Pattern Recognition Letters 34 (2013) 1085–1093

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Overlapping sound event recognition using local spectrogram features and the generalised hough transform

J. Dennis [a,b,*], H.D. Tran [a], E.S. Chng [b]

[a] Institute for Infocomm Research, 1 Fusionopolis Way, #08-01 South Tower Connexis, Singapore 138632, Singapore
[b] School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 30 August 2012
Available online 14 March 2013

Communicated by S. Sarkar

Keywords:
Overlapping sound event recognition
Local spectrogram features
Keypoint detection
Generalised Hough Transform

ABSTRACT

In this paper, we address the challenging task of simultaneous recognition of overlapping sound events from single channel audio. Conventional frame-based methods are not well suited to the problem, as each time frame contains a mixture of information from multiple sources. Missing feature masks are able to improve the recognition in such cases, but are limited by the accuracy of the mask, which is a non-trivial problem. In this paper, we propose an approach based on Local Spectrogram Features (LSFs) which represent local spectral information that is extracted from the two-dimensional region surrounding "keypoints" detected in the spectrogram. The keypoints are designed to locate the sparse, discriminative peaks in the spectrogram, such that we can model sound events through a set of representative LSF clusters and their occurrences in the spectrogram. To recognise overlapping sound events, we use a Generalised Hough Transform (GHT) voting system, which sums the information over many independent keypoints to produce onset hypotheses, that can detect any arbitrary combination of sound events in the spectrogram. Each hypothesis is then scored against the class distribution models to recognise the existence of the sound in the spectrogram. Experiments on a set of five overlapping sound events, in the presence of non-stationary background noise, demonstrate the potential of our approach.

1. Introduction

The topic of sound event recognition (SER) covers the detection and classification of sound events in unstructured environments, which may contain multiple overlapping sound sources and non-stationary background noise. Many sounds contribute to the understanding and context of the surrounding environment, and therefore should not be regarded simply as noise, as is common in automatic speech recognition (ASR). Instead, such sounds are useful in many applications, such as security surveillance (Gerosa et al., 2007), bioacoustic monitoring (Bardeli et al., 2010), meeting room transcription (Temko and Nadeu, 2009; Zhuang et al., 2010), and ultimately "machine hearing" (Lyon, 2010).

Although a variety of techniques has been developed for SER (Cowling and Sitte, 2003), the most popular approaches are often based on frame-based features, such as Mel-frequency cepstral coefficients (MFCCs) from ASR, or MPEG-7 descriptors (Casey, 2001). These can then be modelled with Gaussian Mixture Models (GMMs) and combined with Hidden Markov Models (HMMs) for recognition, or used to train a Support Vector Machine (SVM) for discriminative classification. While these methods are effective in ASR for clean single-source speech recognition (O'Shaughnessy, 2008), such systems may not perform well in the challenging mismatched conditions present in many SER tasks. Missing feature recognition systems can overcome the problem to an extent (Raj and Stern, 2005), however a major challenge is estimating the mask to separate the signal from the background noise (Wang, 2005), and in practise the performance of such systems is highly dependent on the quality of the mask. Such frame-based techniques are also not well suited to recognition of overlapping sounds, as each feature contains information from multiple sources.

Research into the human understanding of speech (Allen, 1994) shows that there is little biological evidence for frame-based features, and that the human auditory system may be based on the partial recognition of features that are local and uncoupled across frequency. This enables the human recognition system to be robust to noise and distortion occurring across separate regions of the spectrum. Therefore, in this paper we develop an SER system based on local spectrogram features (LSFs), which provide a significant departure from conventional frame-based features. This work extends our initial presentation of the idea in (Dennis et al., 2012), where here we introduce a more complete model of the sound

* Corresponding author at: Institute for Infocomm Research, 1 Fusionopolis Way, #08-01 South Tower Connexis, Singapore 138632, Singapore. Tel.: +65 84842036.
E-mail addresses: stujwd@i2r.a-star.edu.sg (J. Dennis), hdtran@i2r.a-star.edu.sg (H.D. Tran), ASESChng@ntu.edu.sg (E.S. Chng).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.02.015

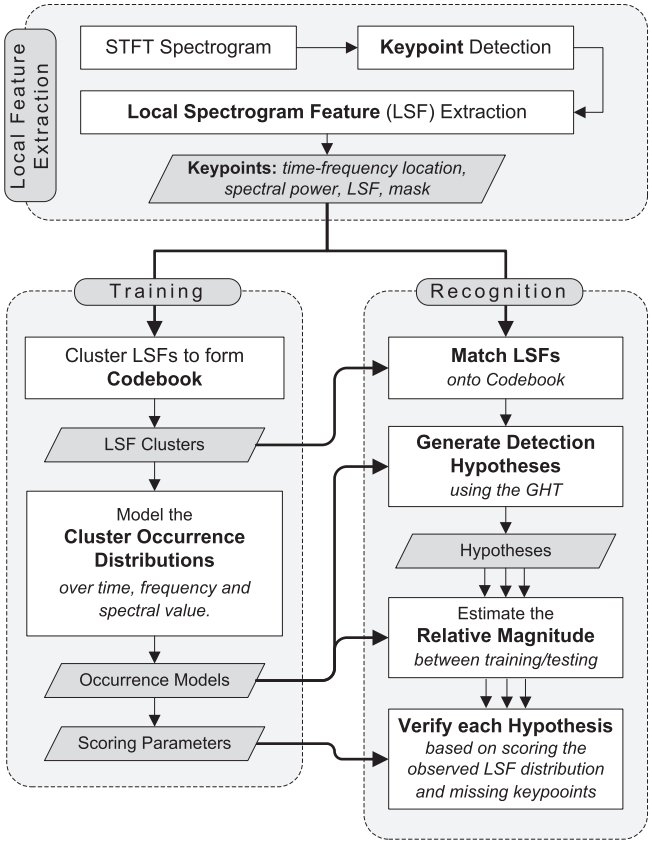


Fig. 1. Overview of the proposed LSF recognition system.

information in the spectrogram, and enhance the recognition and scoring process to achieve more robust recognition across a range of experimental conditions.

Our proposed method takes inspiration from works in the field of object detection from image processing by Lowe (2004) and Lehmann et al. (2011), where finding objects in a cluttered real-world scene can be seen as having many parallels with that of overlapping SER. The central idea is to characterise a spectrogram by a set of independent local features, where each feature represents a glimpse (Cooke, 2006) of the local spectral information. Fig. 1 gives an overview of the idea, which can be broken down into the following three steps:

1. *Local feature extraction*: We first detect "keypoints" in the spectrogram to locate characteristic spectral peaks and ridges. For each keypoint, we extract an LSF and local missing feature mask to represent the local spectral region.
2. *Training*: The extracted LSFs are first clustered to generate a codebook. Each sound event is then modelled through the keypoint-occurrence distribution of the codebook clusters in the training spectrograms, and scoring parameters are extracted for verification during testing.
3. *Recognition*: The LSFs are first matched onto the codebook. We then generate sound onset hypotheses using the Generalised Hough Transform (GHT) (Ballard, 1981), which is a voting system that sums the keypoint-cluster distribution information in the Hough accumulator space. Finally, we verify each hypothesis by estimating the relative magnitude of the sound event between training and testing, and scoring it against the trained model.

The key advantage of our method is the use of local features combined with the GHT, which was successful in object detection in image processing (Lowe, 2004). The local features allow for independent local glimpses of the sound to be extracted, and a local missing feature mask to be estimated, which makes the system more robust to non-stationary noise. In addition, as the Hough accumulator is a summation of local evidence, a sound can still be recognised even when a proportion of features is missing or corrupted due to noise or overlapping sounds. Also, the representation of each sound in the Hough accumulator space is sparse and separable, such that overlapping sounds will produce distinct spikes in the accumulator that can be detected. This is an advantage over conventional HMM recognition systems, where the likelihoods are multiplicative, such that noise or overlapping sounds affecting one part of the feature has an adverse affect on the whole recognition.

Previous work on recognition of overlapping sounds can be separated into two distinct methodologies. The first is blind source separation, where factorisation is commonly used to decompose the input signal. For example, Heittola et al. (2011) use unsupervised non-negative matrix factorisation (NMF) to process the input audio into four component streams, where different sound events may be separated into different streams for recognition. Dessein et al. (2012) apply additional constraints such as sparsity on the NMF to improve the decomposition. Experiments show that both systems can separate overlapping sounds to some extent, although it is noted by Heittola et al. (2011) that the problem of controlling the outcome of the factorisation is one of the major difficulties with the NMF approach.

The second group of methods are based on direct classification. One approach developed for ASR is Factorial HMMs (FHMMs) (Roweis, 2003), based on the MixMax model of source interaction (Nádas et al., 1989), where the best combination of hidden states is found among the trained models to explain the observed feature. A more recent approach by Temko and Nadeu (2009) uses hierarchical SVM, where the first SVM classifies the input as either isolated events or a combined "overlapped" class, and the second SVM then identifies the overlapped combination. Tran and Li (2011) use a different approach that transforms the probabilistic distribution of the subband information to a new domain, where SVM can be used to detect sounds within a confidence interval.

Previous works have also used local spectral information, although not in the context of overlapping sounds. For example, Kleinschmidt and Gelbart (2002) use local Gabor filters to approximate the spectro-temporal response field (STRF) of the human auditory cortex. More recently, Heckmann et al. (2011) learn a set of local features from the data, and combine this in a hierarchical framework to obtain features spanning a larger frequency and time region. Other works have applied image-processing techniques to the spectrogram for speech, sound and music environments (Schutte, 2009; Dennis et al., 2011; Matsui et al., 2011). However, these approaches often just extend frame-based techniques, apply methods directly from image processing, or combine block-based techniques with SVM to classify the whole spectrogram. The Hough transform has also been used previously for tasks such as localisation (Marchand et al., 2009) and word spotting (Barnwal et al., 2012), although in such works it is typically used to detect straight lines, and not general shapes as we perform here with the GHT.

The rest of the paper is organised as follows. Section 2 describes the LSF extraction. Section 3 details the clustering and modelling to train the system. Section 4 describes the recognition approach based on the GHT. Section 5 then details our experiments, before Section 6 concludes the work.

2. Local Spectrogram Feature Extraction

In this section, we describe the extraction of Local Spectrogram Features (LSFs), from the two-dimensional spectrogram representation of the sound. Here we use the log-power Short-Time Fourier

Transform (STFT), $S(f,t)$, where f represents the frequency bin and t is the time frame. We use a sampling frequency of 16 kHz and 16 ms time windows with a 50% overlap, giving a total of $F = 129$ frequency bins.

2.1. Keypoint detection

The idea is to detect a set of keypoints, P_i, which locate the important spectral structures of the sounds in the spectrogram. We write these as:

$$P_i = [f_i, t_i, s_i] \tag{1}$$

where $s_i = S(f_i, t_i)$ is the log-spectral power, and i is the index of the keypoint in the spectrogram. Together, these keypoints capture important geometrical information about the sound, and together with an LSF, L_i, extracted from each keypoint, jointly characterise the sound through these local features and their geometrical distribution.

Here we propose to use a plus-shaped local region, composed of the local horizontal and vertical spectral shapes within a radius D of the central point, to form the basis for both our keypoint detection and LSF extraction, as follows:

$$\begin{aligned} Q_T(f,t) &= \{S(f, t \pm d)\} \\ Q_F(f,t) &= \{S(f \pm d, t)\} \end{aligned} \quad \forall 1 \leqslant d \leqslant D \tag{2}$$

where Q_T and Q_F capture the local time and frequency dimensions respectively. From preliminary experiments we found that $D = 6$ was small enough to localise the spectral peaks, but large enough to provide a feature for clustering, hence is used throughout. This represents a local region with a frequency range of approximately 800 Hz and a time window of 100 ms.

The idea is to capture the spectral and temporal shape separately, such that it gives a "glimpse" of the local spectrogram information in two dimensions (Cooke, 2006). We found that this is more suitable than including the full two-dimensional region, which may become dominated by non-stationary noise or overlapping sounds. This can be seen in the example of a bell and phone overlapping in Fig. 2. Here, although the local region is dominated by the phone sound, the highlighted keypoint can still be detected on the harmonic of the bell, and the extracted LSF provides a glimpse of the bell sound from the overlapping mixture.

We also generate a local noise estimate, $\eta(f,t)$, for each point in the spectrogram by assuming that the noise will be stationary across either the spectral or temporal dimension of the local region. As one dimension may be dominated by the signal, we take the minimum of the two means as our estimate:

$$\eta(f,t) = \frac{1}{2D} \min\left(\sum_{y=1}^{2D} Q_T(f,t,y), \sum_{y=1}^{2D} Q_F(f,t,y) \right), \tag{3}$$

where $1 \leqslant y \leqslant 2D$ are the vector indices of Q_T and Q_F.

Keypoints, $P_i = [f_i, t_i, s_i]$, are then detected at locations that are local maxima across either frequency or time, subject to a local signal-to-noise ratio (SNR) criterion, as follows:

$$S(f_i,t_i) \geqslant \begin{cases} \max\left(Q_T(f_i,t_i,y), Q_F(f_i,t_i,y)\right), & \forall y \\ \eta(f_i,t_i) + \delta_{SNR} \end{cases} \tag{4}$$

where $\delta_{SNR} = 5$ dB is the local SNR threshold that must be exceeded for the keypoint to be detected (Cooke et al., 2001). This is possible since the spectral values represent the relative importance of the keypoint for recognition, unlike in image processing where gradients are typically more important than intensity or colour values (Mikolajczyk and Schmid, 2004).

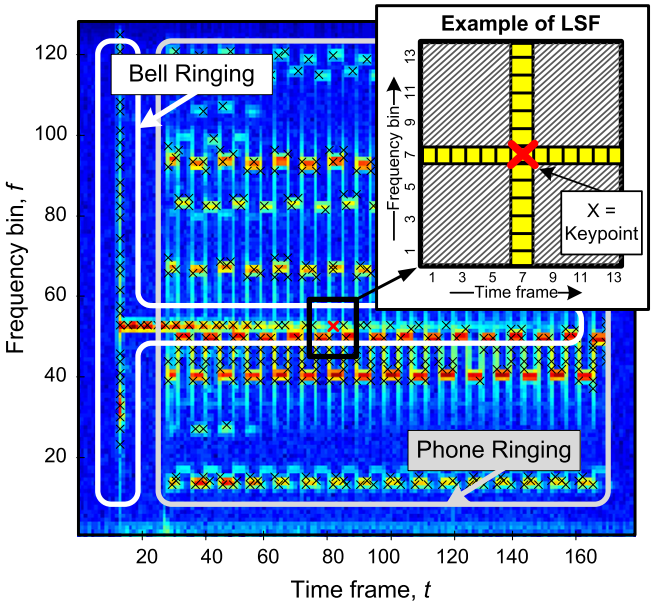


Fig. 2. Example of bell and phone ringing sounds overlapped, where × represents the detected keypoints. The highlighted region gives an example of the proposed plus-shaped LSF, which gives a glimpse of the bell from the mixture.

2.2. Local Spectrogram Feature Extraction

We now extract an LSF, L_i, from each of the detected keypoints in the spectrogram to characterise the local spectral information. For this, we propose to use the normalised plus-shaped local spectrogram region, as follows:

$$L_i = \left[\frac{Q_T(f_i,t_i)}{s_i}, \frac{Q_F(f_i,t_i)}{s_i} \right] \tag{5}$$

where $s_i = S(f_i,t_i)$. This characterises only the local spectral shape and not the magnitude of the sound, as this may vary between training and testing. Instead, the magnitude information is captured in the geometrical distribution model of the sound, which we describe in the next section.

In addition, as noise may affect the local region between training and testing, we extract a missing feature mask, M_i, for each LSF as follows:

$$M_i(z) = \text{sign}\left(L_i(z) - \frac{\eta(f_i,t_i)}{s_i} \right) \tag{6}$$

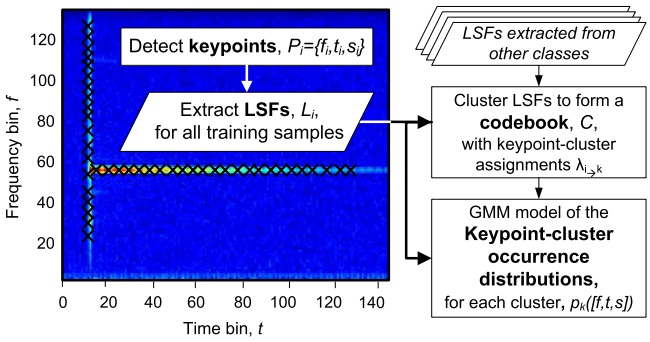
where $z = 1, \ldots, 4D$ is the variable representing the LSF dimensions and $\eta(f,t)$ is the local noise estimate from (3). Note that $M_i(z) = -1$ denotes the unreliable LSF dimensions.

3. Spectro-Geometrical Sound Event Modelling

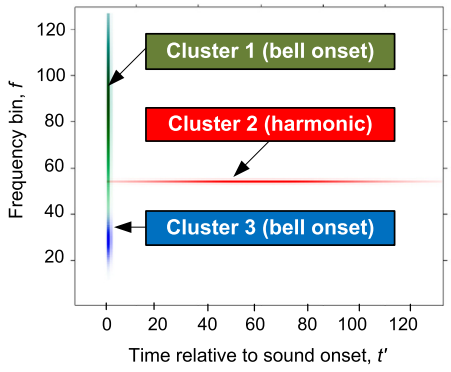
In this section, we describe the process of training a model of each sound class, based on the keypoints and LSFs extracted from the spectrogram. The idea is to characterise sounds jointly through the matching of the local spectrogram features onto the codebook, and the geometrical distribution of the corresponding keypoints in the spectrogram, relative to the sound onset.

An overview of this is shown in Fig. 3a, where the modelling steps can be summarised as follows:

1. Detect keypoints and extract LSFs for all training samples from all classes, using the method described in Section 2.
2. Cluster all LSFs to form a codebook.



(a) Example of the modelling process for the bell sound, where the extracted LSFs are clustered together, and their distribution over time, frequency and spectral power is modelled in a GMM.



(b) The marginal keypoint-cluster occurrence distributions over time/frequency for the top three clusters.

Fig. 3. Overview of the model training process, showing an example for the bell class.

3. For training samples of a given class, model the geometrical distribution of the keypoints assigned to each cluster over frequency, power and time relative to the onset.
4. Extract scoring parameters for recognition.

An example of the time–frequency cluster distribution for the top three clusters for the bell class is shown in Fig. 3b. From the figure, we can see that both the onset and the harmonic of the bell can be neatly modelled by a small number of localised clusters. This demonstrates the spectro-geometrical information captured by the modelling process, such that, during recognition, keypoints must be matched to the same clusters and have the same geometrical distribution.

3.1. LSF codebook clustering

Here, we generate codebook clusters of LSFs that are independent of the sound class, and represent characteristic local patterns found in the training samples. For this we use K-means clustering, where the output is a set of K codebook entries, C_k, where $k = 1,\ldots,K$, such that $\lambda_{i\to k}$ denotes the assignment of LSFs, L_i, to cluster, C_k.

The choice of the number of clusters, K, is important to be able to model the LSF patterns sufficiently well. We found that as long as K is large enough, for example $K = 200$, the performance did not vary significantly and the clustering produced compact clusters with a small variance.

We model each dimension of the codebook entries, $C_k(z)$ as a Gaussian distribution, with the mean, μ_k, and variance, σ_k^2, calculated as follows:

$$\begin{aligned}\mu_k(z) &= \tfrac{1}{n_k}\sum_{\lambda_{i\to k}} L_i(z)\\ \sigma_k^2(z) &= \tfrac{1}{n_k}\sum_{\lambda_{i\to k}}\left[L_i(z)-\mu_k(z)\right]^2\end{aligned} \tag{7}$$

where $z = 1,\ldots,4D$ are the LSF dimensions, and n_k represents the number of LSFs assigned to cluster C_k.

3.2. Spectro-geometrical modelling

We now model each sound class, X, through the geometrical distribution of the observed keypoints, P_i^X, assigned to each cluster in the training samples, $\lambda_{i\to k}^X$. Unlike in image processing, where only two dimensions are typically modelled (Lowe, 2004), here we can model the sound geometry over three dimensions: frequency, time and spectral power. This captures the full trajectory of the sound in the spectrogram, and enables us to distinguish between rising and falling tones.

Prior to modelling, we ensure that the keypoints in each training sample are normalised to have the same onset, such that:

$$t_i' = t_i - t_{ON}, \tag{8}$$

where t_{ON} is the onset time. This ensures that the keypoint timing information is consistent across the different training samples. As clean isolated samples are used for training, we simply take the first keypoint detected in the sample as the sound onset.

The keypoint-cluster occurrence distribution, $p_k^X([f,t,s])$, is then modelled using a three-dimensional GMM probability density function (PDF) as follows:

$$p_k^X([f,t,s]) = \sum_{m=1}^{m_k} c_{km}\mathcal{N}\left([f_i,t_i',s_i];v_{km}^X,\Sigma_{km}^X\right) \tag{9}$$

where m_k is the number of mixture components in cluster k, c_{km} is the weight of the mth component and $\mathcal{N}([f,t,s];v_{km}^X,\Sigma_{km}^X)$ is a multivariate Gaussian model, with mean vector v and covariance matrix Σ. We estimate the PDF using the algorithm from Baggenstoss (2002), which uses a Kurtosis-based mode splitting method that does not require any prior knowledge about the number of mixtures required to model each cluster.

As the LSF codebook clustering takes place over samples from all classes, not every entry in the codebook will appear for every class. Therefore, we model only a subset of clusters with the highest number of keypoints for the given class, n_k^X, such that the trained model explains 95% of the observed keypoints in the training samples. This ensures that only the most consistently occurring clusters are used to model the sound.

3.3. Cluster verification score

The final step is to extract scoring parameters, which are used as a threshold for hypothesis verification during testing. The first is the voting count of the cluster, v_k^X, which represents the average log-spectral power assigned to the cluster:

$$v_k^X = \frac{1}{N}\sum_{\lambda_{i\to k}^X} s_i \tag{10}$$

where N is the number of training samples provided for the class, X, and $\lambda_{i\to k}^X$ represents LSFs from class X assigned to codebook cluster, C_k. This is used as a cluster decision threshold, which must be obtained before the cluster can be determined to have existed in the sound clip. We use the spectral power, as opposed to simply the number of keypoints, as this gives more weight to the characteristic high-power peaks in spectrogram.

The second is the cluster score, w_k^X, which represents the relative weight that the cluster contributes to the sound class:

$$w_k^X = \frac{v_k^X}{\sum_{k=1}^{K} v_k^X} \tag{11}$$

This is used to score the hypothesis. Since $\sum_{k=1}^{K} w_k^X = 1$ for each sound class, we can set a threshold for accepting a hypothesis that provides a tradeoff between false rejection and acceptance.

4. LSF-based sound event recognition

During recognition, we assume that we do not know what combination of sound events may be found in each clip, or the onset or relative magnitude of the sound events between the training and testing. Therefore, we cannot simply fit the keypoint-cluster distribution model, $p_k^X([f,t,s])$, from training.

An overview of our proposed approach is shown in Fig. 4, where the idea is to generate sound event hypotheses from the keypoint-cluster information. The recognition steps can be summarised as follows:

1. Detect keypoints and extract LSFs from the spectrogram as described in Section 2.
2. Match the LSFs onto the codebook.
3. Generate detection hypotheses, $H = [h_1, h_y, \ldots, h_Y]$, using the Generalised Hough Transform (GHT), which uses the geometrical model associated with each codebook entry as a voting function to find the sound onset.
4. Estimate the relative magnitude transfer function between training and testing for each hypothesis.
5. Score the hypotheses against the trained model.

The process therefore jointly uses the local spectral and geometrical information to recognise the sound based on the set of observed keypoints in the spectrogram. A sound event can only by recognised when both the keypoint-cluster assignments, and their associated geometrical distributions, match those found in training. The rest of this section describes steps 2–5 in detail.

4.1. LSF codebook matching

The extracted LSFs, L_i, are first matched onto the codebook, using each keypoint's missing feature mask, M_i from (6), to assigned each keypoint to the closest cluster. For the unreliable feature dimensions, $M_i(z) = -1$, we perform bounded marginalisation against the codebook.This uses the observed spectral information as an upper bound for the probability distribution, and has been shown to perform better than completely marginalising the missing dimensions (Cooke et al., 2001). Together, the reliable and unreliable dimensions sum together to give an overall log-likelihood score, $l_{i,k^\star}$, as follows:

$$l_{i,k^\star} = \sum_{z\in\{M_i(z)=1\}} \log c_{k^\star}(L_i(z)) + \sum_{z\in\{M_i(z)=-1\}} \log F_{k^\star}(L_i(z)) \tag{12}$$

where $c_k(\Lambda) = \mathcal{N}(\Lambda; \mu_k(z), \sigma_k^2(z))$ is the Gaussian probability density function, and $F_k(x) = \int_{-\infty}^{x} c_k(\Lambda)\,\mathrm{d}\Lambda$ is the cumulative distribution function, where Λ represents the normalised spectral power in the LSF, as calculated in (5).

Each keypoint is assigned to the winning cluster, as follows:

$$k = \underset{k^\star}{\operatorname{argmax}} \left(l_{i,k^\star} \right) \tag{13}$$

such that $\lambda_{i\to k}$ denotes a set of keypoint-cluster assignments.

4.2. Overlapping sound event hypotheses using the generalised hough transform

All keypoints belonging to the same sound event in the spectrogram will share a common onset, and their distribution relative to the onset should match that modelled in the training. We can therefore use the geometrical distribution model, p_k^X, as a voting function for the GHT to accumulate evidence for such onsets based on the keypoints-cluster assignments, $\lambda_{i\to k}$. For each observed keypoint, occurring at time t_i, the distribution of the onset time, relative to the keypoint, can be written as $p_k^X([t_i - t])$. Summing these together over the observed keypoints will produce a peak at the corresponding onset time when the keypoint-cluster distribution matches that modelled in the training. Incorrect matches due to noise will not affect this process, as there will not be a significant number of matches that agree on the same onset time.

As the relative magnitude between training and testing is unknown, we use the marginal distribution over frequency and time as the GHT voting function, which we write as follows:

$$g_i^X(t') = p_k^X([f, t_i - t] | f = f_i) \tag{14}$$

where p_k^X is the geometrical model taken from (9), f_i, t_i, are the frequency and time coordinates of observed keypoint, and $t' = t_i - t$ is the variable representing the hypothesised onset time.

The Hough accumulator, $H^X(t')$, is then the summation of the voting functions as follows:

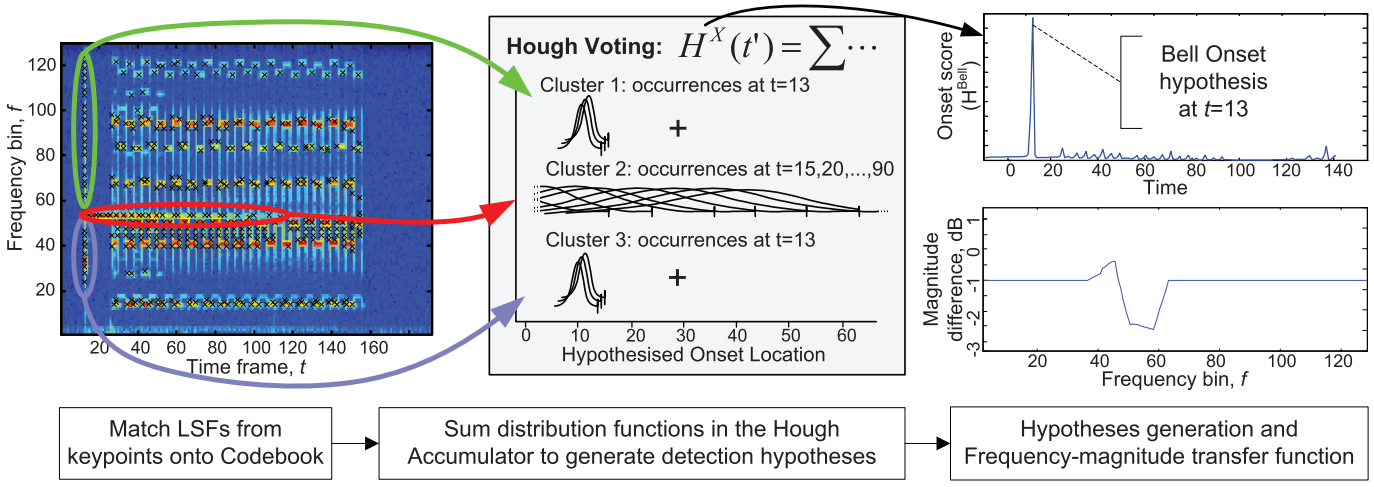


Fig. 4. Example of the recognition steps for the bell class from a mixture two sounds. A sound event can only by recognised in testing when both the clusters and geometrical distributions of the keypoints match those found in training. A detection is indicated by a sparse peak in the Hough accumulator.

$$H^X(t') = \sum_{\lambda_{i\to k}} g_i^X(t') \tag{15}$$

where $\lambda_{i\to k}$ is the set of keypoint-cluster assignments.

Local maxima in the Hough accumulator correspond to hypotheses, h_y, of combined evidence of a particular sound event. We write these as follows:

$$h_y = \{X_y, t_{ON,y}, \lambda^y_{i\to k}\} \tag{16}$$

where y is the index for the hypothesis, and each hypothesis is specified by its class, X_y, onset time, $t_{ON,y}$, and set of contributing keypoints, $\lambda^y_{i\to k}$.

As $H^X(t')$ is a sum of weighted GMM distributions, we use the gradient-based approach of Carreira-Perpinan (2000) to locate these maxima. We also set a minimum threshold at 20% of the mean peak values obtained during training, which defines a minimum amount of observed evidence for a hypothesis to be generated. By doing this, it allows us to recognise an arbitrary combination of sound events in the spectrogram, including two overlapping instances of the same sound class. An example is shown on the right side of Fig. 4, which shows the Hough accumulator for the bell sound from the mixture, clearly showing a sparse peak indicating the onset.

4.3. Relative magnitude estimation

Although we now have a set of sound event hypotheses, and their onset times, we still do not know the relative magnitude transfer function of the sound between training and testing, hence cannot score the hypothesis against the model from Section 3.

Assuming the sound is subjected to an unknown convolutive channel distortion, this becomes additive in the log-power STFT domain as follows:

$$S(f,t) + R(f,t) \approx S(f,t) + R(f) \tag{17}$$

where $S(f,t)$ represents the log-power STFT of a clean training sample, and R is the transfer function between the observed spectrograms in training and testing. Here, we make the assumption that the response time of the channel is short, hence $R(f,t)$ can be approximated as $R(f)$, such that the channel distortion does not vary over time.

We can estimate $R(f)$ using our trained sound model, p_k^X from (9), by using the conditional distribution of the difference in log-spectral power as a voting function for a second GHT. We write this as:

$$g_i^y(s') = p_k^{X_y}([f, t, s_i - s] | f = f_i, t = t'_i) \tag{18}$$

where $s' = s_i - s$ is the variable representing the spectral power difference and f_i, t'_i, s_i is the three-dimensional location of the keypoint in frequency, time and spectral power, where $t'_i = t_i - t_{ON,y}$ is the time relative to the hypothesised onset.

As in (15), the Hough accumulator, $H_f^y(s')$, is the summation of the voting functions. As the transfer function may vary over frequency, we sum up the voting function in each subband, as follows:

$$H_f^y(s') = \sum_{\lambda^y_{i\to k}} g_i^y(s'), \forall f = f_i \tag{19}$$

where $\lambda^y_{i\to k}$ are keypoint-cluster assignments for hypothesis y.

We now find the maximum of the accumulator in each frequency subband, which corresponds to combined evidence for a given relative magnitude transfer function, $R(f)$, in that subband. This is written as:

$$R(f) = \underset{s'}{\operatorname{argmax}}\, H_f^y(s') \tag{20}$$

As some frequency subbands have very few keypoints, they do not contain reliable evidence to estimate $R(f)$, hence these are replaced with the mean of the transfer function across the remaining reliable values. Here, subbands with less than 10% of the number of keypoints of the most reliable subband are considered unreliable. An example is shown on the right side of Fig. 4, where the Bell sound is estimated to be on average 1 dB less than the training samples, with the Bell's harmonic an additional 1.5 dB quieter.

4.4. Hypothesis scoring and decision

Starting with the hypothesis, h_y, that explains the largest number of keypoints, we first evaluate the observed keypoints and spectrogram for the hypothesised class, X_y, and onset time $t_{ON,y}$ against the trained model from (9). We remove keypoints contributing to the hypothesis that have a likelihood less than a threshold:

$$p_k^{X_y}([f_i, t'_i, s'_i]) < \gamma_k^{X_y} \tag{21}$$

where $t'_i = t_i - t_{ON,y}$ and $s'_i = s_i + R(f_i)$ are set to align the keypoints with the trained model. The threshold γ_k^X is set for each cluster based on the likelihood distribution of keypoints found during the training, such that 95% of keypoints are matched.

The cluster voting score for the observed keypoints, $v^y_{k,O}$, is then calculated by summing together the spectral power:

$$v^y_{k,O} = \sum_{\lambda^y_{i\to k}} s'_i. \tag{22}$$

which is analogous to the score obtained during training in (10).

However, for overlapped regions of the spectrogram, some keypoints will be missing according to the MixMax criteria. Therefore, we allow time–frequency locations that had a high likelihood in the training, $p_k^X([f,t]) > \beta_k^X$, to contribute to the cluster score. As the keypoint is missing, the vote is based on the expected spectral power at that location, $S_k^{X_y}(f,t)$, which is calculated in a maximum likelihood sense, as follows:

$$S_k^X(f,t) = \underset{s}{\operatorname{argmax}}\, p_k^X([f,t,s]). \tag{23}$$

The cluster voting score for missing keypoints is then calculated as follows:

$$v^y_{k,M} = \sum_{p_k^{X_y}([f,t])>\beta_k^{X_y}} \begin{cases} S_k^{X_y}(f,t), & \text{if } S(f,t') > S_k^{X_y}(f,t) \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

where $t' = t - t_{ON,y}$ is the time relative to the hypothesised sound onset. In our experiments, we set $\beta_k^X = 0.5 \times \max(p_k^X([f,t]))$, which prevents keypoint locations with a low likelihood from biasing the cluster score.

The final hypothesis score, $score(h_y)$, is calculated as the sum of the cluster weight from clusters exceeding a decision threshold, as follows:

$$score(h_y) = \sum_{k=1}^{K} \begin{cases} w_k^{X_y}, & \text{if } \left(v^y_{k,O} + \alpha_1 v^y_{k,M}\right) > \alpha_2 v_k^{X_y} \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

where v_k^X and w_k^X are the scoring parameters found during training in (10) and (11), $\alpha_1 = 0.8$ is a weighting factor to balance the score between observed and missing keypoints, and $\alpha_2 = 0.5$ is a threshold that defines the minimum score required for the cluster to be matched in the spectrogram.

If the hypothesis score exceeds a threshold:

$$score(h_y) > \Omega, \tag{26}$$

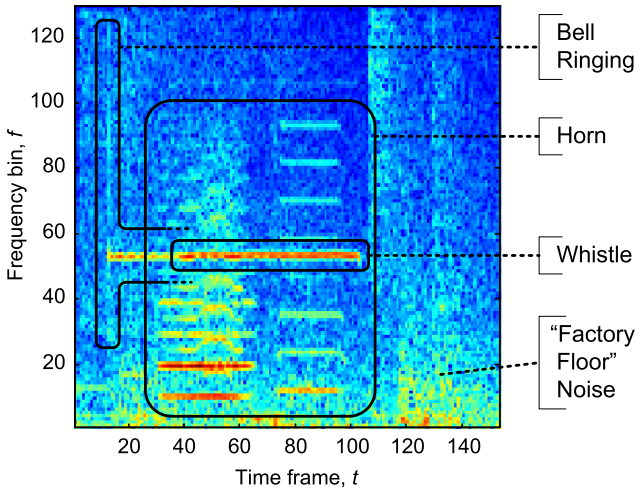
then the hypothesis is accepted. This threshold can be varied to control the tradeoff between false rejection and acceptance of hypotheses. In our experiments, we use $\Omega = 0.5$, such that at least half of

the clusters must be matched for the hypothesis to be accepted, as we found this was a good tradeoff.

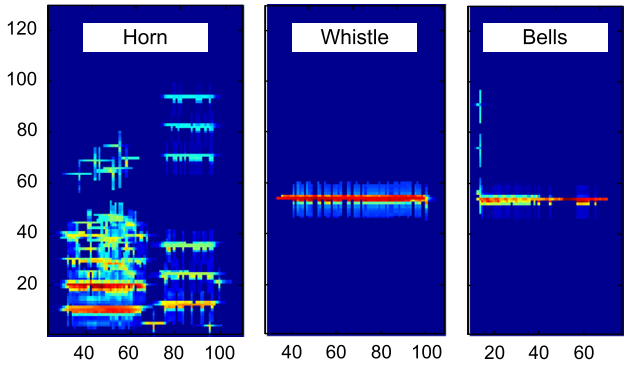
If the hypothesis is accepted, the given sound class X_y is considered to have been detected at the onset time, $t_{ON,y}$. The keypoints that contributed to the hypothesis are then removed from further matches, and the next best hypothesis is evaluated until all valid hypotheses in the clip have been tested.

4.5. Sound event reconstruction

Given a hypothesis, we can reconstruct the observed sound event from the matching keypoints and clusters that contributed to the hypothesis score. An example of this is shown in Fig. 5, where each of the three sounds are recognised, and can be reconstructed by averaging the codebook clusters that were matched for each keypoint. However, here our focus is on the recognition of the sound, hence we leave reconstruction for future work.



(a) Example of three overlapping sounds with non-stationary background noise. Here, the harmonic of the bell sound is covered by the whistle around frame 40, which in turn is overlapped with the horn which is present between frames 40-100.



(b) Spectrogram segmentations, using the detected keypoint-clusters, of the three sounds detected using our approach

Fig. 5. Example recognition of three overlapping sounds to demonstrate that our approach is not limited to a fixed number of overlapping sounds.

5. Experiments

5.1. Database

For our experiments, we generate a database of overlapping sound events from the Real Word Computing Partnership (RWCP) Sound Scene Database (Nakamura et al., 2000). We select the following five classes: horn, bells5, bottle1, phone4 and whistle1. Amongst the sounds, the bottle1 class contains the most variation, with five different bottles being struck by two different objects, although there is some variation across all classes. From this, we generate all 15 overlapping combinations, each consisting of one or two sound events, using randomly chosen onset times ensuring between 50% and 100% temporal overlap.

5.2. Baseline comparison

For comparison with our proposed method, we implement two frame-based baseline classification approaches. It is notable that our proposed method performs recognition as opposed to the simpler task of classification, however we choose these as they provide a well-performing benchmark.

The first approach we call MixMax-GMM, which can be seen as a simplification of the FHMM approach (Roweis, 2003) using one state. Here, for overlapped class $Z = X + Y$, where X and Y are two clean classes, the PDF, p_Z, of class Z can be decomposed as follows (Nádas et al., 1989):

$$p_Z(\alpha) = p_X(\alpha)c_Y(\alpha) + p_Y(\alpha)c_X(\alpha) \tag{27}$$

where c_X is the cumulative density function (CDF) of class X and α represents the 36 dimension log-power Mel-frequency spectral coefficient (MFSC) features. We model the PDF using a 6-component GMM, and take the maximum log-likelihood summed across all frames in the clip as the classification result.

The second baseline we call Overlap-SVM, which is based on the approach proposed by Temko and Nadeu (2009). Here, the mean and variance of the 60-dimension frame-based features is taken over the clip, giving a final feature with 120 dimensions. This is combined with a two-stage SVM, where the first contains the isolated classes and an amalgamated "overlapping" class. However, we simplify the tree-SVM structure for the overlapped class combinations using a one-against-one classification, as we found there was insufficient training data to benefit from the full tree-structure.

5.3. Experimental methods

For training, we randomly select 20 isolated samples of each sound event from the database. Overlap-SVM additionally requires 20 samples for each of the 10 overlapping combinations, which we generate from the isolated samples selected.

For testing, we generate 50 overlapping samples for each of the 15 overlapping combinations, using samples excluded from the training set. We then investigate the performance under the following conditions:

1. **Clean**: both isolated and overlapping.
2. **Mismatched noise**: We add "Factory Floor 1" noise, from the NOISEX'92 database (Varga and Steeneken, 1993), to the testing samples at 20, 10 and 0 dB SNR. This noise is chosen for its challenging, non-stationary nature.
3. **Change of Volume**: We pre-multiply the waveform by the factors $\{0.5, 0.75, 1, 1.5, 2\}$ prior to taking the STFT of the signal, to simulate a channel transfer function.

As evaluation measure, we calculate the recognition accuracy (TP) and false alarm (FA) over each of the sound classes, over 5 runs

Table 1
Experimental results across the various testing conditions, where the best performing method for each condition is highlighted in bold. The values for TP/FA (%) are averaged over 5 runs of the experiments. For the isolated experiment, the results are averaged over the 5 sound classes, while for the overlapping experiments, the results are averaged over the 15 overlap combinations.

Experiment setup		Proposed LSF		Overlap-SVM		MixMax-GMM	
		TP	FA	TP	FA	TP	FA
Isolated	Clean	99.3 ± 2.7	0.4 ± 2.4	**100** ± 0.0	1.5 ± 3.4	99.6 ± 1.4	1.3 ± 5.8
Overlapping		**98.0** ± 3.4	0.8 ± 3.6	96.5 ± 7.3	1.3 ± 2.8	84.0 ± 29.3	5.2 ± 17.0
Overlapping: Added noise	20 dB	**97.2** ± 5.0	0.7 ± 3.2	76.9 ± 39.0	18.6 ± 35.1	52.8 ± 44.9	27.8 ± 42.6
	10 dB	**95.5** ± 9.1	0.9 ± 3.5	74.7 ± 40.9	20.9 ± 36.8	37.8 ± 42.9	25.1 ± 41.2
	0 dB	**90.2** ± 17.6	2.5 ± 8.2	65.7 ± 41.9	25.8 ± 36.1	22.9 ± 38.8	20.9 ± 35.7
Overlapping: Clean with volume change	×0.5	**98.1** ± 3.0	0.7 ± 3.3	84.0 ± 24.8	1.5 ± 5.0	56.0 ± 43.4	12.4 ± 27.8
	×0.75	**98.4** ± 2.9	0.5 ± 1.8	92.8 ± 13.1	1.1 ± 2.9	80.6 ± 30.0	4.4 ± 13.7
	×1.5	**98.4** ± 2.7	0.6 ± 2.1	95.9 ± 9.9	4.0 ± 11.4	82.0 ± 29.8	8.3 ± 21.6
	×2	**98.0** ± 3.3	0.7 ± 2.0	94.1 ± 14.7	7.0 ± 18.3	68.7 ± 40.7	23.7 ± 39.3
Average		**97.0%**	**0.9%**	86.7%	9.1%	64.9%	14.3%

Table 2
Detailed experimental results for the LSF method in 10 dB Factory Floor noise, showing the results for each of the 15 overlapping combinations. The values (%) represent the percentage of clips with the detected sound event. Correct TP detections are highlighted in bold.

Sound	Horn	Horn				Bells	Bells			Bottle	Bottle		Phone	Phone	Whistle
Event		Bells	Bottle	Phone	Whistle		Bottle	Phone	Whistle		Phone	Whistle		Whistle	
Horn	**100**	**100**	**100**	**95.6**	**99.6**	0	0	10.8	0	0	8.4	0	10.8	6.8	0
Bells	0	**97.2**	0	0	0.4	**100**	**100**	**63.2**	**80.8**	0	0	3.2	0	0	0
Bottle	0	0	**98.4**	1.2	0	0	**96.0**	0.8	0	**100**	**82.2**	**96.8**	1.2	0.8	0
Phone	0	0	0	**100**	0	0	0	**100**	0	0	**100**	0	**100**	**100**	0
Whistle	0	0	0	0	**95.6**	0	0	0.4	**94.4**	0	0	**95.6**	0	**93.6**	**96.8**

of the experiment. TP is calculated as the ratio of correct detections to the number of clips containing occurrences of that class. Analogously, FA is the ratio of incorrect detections to the number of clips not containing that class.

5.4. Experimental results

The results for clean conditions can be found in Table 1. For isolated sounds, it can be seen that our proposed LSF approach performs well compared to the baseline, achieving a TP of 99.3% for an FA of only 0.4%. Here, although the TP is marginally lower than the two baselines, the average FA is an improvement of around 1%. For overlapping sounds, our proposed method outperforms the two baseline methods, achieving a TP of 98.0% and an FA of only 0.8%. This is an improvement in TP of 1.5% over the Overlap-SVM baseline, and significantly better than the MixMax-GMM approach, which achieves a TP of just 84.0%. Our result is also significant, considering that the Overlap-SVM requires overlapping sounds samples for training, hence is performing classification in matched training and testing conditions.

Under mismatched noise conditions, the results in Table 1 show that the performance of both baseline methods declines rapidly with increasing noise. Our proposed LSF approach performs consistently well across all four conditions, and can still achieve a TP of 90.2% in 0 dB conditions, for an FP of just 2.5%. One reason for the poor baseline performance is that frame-based features contain a mixture of overlapping signals and noise that occur at the same time instance. Our system overcomes this by using features that are local across frequency, hence recognition can still be performed in the presence of competing signals. Further analysis of the LSF results can be found in Table 2, which shows the average results for each overlapping combination in 10 dB noise. It can be seen that our proposed LSF approach had the most difficulty identifying other sounds in mixtures containing the phone sound, particularly the bottle sound. This is because the sound contains a broad range of frequencies meaning that fewer keypoints for the overlapped class will be detected and assigned to the correct clusters.

For the final experiment with changing volume, the results in Table 1 show that our LSF method performs consistently well, maintaining a low FA, whereas the performance of the baseline methods drops and the FA increases. Comparing the baselines, Overlap-SVM performs relatively well compared to MixMax-GMM, as it includes perceptual features that are unaffected by the change in volume. However, our method can estimate the unknown transfer function using the GHT voting, hence the performance is similar to that obtained in matched conditions.

5.5. Three overlapping sounds

As a final example, in Fig. 5 we demonstrate the ability of our proposed LSF system, without modification, to perform recognition of more than two overlapping sounds. For the baseline methods, this is not possible, as the Overlap-SVM requires training on all expected combinations in advance, and the MixMax-GMM is currently only derived for two overlapping combinations. The spectrogram reconstructions in Fig. 5b show that our approach can recognise each of the three sounds, with very few LSFs incorrectly attributed to the wrong sound.

6. Conclusion

In this paper we propose a method for recognition of sound events in challenging overlapping and noisy conditions. We take inspiration from human perception, where it has been suggested that human hearing is based on local information, and also from image object recognition, which has parallels with overlapping SER. Our approach is to detect keypoints in the spectrogram, and then characterise the sound jointly through the LSF and the keypoint distribution relative to the sound onset. The keypoint distribution is then used as a voting function for the GHT, which can detect an arbitrary combination of sound events in challenging

mismatched conditions. For future work, we aim to investigate the use of local features for reconstruction of the overlapping sounds. While the codebook can be used to reconstruct the recognised keypoints, it remains to find a way to reconstruct missing areas of the spectrogram from the trained model.

References

Allen, J., 1994. How do humans process and recognize speech? IEEE Trans. Speech Audio Process. 2, 567–577.

Baggenstoss, P.M., 2002. Statistical modeling using Gaussian mixtures and HMMs with Matlab. Tech. rep., Naval Undersea Warfare Center, Newport, RI. <http://class-specific.com/csf/pdf.html>.

Ballard, D., 1981. Generalizing the hough transform to detect arbitrary shapes. Pattern Recognit. 13, 111–122.

Bardeli, R., Wolff, D., Kurth, F., Koch, M., Tauchert, K., Frommolt, K., 2010. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. Pattern Recognition Lett. 31, 1524–1534.

Barnwal, S., Sahni, K., Singh, R., Raj, B., 2012. Spectrographic seam patterns for discriminative word spotting, in: IEEE Internat. Conf. on Acoustics Speech and Signal Process. (ICASSP), 2012, IEEE. pp. 4725–4728.

Carreira-Perpinan, M., 2000. Mode-finding for mixtures of gaussian distributions. IEEE Trans. Pattern Anal. Machine Intell. 22, 1318–1323.

Casey, M., 2001. Mpeg-7 sound-recognition tools. IEEE Trans. Circuits Systems Video Technol. 11, 737–747.

Cooke, M., 2006. A glimpsing model of speech perception in noise. J. Acoust. Soc. Amer. 119, 1562–1573.

Cooke, M., Green, P., Josifovski, L., Vizinho, A., 2001. Robust automatic speech recognition with missing and unreliable acoustic data. Speech Comm. 34, 267–285.

Cowling, M., Sitte, R., 2003. Comparison of techniques for environmental sound recognition. Pattern Recognition Lett. 24, 2895–2907.

Dennis, J., Tran, H., Chng, E., 2012. Overlapping sound event recognition using local spectrogram features with the generalised hough transform, in: Proc. Interspeech 2012.

Dennis, J., Tran, H., Li, H., 2011. Spectrogram image feature for sound event classification in mismatched conditions. IEEE Signal Process. Lett. 18, 130–133.

Dessein, A., Cont, A., Lemaitre, G., 2012. Real-time detection of overlapping sound events with non-negative matrix factorization. In: Nielsen, F., Bhatia, R. (Eds.), . Matrix Information Geometry. Springer, pp. 341–371.

Gerosa, L., Valenzise, G., Antonacci, F., Tagliasacchi, M., Sarti, A., 2007. Scream and gunshot detection in noisy environments, in: 15th European Signal Process. Conf. (EUSIPCO-07), Sep. 3–7, Poznan, Poland.

Heckmann, M., Domont, X., Joublin, F., Goerick, C., 2011. A hierarchical framework for spectro-temporal feature extraction. Speech Comm. 53, 736–752.

Heittola, T., Mesaros, A., Virtanen, T., Eronen, A., 2011. Sound event detection in multisource environments using source separation, in: Workshop on Machine Listening in Multisource, Environments, pp. 36–40.

Kleinschmidt, M., Gelbart, D., 2002. Improving word accuracy with gabor feature extraction, in: Proc. ICSLP, pp. 16–38.

Lehmann, A., Leibe, B., Van Gool, L., 2011. Fast prism: branch and bound hough transform for object class detection. Internat. J. Comput. Vision 94, 175–197.

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. Internat. J. Comput. Vision 60, 91–110.

Lyon, R., 2010. Machine hearing: an emerging field. IEEE Signal Process. Mag. 27, 131–139.

Marchand, S., Vialard, A., et al., 2009. The hough transform for binaural source localization, in: Proc. of the Digital Audio Effects (DAFx09) Conf., pp. 252–259.

Matsui, T., Goto, M., Vert, J., Uchiyama, Y., 2011. Gradient-based musical feature extraction based on scale-invariant feature transform, in: 19th European Signal Process. Conf. (EUSIPCO 2011), pp. 724–728.

Mikolajczyk, K., Schmid, C., 2004. Scale & affine invariant interest point detectors. Internat. J. Comput. Vision 60, 63–86.

Nádas, A., Nahamoo, D., Picheny, M., 1989. Speech recognition using noise-adaptive prototypes. IEEE Trans. Acoustics Speech Signal Process. 37, 1495–1503.

O'Shaughnessy, D., 2008. Invited paper: automatic speech recognition: history, methods and challenges. Pattern Recognit. 41, 2965–2979.

Raj, B., Stern, R., 2005. Missing-feature approaches in speech recognition. IEEE Signal Process. Mag. 22, 101–116.

Roweis, S., 2003. Factorial models and refiltering for speech separation and denoising, in: Proc. EuroSpeech, Geneva, pp. 1009–1012.

S. Nakamura, et al., 2000. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition, in: Proc. ICLRE, pp. 965–968.

Schutte, K., 2009. Parts-based Models and Local Features for Automatic Speech Recognition. Ph.D. thesis. Massachusetts Institute of Technology.

Temko, A., Nadeu, C., 2009. Acoustic event detection in meeting-room environments. Pattern Recognition Lett. 30, 1281–1288.

Tran, H., Li, H., 2011. Jump function kolmogorov for overlapping audio event classification, in: IEEE Internat. Conf. Acoustics Speech Signal Process. (ICASSP), 2011, IEEE. pp. 3696–3699.

Varga, A., Steeneken, H., 1993. Assessment for automatic speech recognition: Ii. noisex-92: a database and an experiment to study the effect of additive noise on speech recognition systems. Speech Comm. 12, 247–251.

Wang, D., 2005. On ideal binary mask as the computational goal of auditory scene analysis. Speech Sep. by Humans Machines, 181–197.

Zhuang, X., Zhou, X., Hasegawa-Johnson, M., Huang, T., 2010. Real-world acoustic event detection. Pattern Recognition Lett. 31, 1543–1551.