Acoustic Event Detection and Localization with Regression Forests

Huy Phan[1,2], *Marco Maaß*[1,2], *Radoslaw Mazur*[1], and *Alfred Mertins*[1]

[1]Institute for Signal Processing, University of Lübeck, Germany
[2]Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Germany
{phan,maasz,mazur,mertins}@isip.uni-luebeck.de

Abstract

This paper proposes an approach for the efficient automatic joint detection and localization of single-channel acoustic events using random forest regression. The audio signals are decomposed into multiple densely overlapping *superframes* annotated with event class labels and their displacements to the temporal starting and ending points of the events. Using the displacement information, a multivariate random forest regression model is learned for each event category to map each superframe to continuous estimates of onset and offset locations of the events. In addition, two classifiers are trained using random forest classification to classify superframes of background and different event categories. On testing, based on the detection of category-specific superframes using the classifiers, the learned regressor provides the estimates of onset and offset locations in time of the corresponding event. While posing event detection and localization as a regression problem is novel, the quantitative evaluation on ITC-Irst database of highly variable acoustic events shows the efficiency and potential of the proposed approach.

Index Terms: acoustic event detection, regression forest, random forest, superframe

1. Introduction

Acoustic event detection (AED) finds many applications such as ambient assisted living [1], security surveillance [2], meeting room transcription [3], and "machine hearing" [4] to name a few. It has been under great attention of the research community with many recent evaluation campaigns including CLEAR 2006 [5], CLEAR 2007 [6], and AASP CASA 2013 [7]. AED problem is challenging due to large intra-class variations in terms of event duration and sounds, non-stationary background noise, as well as event overlapping.

A variety of techniques have been proposed. The most popular approaches often attempt to adapt the Automatic Speech Recognition (ASR) framework to deal with the problem. That is, they are based on frame-based features, such as Mel-frequency cepstral coefficients (MFCCs) and log Frequency Filter bank parameters, and Hidden Markov Models (HMMs) for recognition [8] [9]. Other systems employ discriminative approaches, e.g. Support Vector Machines (SVMs), to detect the events in detection-by-classification fashion [3] [10]. In general, the HMMs based ASR framework works better for the detection task while discriminative approaches are more successful for the classification task. Furthermore, for some applications it is necessary to have a good temporal resolution of the detected events. To be able to localize the events in time, discriminative approaches need to perform classification in multiple temporal scales, leading to tremendous computational burden.

In this work, we tackle the AED problem by jointly dealing with detection and localization as a regression problem. Motivated by the success of regression forests [11] [12] in various computer vision tasks, we adapt it for the AED task. Although the idea is plausible, the extension for reliable AED is not trivial since we need to decompose the events into multiple parts, and individual parts are able to be recognized independently at an acceptable accuracy. Fortunately, the acoustic *superframe* proposed in [13] satisfies this criteria and makes the idea practical. The training audio signals, containing multiple event occurrences of different categories, are divided into multiple interleaved superframes. Each superframe is associated with a class label and a 2-dimensional displacement vector to the onset and offset of the corresponding event. Thereafter, two classification models are learned using random forest classification [14]: one of them is to distinguish between event superframes from background superframes and the other is to subsequently classify event superframes into different categories of interest. Using the displacement vectors, category-specific regression models are built to map event superframes to estimates of onset and offset location of the events in time, i.e. we have a multi-variate, continuous parameter estimation problem, based on the random forest regression framework [11] [12]. On testing, the learned classifiers are applied to recognize event superframes which are finally inputted into the category-corresponding regressor to detect and localize the events from test audio signals.

In the domain of AED, our approach is most closely related to the work of Stork *et al.* [15] who use 40 ms frames stored in a codebook learned beforehand to vote for the event centers. However, our approach is different from their work in many perspectives. First, instead of unsupervised learning of codebooks with k-means, we use extremely randomized trees [16] to learn more meaningful discriminative codebooks. Second, their system allowed the frames stored in a codebook to vote backward and forward for the event centers, which are wildly uncontrollable (due to unsupervised learned codebooks). On the contrary, we model superframes in a codebook, i.e. a leaf node, as a continuous distribution and properly provide backward estimates for the onset and forward estimates for the offset. Last but not least, their goal is to find the event centers with assumption that all category-specific events have an equal duration. Yet, some categories experience large variation of intra-class duration in practice. Alternatively, our approach is able to provide scale-invariant continuous estimates of event onset and offset position.

The rest of this paper is organized as follows. We describe our algorithm to learn the multivariate regression forests in Section 2 and our event detection and localization system in Section 3. The experimental setup and results are presented in Section 4 followed by the conclusion in Section 5.

2. Multivariate Random Forest Regression

2.1. Random forest regression

A regression forest is an ensemble of different regression trees. Each of the trees implements a nonlinear mapping from complex input spaces into continuous output spaces. The nonlinearity is achieved by splitting the original problem into smaller ones, solvable with simple predictors. Each split node in the tree consists of a test that is applied to a data sample to send it toward the left or the right child node. The tests are picked by some criteria to group the training samples into clusters where a good prediction can be achieved by simple models. These models are computed from the annotated data samples that reached the leaves and were stored there. While overfitting likely happens for standard decision trees alone, an ensemble of randomly trained trees saw high generalization power [16].

2.2. Training

The training of our regressors is supervised and category-specific. Given a set of annotated superframes $\mathcal{S}^c = \{(\mathbf{x}_i, c, \mathbf{d}_i)\}$ of an event category $c \in \{1, \ldots, C\}$, each superframe $\mathbf{x} \in \mathbb{R}^M$ is associated with the class label c and a displacement vector $\mathbf{d} = (d_s, d_e) \in \mathbb{R}^2$. M is the dimensionality of feature space and C denotes the number of event categories of interest. d_s and d_e, respectively, denote the displacements (in superframes) of the current superframe to the onset and offset of the corresponding event as illustrated in Figure 1. Our aim is to learn to cluster superframes based on their features and their confidence in predicting the onsets and offsets of the events.

Generally, the tree construction for regression forests follows the common extremely randomized trees framework [16]. Each tree T in the forest $\mathcal{T} = \{T_t\}$ is constructed from a subset of superframes $S_i^c = \{(\mathbf{x}_i, c, \mathbf{d}_i)\}$ randomly sampled from $\mathcal{S}^c$. Starting from the root node, at each split node a large set of possible binary tests is randomly generated. A binary test is defined as $t_{f,\tau}$:

$$t_{f,\tau} = \begin{cases} 1, & \text{if } \mathbf{x}^f > \tau \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathbf{x}^f$ indicates the value of $\mathbf{x}$ at the feature channel $f \in \{1, \ldots, M\}$, and τ is a threshold. During the construction of the tree, at each split node, a pool of binary tests is generated with a random selected feature channel f and random values for τ generated in the range of $\mathbf{x}^f$. In our implementation, 20,000 random binary tests were considered for each split node. A test is selected from this pool to split the training samples into two sets: those satisfying the test are sent to the right child and the rest are sent to the left child. The data samples arriving at the node is evaluated by all binary tests in the pool and the test maximizing a predefined measure is selected and assigned to the node. In this work, the test is selected to minimize *displacement uncertainty* which is defined as:

$$U = \sum \left\| \mathbf{d}_i^{left} - \bar{\mathbf{d}}^{left} \right\|_2^2 + \sum \left\| \mathbf{d}_i^{right} - \bar{\mathbf{d}}^{right} \right\|_2^2, \quad (2)$$

where $\bar{\mathbf{d}}$ denotes the mean displacement vectors over all superframes in the set. This measure corresponds to the impurity of the displacement vectors. A leaf node is created when the maximum depth D_{max} is reached or a minimum number of superframes N_{min} is remained.

After training, each split node remains associated with the feature channel f and the threshold τ of the selected binary test. At each leaf node, we store the learned mean offset

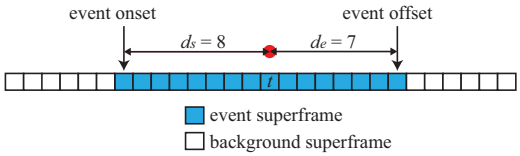

Figure 1: Displacements of the superframe at the time index t to the onset and the offset of an event.

$\bar{\mathbf{d}} = (\bar{d}_s, \bar{d}_e)$ and covariance matrix $\mathbf{\Gamma} = \begin{pmatrix} \Gamma_s & 0 \\ 0 & \Gamma_e \end{pmatrix}$, i.e. the parameters of a multivariate Gaussian distribution $\mathcal{N}(\bar{\mathbf{d}}, \mathbf{\Gamma})$. However, as it can be seen from the matrix $\mathbf{\Gamma}$, we do not consider covariance between the onset and offset displacements. That is, $\mathcal{N}(\bar{\mathbf{d}}, \mathbf{\Gamma})$ is equivalent to two univariate Gaussian distributions $\mathcal{N}(\bar{d}_s, \Gamma_s)$ and $\mathcal{N}(\bar{d}_e, \Gamma_e)$.

2.3. Testing

Each superframe classified as category c is passed through all the trees in the regression forest. At each split node, the stored binary test is applied to the superframe, sending it either to the right or left child until reaching a leaf node. At a leaf node l, the superframe gives estimates for onset and offset positions of the corresponding event in terms of the stored distribution $p(\mathbf{d}|l) = \mathcal{N}(\mathbf{d}; \bar{\mathbf{d}}, \mathbf{\Gamma})$. The posterior probabilities are summed up over all trees:

$$p(\mathbf{d}) = \sum_{l \in \bar{L}} p(\mathbf{d}|l). \quad (3)$$

Here, $\bar{L}$ is a subset of the corresponding leaf nodes.

3. Event Detection and Localization System

3.1. Acoustic superframe and its representation

In our system, it is essential that audio signals are decomposed into multiple parts, and each individual part is recognized independently. Therefore, instead of using small frames, e.g. 30 ms, we employ superframes, which are 100 ms long segments of acoustic signal, as proposed in [13]. It is a mid-level representation offering more discriminative power, hence being more reliable to be recognized independently. Furthermore, its temporal resolution is sufficient for event detection in superframe fashion since the detection error tolerance is usually set to 100 ms as in the most recent campaigns [7]. The temporal resolution can be further improved by overlapping.

A superframe is divided into interleaved small frames of 30 ms with Hamming window and 20 ms overlap. We utilized the set of 60 acoustic features suggested in [3] to represent a small frame. They consists of: (1) 16 log frequency filter bank parameters, along with the first and second time derivatives, and (2) the following set of features: zero-crossing rate, short time energy, four sub-band energies, spectral flux calculated for each sub-band, spectral centroid, and spectral bandwidth. Eventually, the empirical mean and the standard deviation of the frame feature vectors are calculated to form a 120-dimensional feature vector to represent the superframe.

3.2. System description

Given training audio signals annotated with events of C categories of interest, we decompose them into interleaved superframes with an overlap of 90% of their duration to obtain the training set $\mathcal{S} = \{(\mathbf{x}_i, c, \mathbf{d}_i)\}$. The dense overlap is to ensure a high level of data correlation, where the computational efficiency of decision trees allows us to do so. Each superframe, represented by a 120-dimensional feature vector as described in

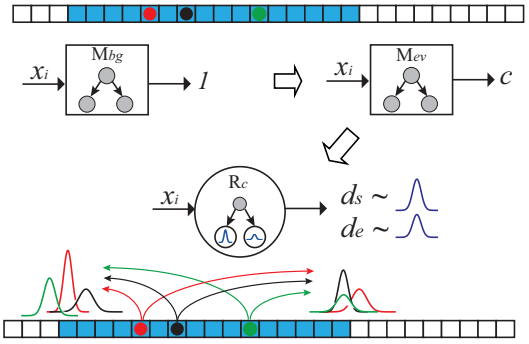

Figure 2: Pipeline for event detection and localization with the learned models.

Section 3.1, is annotated with the class label $c \in \{1, \ldots, C\}$ and the displacement vector $\mathbf{d} = (d_s, d_e)$. The background superframes are labelled as 0, and no offset vectors are required.

The system consists of the following classification and regression models which are trained using the training data $\mathcal{S}$:

- M_{bg}: the classifier to distinguish foreground superframes from background ones. It outputs 0/1 if the input superframe is background/foreground.
- M_{ev}: the classifier to recognize superframes between different event categories. It outputs c if the input superframe is of category c.
- R_c: the multivariate category-specific regressor to estimate the temporal onsets and offsets of the events of category c. C regressors are learned for C event categories.

Since the background noise can be easily distinguished from the events, it is reasonable to recognize and discard them first. Therefore, we learned two classifiers M_{bg} and M_{ev} for cascading classification rather than dealing with all the events and background at once. Due to dense overlapping of superframes, a large data is generated. For the dataset we use, the training and testing data contain 614,460 and 156,745 samples respectively. We adopt random forest classification [14] to train the classifiers to take advantage of its computational efficiency. For both classifiers, the number of random trees is conservatively set to 300. The regressors are trained with the random forest regression algorithm from Section 2 with ten random trees each. A randomly sampled subset containing 50% superframes of the category c training set $\mathcal{S}^c$ is used to train each random tree of R_c. In addition, we set the maximum depth $D_{max} = 12$ and minimum number of superframes at leaf nodes $N_{min} = 10$.

On testing, the pipeline of the event detection and localization system is illustrated in Figure 2. Given a test audio signal, we again divided it into multiple interleaved superframes as in the training phase. Afterwards, each superframe is inputted into M_{bg} to test for background. If the superframe is recognized as foreground by M_{bg}, it is further fed into M_{ev} to predict the event class label. After the recognition phase, the superframes with predicted class label c are pushed through the regressor R_c to estimate the onset and offset positions of the events of category c in the audio signal.

3.3. Event localization

To detect and localize the event of category c, we separately score each superframe at the time index t with the confidence of being event onset and offset using the regressor R_c:

$$Z_s(t) = \sum_d p(d_s) \quad \text{and} \quad Z_e(t) = \sum_d p(d_e). \qquad (4)$$

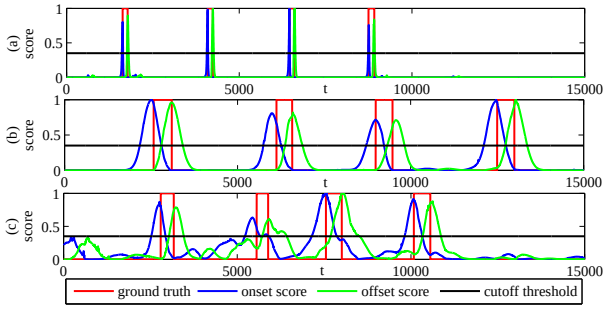

Figure 3: Alignment of the score peaks to the ground truth of the events: (a) door slam, (b) spoon cup jingle, and (c) steps.

In order to reduce the computation overhead during calculating the scores, we only evaluate the Gaussian distributions for the superframes in the displacement range of all superframes arriving at a leaf node during training. Moreover, we ignore the leaf nodes with the number of samples less than $N_{min} = 10$. Eventually, the larger the scores of a superframe are, the higher confidence we have that the event onset and offset occur at it.

Since the audio signals contain multiple event occurrences, in order to localize them, we need to determine the pairs of peaks in the Z_s and Z_e spaces. Furthermore, since our classifiers are not perfect, Z_s and Z_e are likely to be noisy, especially for events with low SNR. However, the peaks are expected to be dominant above the noise floor. We normalize the score to $[0; 1]$ and employ a cutoff threshold β for both Z_s and Z_e to discard the noise below it. Thereafter, the peaks in Z_s and Z_e are determined as the maximum values in the connected positive regions. These ideas are demonstrated in Figure 3. A pair of peaks, a Z_s peak followed by a Z_e peak in temporal order, is considered as a detection hypothesis. We impose a constraint that event duration should not exceed twice of the maximum duration of the events in the training audio signals.

4. Experiments

4.1. ITC-Irst acoustic event database

We use the database ITC-Irst of isolated meeting-room acoustic events [17], which has been extensively examined in recent CLEAR evaluations [5] [6], throughout our experiments. This database includes twelve recording sessions with 32 microphones and nine participants under the CHIL project [18]. It contains 16 semantic classes of events including door knock (kn), door slam (ds), steps (st), chair moving (cm), spoon cup jingle (cl), paper wrapping (pw), key jingle (kj), keyboard typing (kt), phone ring (pr), applause (ap), cough (co), laugh (la), mimo pen buzz, falling object, phone vibration, and unknown. Many of them are either subtle (low SNR, e.g. steps, chair moving, and keyboard typing), or/and overlapping with speech, making the task more challenging. Following the CLEAR 2006 setup, we only evaluate the first twelve classes. Nine recording sessions were employed as training files and three remaining sessions were employed as testing files. Only one channel *TABLE_1* was used.

4.2. Experimental setup and results

First of all, the audio signals were downsampled to 16 kHz. Using training files, we trained the classifier M_{bg} to separate background superframes from event ones and M_{ev} to classify superframes among 16 semantic event categories. Twelve categories-specific regressors were also trained for each of the twelve event categories of interest.

Table 1: Event detection performance for different categories with $\beta = 0.35$.

	kn	ds	st	cm	cl	pw	kj	kt	pr	ap	co	la
F-score (%)	100	90.9	91.7	81.5	100	100	95.7	91.7	75.7	100	86.9	90.9
E_{loc} (%)	17.3	45.1	32.5	43.6	23.9	27.9	31.1	22.4	27.5	7.9	60.5	41.2

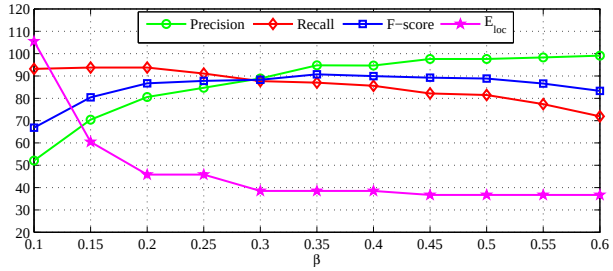

Figure 4: Event detection and localization results as a function of β.

We evaluated our system using two metrics: an F-score measure of detection accuracy, and an error rate E_{loc} which focuses more on localization error. Both of them were used in the CLEAR 2007 challenge [6]. They are defined as follows:

$$F\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{5}$$

$$E_{loc} = \frac{\sum_{seg}\{L \times (\max(N_\star, N_\triangleright) - N_\diamond)\}}{\sum_{seg}\{L \times N_\star\}}, \tag{6}$$

where in (5), $Precision$ denotes the ratio of the number of correctly outputted events over the number of all outputted events, and $Recall$ is the ratio of correctly detected ground-truth events over the number of all ground-truth events. In (6), E_{loc} is computed on the audio segments which only contain event duration, either ground-truth or system outputted. For each of such segment seg, L is the duration. $N_\star$, $N_\triangleright$, and $N_\diamond$ denote the number of ground-truth events, the number of outputted events, and the number of ground-truth events which coincides with outputted events in the segment seg, respectively. Note that E_{loc} may be larger than 100%.

The testing accuracies for M_{bg} and M_{ev} were 87.0% and 70.3% superframe-wise, respectively. The testing results of event detection and localization are shown in Figure 4 with different values of cutoff threshold β from 0.1 to 0.6 with a step size of 0.05. For simplicity, we used the same cutoff threshold across all categories. $\beta = 0.35$ appears to be a good choice where the F-score reaches the optimal value of 90.3%, and E_{loc} becomes stable with a value of 48.5% as β is decreasing. The detection performances for individual categories with $\beta = 0.35$ are shown in Table 1.

We compare our system performance with three submission systems to CLEAR 2006 [6] on the same dataset as in Table 2. The UPC-D and CMU-C1 share the same idea in that they first perform segmentation and then classification. However, while UPC-D employs sliding window with discriminative SVM, CMU-C1 relies on HMM models. The ITC-C1 merges the segmentation and classification in one step with ASR framework. Since these systems were only evaluated on *Acoustic Event Error Rate* (E_{det}) defined in the CLEAR 2006 challenge [19], we only use this metric for comparison. From Table 2, we see that our system outperforms all other systems and some with a large margin. Noticeably, this is also the case with most of the values of β in Figure 4. The rational is that these systems bring the noisy segmentation results into the final detection hypothesis, whereas we use them to further estimate the boundaries of the events with high confidence. As a result, the unreliable hypotheses outside the event boundaries are rejected by the cutoff threshold β.

Table 2: Performance comparison with CLEAR 2006 systems.

	Our system	UPC-D	CMU-D1	ITC-D1
E_{det} (%)	**18.5**	64.6	45.2	23.6

4.3. Discussion

In our experiments, we utilized a common cutoff threshold β for all event categories for the sake of simplicity. However, it is more reasonable that different threshold values should be adapted for different event categories since their scoring spaces behave differently as illustrated in Figure 3. Short events (like door slam) produce isolated peaks, periodic events (such as phone ring) lead to high value plateaus, and low-SNR events (like steps) experience a significant noise floor.

We argue that our system is robust to short-term noise. The recognition accuracies of the classifiers M_{bg} and M_{ev} are only at acceptable level and, in fact, they do not need to be perfect since we only need a portion of event superframes to be correctly recognized to estimate the onset and offset. In contrast, the performance of commonly adopted approaches, like sliding window with SVM and adapted ASR framework [19], strongly relies on the quality of the classifiers. In addition, this property also lead to the robustness to partial event overlapping, which is usually the case, although we have not discussed it in this paper.

Clearly, events in one category and events cross different categories can differ largely in their durations. Other approaches, like sliding windows [19] and event center detection [15], need to search on a huge temporal scale space to be able to localize the events. Our approach provides the continuous estimates for the onset and offset locations of the events. Therefore, we implicitly deal with this issue.

5. Conclusions

We proposed a novel approach for efficient automatic detection and localization of acoustic events based on regression forests. With the concept of acoustic superframe, we trained two classifiers to recognize the superframes of background and different event categories of interest. Based on the random forest regression framework, we further learn category-specific regressors using the event superframes annotated with their displacements to the onsets and offsets of the events. On testing, after an event superframe is recognized, the corresponding regressor will provide the estimates of the onset and offset of the event hypothesis in time. The excellent results on ITC-Irst acoustic event database demonstrate the efficiency and potential of our proposed approach.

6. Acknowledgements

This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1].

7. References

[1] J. Schröder, S. Wabnik, P. W. J. van Hengel, and S. Götze, *Ambient Assisted Living*. Springer, 2011, ch. Detection and Classification of Acoustic Events for In-Home Care, pp. 181–195.

[2] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE International Conference on Advanced Video and Signal based Surveillance*, 2007.

[3] A. Temko and C. Nadeu, "Acoustic event detection in meeting-room environments," *Pattern Recognition Letters*, vol. 30, pp. 1281–1288, 2009.

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

[5] CLEAR 2006: Classification of events, activities and relationships. evaluation and workshop. http://isl.ira.uka.de/clear06, accessed on 20 Mar 2014.

[6] CLEAR 2007: Classification of events, activities and relationships. evaluation and workshop. http://www.clear-evaluation.org, accessed on 20 Mar 2014.

[7] IEEE AASP CASA challenge 2013: Detection and classification of acoustic scenes and events. http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/, accessed on 20 Mar 2014.

[8] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "Hmm-based acoustic event detection with adaboost feature selection," *Lecture Notes in Computer Science*, vol. 4625, pp. 345–353, 2008.

[9] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.

[10] A. Temko, C. Nadeu, and J.-I. Biel, "Acoustic event detection: Svm-based system and evaluation setup in clear'07," *Lecture Notes in Computer Science*, vol. 4625, pp. 354–363, 2008.

[11] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu, "Regression forests for efficient anatomy detection and localization in computed tomography scans," *Medical Image Analysis*, vol. 17, no. 8, pp. 1293–1303, 2013.

[12] J. Gall, A. Yao, N. Razavi, L. V. G. Member, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[13] H. Phan and A. Mertin, "A voting-based technique for acoustic event-specific detection," in *40th Annual German Congress on Acoustics (DAGA 2014)*, 2014.

[14] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[15] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-markovian ensemble voting." in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'12)*, 2012.

[16] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[17] C. Zieger and M. Omologo, "Acoustic event detection - ITC-irst AED database," Internal ITC report, Tech. Rep., 2005.

[18] CHIL. Computers in the human interaction loop. http://www.ipd.uka.de/CHIL/, accessed on 20 Mar 2014.

[19] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," *Lecture Notes in Computer Science*, vol. 4122, pp. 311–322, 2007.