2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance

Audio Surveillance Using a Bag of Aural Words Classifier

Vincenzo Carletti, Pasquale Foggia, Gennaro Percannella, Alessia Saggese, Nicola Strisciuglio, Mario Vento
University of Salerno
Department of Information Engineering, Electrical Engineering and Applied Mathematics (DIEM)
Via Giovanni Paolo II, 132 - 84084 Fisciano (SA) - Italy
{vcarletti, pfoggia, pergen, asaggese, nstrisciuglio, mvento}@unisa.it

Abstract

In this paper we propose a novel approach for the audio-based detection of events. The approach adopts the bag of words *paradigm, and has two main advantages over other techniques present in the literature: the ability to automatically adapt (through a learning phase) to both short, impulsive sounds and long, sustained ones, and the ability to work in noisy environments where the sounds of interest are superimposed to background sounds possibly having similar characteristics.*

The proposed method has been experimentally validated on a large database of sounds, including several kinds of background noise, which are superimposed to the sounds to be recognized. The obtained performance has been compared with the results of another audio event detection algorithm from the literature, showing a significant improvement.

1. Introduction

While audio analysis has been traditionally focused mostly on the recognition of the speech and on the identification of the speaker, in the recent years several researchers have proposed audio-based systems for the automatic detection of abnormal or dangerous events. Such systems can be an inexpensive addition to existing video surveillance infrastructures, where video analytics solutions based on object tracking algorithms [**?**] are often used; in fact, many IP cameras are already predisposed to connect to a microphone, making available an audio stream together with the video stream. But audio event detection can be useful on its own, for example in contexts where video information is not feasible (e.g. large unlit areas at night, or environments with too many obstacles on the line of sight). Finally, there are some events that have a very distinctive audio signature, but are not so easy to spot on a video: for instance, a gunshot, or a person screaming. For these reasons, in the recent years the research community has shown a growing interest towards these applications.

One of the open problems in the design and implementation of a reliable and general audio event detector is that the properties characterizing the different events of interest might be evident at very diverse time scales: compare, for instance, an impulsive sound like a gunshot with a sustained sound, like a scream, that can have a duration of several seconds; so it is not easy to find a set of features that can accomodate both kinds of situations. Also, in real applications there is often the problem that the sounds of interest are superimposed to a significant level of background sounds; thus it might be difficult to separate the noise to be ignored from the useful sounds to be recognized.

In [2] Clavel et al. propose a method for gunshot detection, that operates by dividing the audio stream into 20 milliseconds frames, and computing for each frame a vector with such features as short-time energy, Mel-Frequency Cepstral Coefficients (MFCC) and spectral statistical moments. The vectors are classified using a Gaussian Mixture Model (GMM). Then, the final decision is taken over 0.5 seconds intervals using a Maximum A Posteriori (MAP) decision rule. Vacher et al. in [12] also adopt a GMM classifier, with wavelet-based cepstral coefficients as features, for the detection of screams and broken glass. Rouas et al. [10] use MFCC features and a combination of the GMM and Support Vector Machine (SVM) classifiers for detecting screams in outdoor environments. Their method uses an adaptive thresholding on sound intensity for limiting the number of false detections. Gerosa, Valenzise et al. [4, 13] propose a system for the detection of gunshots and screams which specifically address the ambient noise problem. Their method uses two parallel GMM classifiers trained to separate screams from noise and gunshots from noise. The paper by Ntalampiras et al. [8] proposes a two stage classifier: the first stage is used to discriminate between vocalic sounds (such as screams and normal speech) and impulsive sounds (such as gunshots or explosions); then specific second stage GMM classifiers are activated, using different features for the two kinds of sounds, to provide

978-1-4799-0703-8/13/$31.00 ©2013 IEEE

the final classification of the sound. Conte et al. [3] present a method with two classifiers that operate at different time scales; the method uses a quantitative estimation of the reliability of each classification to combine the classifier decisions and to reduce the false detections by rejecting the classifications that are not considered sufficiently reliable. Chin and Burred [1] propose a system in which the audio is represented as a sequence of symbols, each corresponding to a spectral shape observed over a 10 millisecond window. To these sequences, they apply Genetic Motif Discovery, a technique introduced for the analysis of gene sequences, in order to discover subsequences that can be used to recognize the audio events of interest. The algorithm is able to consider subsequences of different lengths for different classes, and the subsequences may contain *wildcard* elements that can be used to skip variable symbols due to the background noise.

In this paper we present an audio event detection system that is based on the *bag of words* approach, commonly used for the categorization of textual documents [6], and recently applied with success to video-based object detection and other similar problems [11]. The proposed method uses a two level description: first-level features are computed on a very short time interval, and are somewhat analogous to the words of a text; second-level features characterize a longer time interval, and are constructed by means of a learning process, on the basis of the actual sounds to be recognized. Finally, a classifier is trained on second level features, so as to learn which of them are significant for the recognition of a particular event class and which ones are irrelevant. This architecture is thus able to work on a longer time scale, but still remains able to give the right weight to short relevant sounds; furthermore, the presence of background noise has a reduced impact because the classifier can learn to ignore the second level features that are due to the background.

2. The proposed method

The system described in this paper is devoted to audio event detection; given M classes of sounds of interest $C_1, \ldots, C_M$, each represented by a finite set of examples, and an audio stream, the goal of the system is to find if (and where) there are occurrences of the sounds of interest within the stream. The audio stream usually contains other sounds not belonging to the classes of interest, that are considered as *background* sounds; we will indicate as C_0 the class containing all the background sounds.

In the *bag of words* approach, the datum to be classified is represented by detecting the occurrence of local, low-level features (*words*) and constructing a vector whose dimensionality corresponds to the number of possible words, and whose elements are indicators of the presence of the corresponding words, or a count of their occurrences. For instance, in text characterization the low-level features are the actual natural language words of a document (after removing suffixes, articles etc.), and the whole document is represented by a (high dimensional) vector of word occurrences; such vectors are then classified using traditional Pattern Recognition tools.

For the extension of these approaches to Computer Vision, the words are replaced either by small fixed-size image patches, or by salient points (e.g. SIFT features). Since the space of the possible words is huge (theoretically infinite), a quantization is performed using a training set; the result is a *codebook* that allows to associate each low-level feature with one word chosen from a finite set.

The overall architecture of the proposed method is shown in Fig. 1. The K-Means clustering (blue box) is used only during the training phase of the system, while the other modules (green boxes) of the pipeline are used both in the training and in the test phases. Each phase of the proposed method is explained in detail in the following.

2.1. First-level features

The input audio stream, sampled at a rate F_s, is divided in groups of N partially overlapped frames, with L PCM samples per frame. Every frame is built by forward shifting the previous one of $L/4$ samples.

For each frame, a first-level feature vector is computed. In particular, we have considered a set of 11 features from the literature on audio event detection belonging to the category of the *spectral features*, namely spectral centroid, spectral spread, spectral rolloff, spectral flux [5, 9], of the *energy features*, namely total energy, sub-band energy ratios (for 4 sub-bands), volume [5, 7], of the *instantaneous temporal feature*, namely the zero-crossing rate [5, 9]. Following, we report a brief description of the first-level features.

2.1.1 Spectral centroid and spectral spread

In digital signal processing, the spectral centroid (SC) and the spectral spread (SS) are measures for characterising the distribution of the frequency components of a signal. The spectral centroid is defined as the *"center of mass"* of the spectrum and is computed as follows:

$$SC = \frac{\sum_{i=1}^{L_F} i \frac{F_s}{L_F} |X(i)|}{\sum_{i=1}^{L_F} |X(i)|}, \qquad (1)$$

while the spectral spread is computed as the dispersion of the frequency components of the signal around the centroid:

$$SS = \sqrt{\frac{\sum_{i=1}^{L_F} \left[i \frac{F_s}{L_F} - SC \right]^2 |X(i)|}{\sum_{i=1}^{L_F} |X(i)|}}, \qquad (2)$$

where L_F and $|X(k)|$ are the length and the module of the *FFT* of the imput signal $x(n)$, respectively.

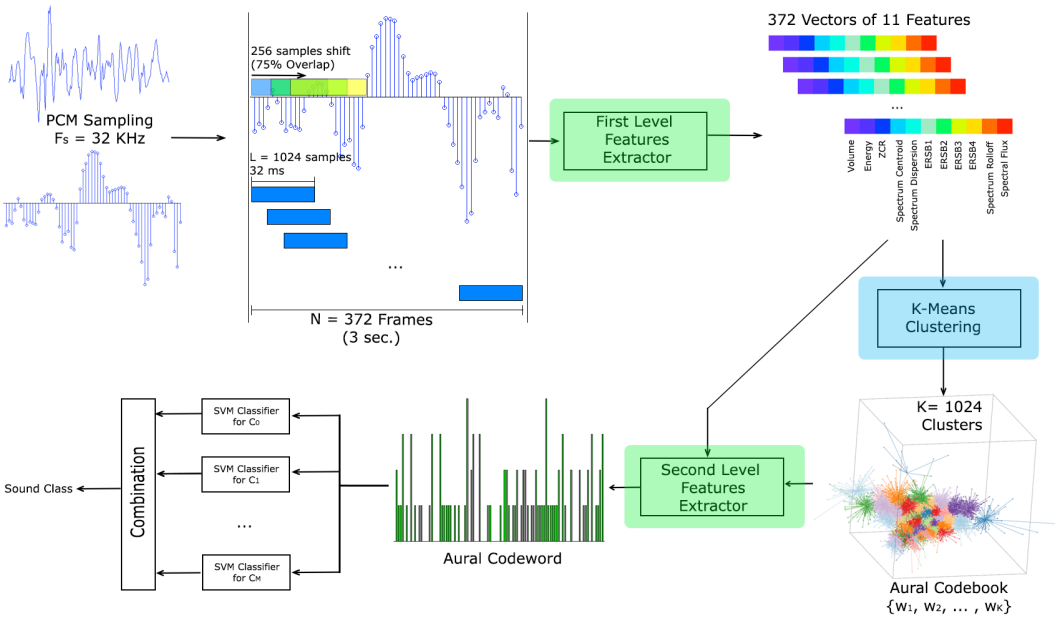

Figure 1: System architecture of the proposed method. The modules used in both the training and the operative phases are shown in green, while the blue module is used only during the training phase. The values of the parameters F_s, L, N and K used for the experimental validation are also reported.

2.1.2 Spectral rolloff

The spectral rolloff is a measure of the skewness of the spectrum and is defined as the frequency f_{ro} at which the $P\%$ of the spectral components of the signal is at lower frequency. In our case, we consider $P = 90$ and determine the value f_{ro} from the following relation:

$$\sum_{i=1}^{f_{ro}} |X(i)| = \frac{P}{100} \sum_{i=1}^{F_{max}} |X(i)| \,. \tag{3}$$

2.1.3 Spectral flux

The spectral flux (SF) indicates how quickly the spectral information of a signal is changing and it is computed by considering the squared-difference between the spectra of two consecutive audio frames, as reported in the following equation:

$$SF = \sum_{i=1}^{L_F} [X_n(i) - X_{n-1}(i)]^2 \,. \tag{4}$$

2.1.4 Energy ratios in sub-bands

The energy ratios in sub-bands ($ERSB$) give a rough approximation of the energy distribution of the spectrum. We divided the spectrum of the signal into four sub-bands, which are reported in Eq. 6, and for each sub-band we computed the ratio between the energy contained in that sub-band and the overall energy of the audio frame.

$$ERSB_n = \frac{\sum_{i=k_{n_1}}^{k_{n_2}} |X(i)|^2}{\sum_{i=1}^{F_{max}} |X(i)|^2}, \tag{5}$$

where

$$[k_{n_1}, k_{n_2}] = \begin{cases} [1, 630], & n = 1 \\ [631, 1720], & n = 2 \\ [1721, 4400], & n = 3 \\ [4401, 22000], & n = 4 \end{cases}. \tag{6}$$

2.1.5 Volume and energy

We calculate the volume feature (V) as the root mean square (RMS) of the amplitude value of the samples in an audio frame:

$$V = \sqrt{\frac{1}{L} \sum_{i=1}^{L} x(i)^2}, \tag{7}$$

while the energy (E) is the squared-sum of the amplitude value of the audio samples:

$$E = \sum_{i=1}^{L} x(i)^2. \tag{8}$$

2.1.6 Zero crossing rate

The zero crossing rate (ZCR) is the rate of the sign-changes along a frame and is especially used to characterise percussive sounds and environmental noise. For a frame $x(i)$ of L samples, the ZCR is computed as follows:

$$ZCR = \frac{1}{2L}\sum_{i=1}^{L}|sgn(x(i+1)) - sgn(x(i))| \quad (9)$$

2.2. Second-level features (Aural words)

In order to derive a finite set of *aural words* that play the role of the words in a textual document, we have performed a quantization of the vector space of the first-level features using the well known K-Means clustering algorithm during the training phase of the system. Since this algorithm requires as a parameter the desired number of clusters K, a grid search is conducted to find the value that maximizes the final classification accuracy.

It is worth noting that the method only requires unlabeled samples for performing the clustering; thus for the training set it is not necessary to have a ground truth with a granularity of a single frame; this can be a significant advantage over other methods, greatly reducing the human labor time required to train the event detection system on a new set of sounds.

The output of the K-Means algorithm is the set of cluster centroids, which constitutes the *codebook* of the system:

$$CB = (w_1, \ldots, w_K) \quad (10)$$

Conceptually, the entries w_i in the codebook can be thought as the elementary words that can be detected in the input data to be classified; we call them *aural words*, to emphasize the fact that they are related to atomic, perceptual units of hearing, and not to linguistic units.

In the same way as the topic of a document cannot be inferred from a single word, but for a larger body of text it can be reasonably estimated in many cases by considering the presence or the absence of a certain number of relevant words, we assume that a single aural word is not sufficient to classify a sound event, but the presence or the absence of certain specific words over a longer time interval may lead to a reliable classification. Thus, in order to perform the classification, we build a second-level feature vector as follows: first, the input is divided into (partially overlapping) *intervals* of N frames, so that an interval covers a time scale sufficient to recognize also sustained sounds (in our experiments we have used 3 seconds intervals, chosen by testing several values of N and selecting the one providing the best results). For each frame of the interval, the first-level feature vector v_i is computed (with $i = 1, \ldots, N$).

Then, for each frame, the codebook is searched for the word that is closest to v_i; let us denote as b_i the index of this word:

$$b_i = \arg\min_j |v_i - w_j| \text{ for } j \in \{1, \ldots, K\} \quad (11)$$

Finally, the second-level feature vector $U = (u_1, \ldots, u_K)$ is defined as follows:

$$u_j = \sum_{i=1}^{N} \delta(b_i, j) \text{ for } j \in \{1, \ldots, K\} \quad (12)$$

where $\delta(\cdot)$ is the Kronecker delta.

Thus, the second-level feature vector is the histogram of the occurrences of the aural words detected in the interval as shown in Fig. 1.

2.3. The classifier

The second-level feature vectors are used to train a Support Vector Machine (SVM) classifier, using a labeled training set, with a ground truth defined at a time scale corresponding to an interval.

The choice of the SVM classifier is motivated by the ability of this algorithm to find a hyperplane separating the classes to be recognized that is maximally stable, in the sense that it maximizes the margin between the decision boundary and the training samples, so as to avoid overfitting on small training sets. We have used the original, linear version of the SVM, and not the kernelized one, since it provided satisfactory results in our experiments.

The SVM (like other classifiers based on discriminant analysis, but differently from distance-based classifiers like the Nearest Neighbor) is able to construct a decision function that gives only a subset of the input features a non zero weight; in this way it can learn which are the aural words that are really discriminants for the events of interest, and ignore the others.

Since SVM is a binary classifier (i.e. it works on a two-classes problem), our system is organized so as to have several SVM instances operating in parallel, as shown in Fig. 1. Namely, we have $M + 1$ classifiers (where M is the number of the classes to be recognized); classifier i (with $i = 0, \ldots, M$) is trained using as positive examples the samples from class C_i and as negative examples all the other classes. Given a second-level feature vector, the classifiers produce an output and a score; if at least one classifier yields a positive output, the vector is assigned to the class with the maximum score (which might be the background class C_0, and so no event is reported); if all classifiers give a negative output, the vector is assigned to C_0.

3. Experimental results

3.1. The dataset

The experimental validation of the system has been carried out considering a typical audio surveillance application

that involves the recognition of the following three classes of abnormal audio events: scream, broken glass and gunshot. From the user perspective the system should raise an alarm only in presence of an abnormal audio event, while the sounds produced by any other source should be assigned to the *background noise* class.

Since at the best of our knowledge, there are no public datasets available for benchmarking purposes, we build a dataset with audio clips sampled at 32 KHz and 16 bits of resolution. The dataset contains sound samples collected from the Internet and other sounds that we recorded in different environment conditions. In particular, in our database we included 278 audio clips belonging to the three classes of abnormal sounds defined above. Furthermore, in order to test the robustness of our system with respect to the presence of other typical background sounds from indoor and outdoor environments, we inserted in the dataset also 173 clips representative of the following classes: rain, whistles, child cheerings, crowds, vehicles, household appliances, applauses, bells.

Other key requisites for an audio surveillance system are the ability of detecting the events of interest (hereinafter foreground sounds) when the source is at different distances from the microphone and when the relevant sound is overlapped to one or more background noises. Thus, in order to account for this variability we defined and adopted a procedure for creating a new dataset of audio clips obtained by mixing foreground sounds with background noises. Specifically, all the sounds in our datasets were first normalized to the same reference amplitude value. Then, each audio sample $\hat{f}$ of the new dataset was obtained as:

$$\hat{f} = \alpha \cdot f + \sum_{j=1}^{r} \beta_j \cdot b_j \tag{13}$$

where α and β_j are random variables uniformely distributed in the ranges $[0.25,\ 1.00]$ and $[0,\ 0.15]$, respectively; f and b_j are normalized audio samples belonging to the foreground and background classes, both randomly chosen, and r is randomly chosen in the interval $[1,\ 3]$. With the sum operator in equation 13 we represent the operator that allows to mix two signals by calculating the average value, while the product of the signal with a float scalar allows to represent the attenuation of the amplitude of the signal in order to simulate a sound source at certain distance from the microphone. It is worth pointing out that for the generation of the background noise samples for the new dataset, we did not consider the first addend in the previous equation.

The audio files in the final dataset are organized into four classes: background noise (BN), broken glass (BG), gunshot (GS) and scream (S). We partitioned the dataset into a training set and a test set composed by 1000 and 1500 audio clips for each class, respectively.

3.2. Performance evaluation

For the experimental evaluation of our system we used the following values of $F_s = 32kHz$, $L = L_F = 1024$, $N = 372$, $K = 1024$, $M = 3$. The accuracy obtained on the test set by the method is 95.8% confirming the validity of the proposal. Furthermore, the confusion matrix reported in Table 1 shows that all the considered classes were detected with accuracies above 96% with the only exception of the BG class whose accuracy is about 3% below. We can also note that most misclassification errors over the three foreground sounds (BG, GS, S) are directed to the BN, consequently being missed detections.

	BN	**BG**	**GS**	**S**
BN	0.961	0.032	0.004	0.003
BG	0.060	0.937	0.003	0.000
GS	0.021	0.011	0.968	0.000
S	0.030	0.005	0.000	0.965

Table 1: The confusion matrix of the proposed method over the test set. The entry at the row i and column j represents the fraction of samples belonging to the i-th class and attributed by the system to the j-th class.

It can be interesting also to focus on the performance indices computed by aggregating the foreground classes in a macro-class; such analysis is interesting from the final user perspective. In fact, the indices calculated in this way, and reported in Table 2, provide a more immediate insight of the reliability of the system in the detection of abnormal events. We notice again that the performance are very balanced among the two considered macro-classes as the confusion matrix is almost symmetric. We also obtain $Recall = 96.3\%$ and $Precision = 98.7\%$.

	BN	**BG + GS + S**
BN	0.961	0.039
BG + GS + S	0.037	0.963

Table 2: The confusion matrix of the proposed method over the test set when the three classes of interest related to abnormal events are reported as aggregated (BG + GS + S).

3.3. Performance comparison

In order to estimate the advantage introduced by the proposed architecture based on the bag of words approach, over a more classical architecture, we report here the performance obtained by a system using the same set of features adopted in this paper at the first level, but using a LVQ classifier. It has to be noted that this architecture is derived

by the one presented in [3] that already showed good performance on a similar problem.

The LVQ classifier was trained, using the same set of features, in order to discriminate among the four considered classes, i.e. the three foreground classes plus the background noise class. The considered system classifies audio chunks of 32ms (frames); the answers at the frame level are aggregated on a 3 seconds interval so that the system attributes the audio clip under test to the class C_i obtaining the highest score $z_i = (n_i - \hat{n}_i)/\hat{n}_i$, where n_i is the number of frames in the interval assigned by the LVQ classifier to C_i and $\hat{n}_i$ is a threshold that has to be passed by n_i in order to consider the i-th class as a candidate. Note also that in case $z_i < 0$ for $\forall i = 1, ..., M$ the sample is attributed to the background class C_0.

In our tests we determined the following optimal values of $\hat{n}_{BN} = 252$, $\hat{n}_{BG} = 120$, $\hat{n}_{GS} = 22$, $\hat{n}_S = 55$, from the analysis of the ROC curves over the training set. The LVQ based classification system reported an overall accuracy over the test set of 79.2% that is significantly below the accuracy obtained by the proposed method. The performance obtained on all the classes are in the confusion matrix in Table 3.

	BN	**BG**	**GS**	**S**
BN	0.773	0.103	0.099	0.025
BG	0.137	0.749	0.106	0.009
GS	0.081	0.128	0.785	0.007
S	0.026	0.034	0.078	0.862

Table 3: The confusion matrix of the system using the LVQ classifier over the test set.

4. Conclusions

In this paper we described a system for the detection of events of interest through audio recognition. The proposed method adopts the *bag of aural words* approach that is inspired by the classical bag of words approach used for textual document categorization. Although the metodology is rather general and can be adopted for the recognition of any type of audio event, the performance assessment described in this paper has been carried out on a significant dataset of audio samples referred to the domain of audio surveillance, where the aim is to raise an alarm only in presence of the sound produced by a scream, a broken glasses or a gunshot. The results obtained are very convincing thus confirming the feasibility of the approach, above all when the performance are compared with the results obtained adopting the same audio features with a LVQ classifier.

References

[1] M. Chin and J. Burred. Audio event detection based on layered symbolic sequence representations. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1953–1956, 2012.

[2] C. Clavel, T. Ehrette, and G. Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1306 –1309, 2005.

[3] D. Conte, P. Foggia, G. Percannella, A. Saggese, and M. Vento. An ensemble of rejecting classifiers for anomaly detection of audio events. In *Advanced Video and Signal Based Surveillance, IEEE Conference on*, pages 76–81, 2012.

[4] L. Gerosa, G. Valenzise, M. Tagliasacchi, F. Antonacci, and A. Sarti. Scream and gunshot detection in noisy environments. In *Proc. EURASIP European Signal Processing Conference*, Poznan, Poland, 2007.

[5] IEEE and ISO/IEC. Multimedia Content Description Interface - Part 4: Audio. *ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15*, 2001.

[6] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[7] Z. Liu, Y. Wang, and T. Chen. Audio Feature Extraction and Analysis for Scene Segmentation and Classification. *The Journal of VLSI Signal Processing*, 20(1):61–79, Oct. 1998.

[8] S. Ntalampiras, I. Potamitis, and N. Fakotakis. An adaptive framework for acoustic monitoring of potential hazards. *EURASIP J. Audio Speech Music Process.*, 2009:13:1–13:15, Jan. 2009.

[9] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM, 2004.

[10] J.-L. Rouas, J. Louradour, and S. Ambellouis. Audio events detection in public transport vehicle. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 733–738, 2006.

[11] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):591–606, 2009.

[12] M. Vacher, D. Istrate, L. Besacier, J. F. Serignat, and E. Castelli. Sound Detection and Classification for Medical Telesurvey. In C. ACTA Press, editor, *Proc. 2nd Conference on Biomedical Engineering*, pages 395–398, Innsbruck, Austria, Feb. 2004.

[13] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti. Scream and gunshot detection and localization for audio-surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 21–26, 2007.