf{X}$ , containing the  $k$  closest points to a given vector  $\mathbf{x}$ . The mapping of the  $k$ -NN classifier is:  $g_{k\text{NN}}(\mathbf{x}, c) = |\{\tilde{\mathbf{x}} \in S_k(\mathbf{x}) | c(\tilde{\mathbf{x}}) = c\}|$ , where  $c(\tilde{\mathbf{x}})$  means the class of  $\tilde{\mathbf{x}}$ .  $g_{k\text{NN}}(\mathbf{x}, c)$  is the number of feature vectors among the  $k$ -nearest neighbors of  $\mathbf{x}$  that belong to the class  $c$ . In other words, each of the  $k$  neighbors votes for its own class, and the class with more votes is assigned to  $\mathbf{x}$ .

### B. Quantized Nearest Neighbor

The previous method needs to keep in memory all the training data during the recognition stage. QNN is able to circumvent this issue by quantizing the features previously to the nearest neighbor search. More precisely, the vectors are first divided in  $P$  parts, leading to  $P$  feature subspaces. If  $\mathbf{x}_{n,p}$  denotes the  $p$ -th part of the  $n$ -th training vector, we define  $\mathbf{X}_{p,p} = \{\mathbf{x}_{n,p}\}_{n=1}^N$ , the training set of the  $p$ -th feature subspace. A  $K$ -means algorithm [28] is ran for every  $\mathbf{X}_{p,p}$ , providing for a set of centroids. The quantization function,

that assigns the  $p$ -th subvector  $\mathbf{x}_{p,p}$  of  $\mathbf{x}$  to its closest centroid is denoted by  $Q_p(\mathbf{x}_{p,p})$ . In that case the mapping  $g$  is:

$$g_{\text{QNN}}(\mathbf{x}; c) = - \min_{\tilde{\mathbf{x}} \in \mathbf{X}_c} \left( \sum_{p=1}^P \|Q_p(\tilde{\mathbf{x}}_{p,p}) - Q_p(\mathbf{x}_{p,p})\| \right)^{\frac{1}{2}},$$

where  $\mathbf{X}_c = \{\mathbf{x} \in \mathbf{X} | c(\mathbf{x}) = c\}$ . This corresponds to finding the quantized vector in the training set closest to the quantized test vector, and assigning its class to  $\mathbf{x}$ . See [11] for more details on this technique. The method is parametrized by  $K$  and  $P$ . The higher  $K$  and  $P$  are, the more costly the method is, and the higher the recognition rate is. Increasing  $P$  may allow us to reduce  $K$  with no negative effects on the recognition rate.

### C. Support Vector Machines

The Support Vector Machines (SVM) is a discriminative binary classification method [28]. It has been used in sound recognition in multiple situations as in [15] and [29] with hierarchical structures or in [30] with 1-class SVMs. SVMs provides a discriminative function  $h(\mathbf{x})$ , learnt from a set of positive examples and a set of negative examples. The points satisfying  $h(\mathbf{x}) = 0$  form a hyperplane in the space induced by the kernel function  $k(\cdot, \cdot)$ .  $h(\mathbf{x}) > 0$  means that  $\mathbf{x}$  should be classified as positive and  $h(\mathbf{x}) < 0$  as negative. We refer the reader to [28] for details on the formulation. Importantly, a parameter  $\mathcal{Q}$  regulates the amount of allowed misclassification in the training set, such that SVMs deal with overlapping classes.

Since SVMs are binary classifiers, two strategies have been developed to use them in the multiclass task. On one hand the *one-versus-rest* (1vR), in which  $C$  different SVMs are trained, one per class. In that case the mapping  $g$  is defined as  $g_{1\text{vR}}(\mathbf{x}; c) = h_c(\mathbf{x})$  where  $h_c(\mathbf{x})$  is the discriminant function trained with  $\mathbf{X}_c$  and  $\mathbf{X} \setminus \mathbf{X}_c$ . On the other hand the *one-versus-one* (1v1) strategy, which corresponds to evaluate all possible binary classification problems with  $C$  classes. The classification mapping is then:  $g_{1\text{v1}}(\mathbf{x}; c) = |\{d \in \mathcal{C} | d \neq c, h_{c,d}(\mathbf{x}) > 0\}|$ , where  $h_{c,d}$  is the discriminant function trained with  $\mathbf{X}_c$  and  $\mathbf{X}_d$ . As for  $k$ -NNs, this is equivalent to say that each SVM is voting for one class and  $\mathbf{x}$  is classified to the class with more votes. In our experiments, the 1V1 approach always outperformed 1VR both in terms of accuracy and speed, and we only consider 1V1 in the following.

Five different kernels are tested, namely: linear  $k_L(\mathbf{x}, \mathbf{y}) = \mathbf{x}^t \mathbf{y}$ , polynomial  $k_P(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^t \mathbf{y} + c_0)^d$ , radial basis  $k_R(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$ , sigmoid  $k_S(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^t \mathbf{y} + c_0)$ , and  $k_{\chi^2}(\mathbf{x}, \mathbf{y}) = 1 - 2 \sum_{i=1}^M \frac{(x_i - y_i)^2}{x_i + y_i}$ ,  $M$  being the dimension of the features. The parameters of the SVMs are the misclassification regulation parameter  $\mathcal{Q}$ , the multiclass strategy, the kernel used and, if any, the kernel parameters.

#### D. Gaussian Mixture Models

The Gaussian Mixture Model (GMM) is a probabilistic generative model widely used in classification tasks. In our case, we use one GMM per sound class. Each GMM is a weighted sum of  $M$  Gaussian components (in this model, each observation is assumed to be generated by one of these components), which parameter set denoted by  $\lambda_c$  is composed of  $M$  weights, mean vectors and covariance matrices. We learn  $C$  sets of parameters  $\lambda_c$ ,  $C$  being the number of classes using the well-known Expectation-Maximization (EM) algorithm. The mapping  $g$  corresponds to the likelihood of the observed data given the model parameters. For a sequence of feature vectors  $\mathbf{x} = [\mathbf{x}^1, \dots, \mathbf{x}^T]$ , which are assumed to be independent, we have:  $g_{\text{GMM}}(\mathbf{x}; c) = p(\mathbf{x}|\lambda_c) = \prod_{t=1}^T p(\mathbf{x}^t|\lambda_c)$ . This method is parametrized by the number  $M$  of Gaussians in the mixture, the maximum number of EM iterations and the shape of the covariance matrices (full or diagonal). We refer the reader to [28] for more details about GMM.

#### E. Hidden Markov Models

The Hidden Markov Models (HMM) also belong to the family of generative models [28], [12]. In a HMM the observations depend on a hidden discrete random variable usually called state, taking values from 1 to  $S$ . The probability of the observations given the state value is called emission probability. The state is assumed to be Markovian, that is, the state at time  $t$  only depends on the state at time  $t - 1$ . In addition, the states are constrained to happen in order, i.e. state  $s$  before the state  $s + 1$ ; this is usually known as *left-to-right* HMM. The emission probability is usually Gaussian or GMM. As in the case of GMM, one model  $\xi_c$  per class is learnt (through an EM algorithm). The model consists of the parameters of the emission probability and the parameters modeling the markovian dynamics. The function  $g$  is also the likelihood of the observations given the model:  $g_{\text{HMM}}(\mathbf{x}; c) = p(\mathbf{x}|\xi_c)$ . The parameters of the HMM are the parameters of the emission probability, the number of states  $S$ . We refer the reader to [12] for more details about HMM.

#### F. Implementations

The  $k$ -NN, GMM algorithms comes from the Matlab toolboxes. The QNN algorithm is our own Matlab code inspired from [11]. The HMMs are developed using the machine learning PMTK3 library [31]. The SVMs are implemented using libSVM [32].

### V. EXPERIMENTS

Given the database described in Section II, a large set of combinations of feature types, features post-processing, and classifiers have been tested (note that all combinations do not make sense, e.g., some features are not appropriate for time interpolation; we implemented only relevant combinations). In order to be able to statistically compare the different sound

recognition methods, we perform  $k$ -fold cross-validation repeated on  $n$  different runs. The results are averaged on these  $n$  runs, with  $k$  and  $n$  being set to 10.

Tables II to V gather the different statistics on the different combinations of features (rows) + post-processing, and classifiers (columns). *GMM-1* stands for the GMM method (section IV-D) applied when  $T = 1$ , while *GMM-T* corresponds to  $T > 1$ . It is important to note that GMM-T and HMM methods are fed with sounds represented by the original (variable-length) sequence of feature vectors, whereas all the other classifiers are fed with a single fixed-size vector representation issued from post-processing by either mean (rows 1–4), Bag-of-Words (rows 5 and 6), or fixed-sized interpolation (rows 7 and 8). The latter still represents a vector time-sequence but of fixed length, and hence can be reshaped in a single vector. *TTF* stands for Time and Time-Frequency Features (corresponding to the features of section III-A and III-B). Cells filled with gray correspond to irrelevant combinations.

#### A. Results

Note first that the best results using *TTF* or by concatenating *TTF*+MFCC have been found using the features *Energy*, *ZCR*, *Spectral Decrease*, *Spectral Flatness*, *Spectral Slope*. Therefore these features have been used in the presented results. Adding the *Roll-off* and the *Spectral Moments* gives similar results. The *Sound Duration* is not a reliable feature in the present context, since it lead to drop in scores.

As for accuracy, the best results are obtained using SVM classifiers on interpolated MFCC+TTF coefficients (97% accuracy), followed by  $k$ -NN with interpolated MFCC (96.2%). HMM on MFCC coefficients, which is a very usual combination in the literature, provides a very good baseline at 92.6% good accuracy. Therefore, a major result here is that, for short pre-segmented domestic sound recognition, a quite simple technique such as  $k$ -NN, that requires no training, can perform better than ASR reference methods such as HMMs. The latter requires both training and much longer decoding time (see Table IV) and may be more appropriate for long and complex sound sequences such as speech signals. As could be predicted, the preservation of dynamic information is important for accurate recognition: see the 96.2% good accuracy for  $k$ -NN with MFCC + interpolation vs. 87.4% for  $k$ -NN with MFCC + mean; see also the difference between GMM-1 and GMM-T. This is confirmed by the poor results obtained with the Bag-of-Words approach which has not proven being relevant in these experiments (remind that BoW histograms cumulate information over frames but loose the temporal structure; also, the histogram codebook cannot be large because the training time grows up exponentially with  $K$ : for the experiments, we used  $K = 50$ ). However, accurate vector alignment using advanced DTW as used in HMMs do not seem as crucial as for ASR: here basic fixed-size interpolation seems efficient enough for the task at hand. This rises many questions about the (temporal and spectral)

structure of domestic sounds, that go beyond the scope of the present study. Waiting for further investigations, the fact that  $k$ -NN with simple feature sequence interpolation outperforms HMMs (and GMM-T) can be partly explained by the fact that  $k$ -NNs use original data in the recognition task while HMMs (and GMM-T) use data models. In addition  $k$ -NN is a discriminative technique, whereas HMMs (and GMM-T) are generative models. A consequence is that  $k$ -NNs have a very large memory requirement to store the prototypes (see Table V), which is a major drawback for autonomous robotics.

Obviously, SVM is an interesting alternative, modestly increasing the recognition time over  $k$ -NN for a much smaller memory cost. And so is QNN which has a larger recognition time but an even smaller memory cost. Therefore, the choice between  $k$ -NNs, SVM and QNN should depend on the specifications of the autonomous robot in terms of computation and memory resources. GMM-T with MFCC has good accuracy performance but the recognition time is quite high, making it less interesting than the above-mentioned methods. It remains unclear why SVMs perform significantly better with MFCC+TTFF+interpolation than with MFCC+interpolation, whereas the difference is not so pronounced for  $k$ -NN and QNN (and for some other settings, adding TTFF even decreases the accuracy scores; this is difficult to explain, except appealing to the redundancy between some TTFF features and MFCC information). Anyway, a major point that arises from this study is that, for short domestic sounds recognition, the three methods  $k$ -NNs, SVM and QNN, combined with simple time interpolation of features, seem preferable to the (more complex) HMMs widely used for speech recognition and recently extended to the more general problem of sound scene analysis.

To complement those results, we present in Table VI, the time to compute feature vector(s) from a sound (mean or sequence; column *Feature*). In the column *BoW*, *K-means* is the training time of the codebook, and *Histo* the time to transform the feature vector(s) of one sound into a histogram. *Interpolation* is the time to perform the fixed-length time interpolation on the feature vector(s) of one sound. We can see that the time to compute the feature vector (sequence) is reasonable but not negligible: for example, it is an order of magnitude larger than the recognition time of  $k$ -NN, but it is also more than an order of magnitude lower than the recognition time for HMMs. For MFCC coefficients, the time needed to interpolate the MFCC sequence is comparable to the time needed to calculate the coefficients. Note that the memory cost for training the models from data are not considered in the present study, since this can be processed offline.

## VI. CONCLUSIONS AND FUTURE WORK

We addressed the problem of sound recognition by an autonomous humanoid robot, by benchmarking a large set of audio feature representations, post-processing, and classification techniques. A major result of this work is that, for the

42 classes of kitchen/office/voice sounds that we considered, very good accuracy scores (larger than 92% and up to 97%) were obtained for three techniques of very reasonable complexity (at least for decoding), namely  $k$ -NN, SVM and QNN. Moreover these methods were applied successfully on fixed-size sequences of MFCC vectors obtained with very simple DTW (fixed-size interpolation). The performance in accuracy is of the same order and even outperforms the performance of HMMs applied on the original vector sequences, whereas the decoding time (hence computational cost) is much lower. Therefore, these three methods seem to be appropriate within the context of a robotic implementation.

A more thorough analysis of the nature of domestic sounds must be carried out to reveal if they are characterized by an inner structure, in a similar way as, e.g. speech signals are characterized by successive phonemes (and transitions between them). Domestic/environmental sounds can also be analyzed in terms of taxonomy, nature (matter of the object that generated the sound: metal, wood, glass, etc.), interactions or dynamics (friction, shock, etc.). To reach this goal, the number of classes must be increased radically to reach several hundreds. The introduction of a “garbage class” is absolutely necessary, since, it is impossible to consider all the possible sound categories. Future work will also consider the processing of continuous audio streams, e.g., taking into account stationary and less stationary background noise, or “longer” sounds indicating a specific activity (e.g., tap water flushing). In addition to external noise, we will address the problem of ego-noise (generated by robot joints in motion) detection and removal, as in [33]. In the long run, we aim at merging the sound recognition system in a complete framework for acoustic scene analysis including source localization and separation, embedded in the robot NAO.

TABLE II  
ACCURACY RATES (IN %).

	$k$ NN	QNN	GMM-1	GMM-T	HMM	SVM
TTFF	65.9	62.8	67	71		74.3
MFCC	87.4	82.1	89.5	95.4	<b>92.8</b>	91.5
MFCC+TTFF	88.4	77.7	76.4	88	92.2	91.3
Wavelets	60.5	58.4	63	36.4	61.3	57
MFCC+BoW	55.8	53.9	45.2			52.6
MFCC+TTFF+BoW	60	55.8	41.1			62.5
MFCC+Interp	<b>96.2</b>	95.7				92.3
MFCC+TTFF+Interp	94.1	94.2				<b>97</b>
SAI	83	80				87

## REFERENCES

- [1] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT Press, 1994.
- [2] J. Huang, T. Supaongprapa, I. Terakura, F. Wang, N. Ohnishi, and N. Sugie, “A model-based sound localization system and its application to robot navigation,” *Robotics and Autonomous Systems*, vol. 27, no. 4, pp. 199–209, 1999.
- [3] K. Nakadai, H. G. Okuno, and H. Kitano, “Real-time sound source localization and separation for robot audition,” in *Int. Conf. on Spoken Language Processing*, 2002, pp. 193–196.



TABLE III  
TRAINING TIME (IN S).

	kNN	QNN	GMM-1	GMM-T	HMM	SVM
TFFF		0.6	0.3	10.8		0.076
MFCC		1.1	0.4	9.3	14.2	0.150
MFCC+TFFF		0.6	0.350	7.6	24.8	0.092
Wavelets		1.3	1	7.6	52	0.065
MFCC+BoW		1.5	0.360			0.350
MFCC+TFFF+BoW		1.5	0.5			0.380
MFCC+Interp		7.7				2
MFCC+TFFF+Interp		8.4				2.3
SAI		27.4				0.79

TABLE IV  
RECOGNITION TIME (IN MS).

	kNN	QNN	GMM-1	GMM-T	HMM	SVM
TFFF	0.3	1	18	36.3		0.1
MFCC	0.3	1	19.7	45.8	89	0.2
MFCC+TFFF	0.3	1	20	31	92	0.2
Wavelets	0.1	1	2.1	3	10	0.1
MFCC+BoW	0.4	1.4	19			8.2
MFCC+TFFF+BoW	0.3	1.3	19			0.9
MFCC+Interp	1.5	9				2.3
MFCC+TFFF+Interp	1.5	9.4				2.5
SAI	1.2	5.2				2.2

TABLE V

MEMORY NEEDED TO STORE THE TRAINED CLASSIFIERS (IN KB).

	kNN	QNN	GMM-1	GMM-T	HMM	SVM
TFF	370	6	39	130		520
MFCC	430	6	50	180	1100	540
MFCC+TFF	460	4	46	200	1300	680
Wavelets	350	10	58	65	430	330
MFCC+BoW	38	11.2	10.6			271
MFCC+TFF+BoW	31	16.5	52.1			206
MFCC+Interp	5300	715				2100
MFCC+TFF+Interp	6100	967				3800
SAI	4230	593				5550

TABLE VI

FEATURE COMPUTATION TIME (IN MS).

	Feature	BoW		Interpolation
		K-means	Histo.	
TFFF	3			
MFCC	2.4	12.3	0.8	2.3
MFCC+TFFF	5.4	13.3	0.9	2.7
Wavelets	9.6			
SAI	350			

- [4] S. Yamamoto, K. Nakadai, M. Nakano, H. Tsujino, J.-M. Valin, K. Komatani, T. Ogata, and H. G. Okuno, "Real-time robot audition system that recognizes simultaneous speech in the real world," in *Int. Conf. on Intell. Rob. and Syst.*, 2006, pp. 5333–5338.
- [5] K. Nakadai, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "An open source software system for robot audition hark and its evaluation," in *Int. Conf. on Humanoid Robots*, 2008, pp. 561–566.
- [6] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *Int. Conf. on Intell. Rob. and Syst.*, 2002, pp. 2478–2483.
- [7] S. Chu, S. Narayanan, C.-C. Kuo, and M. J. Mataric, "Where am I? scene recognition for mobile robots using audio features," in *Int. Conf. on Multimedia and Expo*, 2006, pp. 885–888.
- [8] Y. Sasaki, M. Kaneyoshi, S. Kagami, H. Mizoguchi, and T. Enomoto, "Daily sound recognition using pitch-cluster-maps for mobile robot audition," in *Int. Conf. on Intell. Rob. and Syst.*, 2009, pp. 2724–2729.
- [9] N. Yamakawa, T. Takahashi, T. Kitahara, T. Ogata, and H. G. Okuno, "Environmental sound recognition for robot audition using matching-pursuit," in *Modern Approaches in Applied Intelligence*, ser. Lecture Notes in Computer Science. Springer, 2011, pp. 1–10.
- [10] J. Stork, L. Spinello, J. Silva, and K. Arras, "Audio-based human activity recognition using non-markovian ensemble voting," in *International Symp. on Robots and Human Interactive Communications*, 2012, pp. 509–514.
- [11] M. Janvier, X. Alameda-Pineda, L. Girin, and R. P. Horaud, "Sound-event recognition with a companion humanoid," in *IEEE Int. Conf. on Humanoid Robotics*, 2012.
- [12] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [13] H. D. Tran and H. Li, "Sound event recognition with probabilistic distance SVMs," *IEEE Transactions on Speech Audio Processing*, vol. 19, no. 6, pp. 1556–1568, 2011.
- [14] Y. Toyoda, J. Huang, S. Ding, and Y. Liu, "Environmental sound recognition by multilayered Neural Networks," in *Int. Conf. on Computer and Information Technology*, 2004, pp. 123–127.
- [15] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 209–215, 2003.
- [16] V. Ramasubramanian, R. Karthik, S. Thiyagarajan, and S. Cherla, "Continuous audio analytics by HMM and Viterbi decoding," in *Int. Conf. Acoust., Speech, Sig. Process.*, 2011, pp. 2396–2399.
- [17] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition," *Patt. Rec. Lett.*, vol. 24, no. 15, pp. 2895–2907, 2003.
- [18] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidoado project," 2004.
- [19] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Int. Conf. Acoust., Speech, Sig. Process.*, 1996, pp. 993–996.
- [20] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Int. Conf. Acoust., Speech, Sig. Process.*, 1997, pp. 1331–1334.
- [21] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.
- [22] C. Lin, S. Chen, T. Truong, and Y. Chang, "Audio classification and categorization based on wavelets and support vector machine," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 644–651, 2005.
- [23] G. Tzanetakis, G. Essl, and P. Cook, "Audio analysis using the discrete wavelet transform," in *Conf. in Acoust. and Music Theory App.*, 2001.
- [24] T. C. Walter, "Auditory-based processing of communication sounds," Ph.D. dissertation, University of Cambridge, 2011.
- [25] 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.
- [26] T. Walters and W. van Engen. (2012) AIMC: A C++ implementation of the auditory image model. [Online]. Available: <https://code.google.com/p/aimc/>
- [27] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software," in *Int. Conf. for Music Information Retrieval (ISMIR)*, 2010.
- [28] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [29] A. Temko and C. Nadeu, "Classification of acoustic events using SVM-based clustering schemes," *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.
- [30] R. Asma, K. Hachem, L. Zied, and E. Noureddine, "One-class SVMs challenges in audio detection and classification applications," *EURASIP Journal on Advances in Signal Processing*, 2008.
- [31] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT Press, 2012.
- [32] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [33] A. Ito, T. Kanayama, M. Suzuki, and S. Makino, "Internal noise suppression for speech recognition by small robots," in *European Conf. on Speech Communication and Technology*, 2005.