Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering

Saman Mousazadeh and Israel Cohen, *Senior Member, IEEE*

***Abstract*—Voice activity detection has attracted significant research efforts in the last two decades. Despite much progress in designing voice activity detectors, voice activity detection (VAD) in presence of transient noise is a challenging problem. In this paper, we develop a novel VAD algorithm based on spectral clustering methods. We propose a VAD technique which is a supervised learning algorithm. This algorithm divides the input signal into two separate clusters (i.e., speech presence and speech absence frames). We use labeled data in order to adjust the parameters of the kernel used in spectral clustering methods for computing the similarity matrix. The parameters obtained in the training stage together with the eigenvectors of the normalized Laplacian of the similarity matrix and Gaussian mixture model (GMM) are utilized to compute the likelihood ratio needed for voice activity detection. Simulation results demonstrate the advantage of the proposed method compared to conventional statistical model-based VAD algorithms in presence of transient noise.**

***Index Terms*—Gaussian mixture model, spectral clustering, transient noise, voice activity detection.**

I. Introduction

SPEECH/NON-SPEECH classification is an unsolved problem in speech processing and affects diverse applications including robust speech recognition [1], [2], discontinuous transmission [3], real-time speech transmission on the Internet [4] or combined noise reduction and echo cancellation schemes in the context of telephony [5]. Elementary methods for voice activity detection such as G.729 standard [3], calculate line spectral frequencies, full-band energy, low-band energy (< 1 kHz), and zero-crossing rate. Each frame is then simply classified using a fixed decision boundary in the space defined by these features. Smoothing and adaptive correction can be applied to improve the estimate. Although these methods have acceptable performance when applied to clean signals, their performance essentially degrades in noisy environments even in moderately high signal to noise ratios (SNRs). To overcome this shortcoming, several statistical model-based VAD algorithms have been proposed in the last two decades. Sohn *et al.* [6] assumed that the spectral coefficients of the noise and speech signal can be modeled as complex Gaussian random variables, and developed a VAD algorithm based on the likelihood ratio test (LRT). Following their work, many researchers tried to improve the performance of model-based VAD algorithms by assuming different statistical models for speech signals, see [7]–[12] and references therein. While these methods have superior performances in presence of stationary noise over the elementary methods, their performances degrade significantly in presence of transient noise such as coughing, sneezing, keyboard typing, and door knocking sounds. This means that with high probability, these sounds are detected as speech.

VAD is usually a preprocessing step in speech processing applications such as speech or speaker recognition. A straightforward application of VAD would be an automatic camera steering task. Suppose a scenario in which there exist multiple speakers with a camera assigned to each of them (a popular example can be videoconferencing). The camera must be steered to the dominant speaker automatically. While stationary noise can be treated very well using a statistical mode-based method, transient noise could be very annoying [13]. This means that a silent speaker might be identified as a dominant speaker while he/she is just typing or there is a knock on the door. Hence, finding a VAD algorithm which is robust to transient noise would be of practical interest.

VAD can be regarded as an acoustic event detection (AED) task which detects some acoustical event including transient noise, e.g., door knocking, footsteps, etc. Current most prominent works in AED reflect the aim of bringing the most successful technologies of speech recognition to the field. Zhou *et al.* [14] implemented a hidden Markov model (HMM)-based AED system with lattice rescoring using a feature set selected by AdaBoost based approach. Haung *et al.* [15] improved AED via audio-visual intermediate integration using generalizable visual features. Using optical flow based spatial pyramid histograms, they proposed a method for representing the highly variant visual cues of the acoustic events. Espi *et al.* [16] introduced the usage of spectro-temporal fluctuation features in a tandem connectionist approach, modified to generate posterior features separately for each fluctuation scale and then combine the streams to be fed to a classic Gaussian mixture model-hidden Markov model (GMM-HMM) procedure. Voice activity detection can also be regarded as a clustering problem, in which the goal is to classify the input signal into speech absence and speech presence frames. Hence, after choosing an appropriate feature space, one can use a clustering algorithm to obtain a VAD algorithm. Among different clustering methods, spectral clustering has recently become one of the most popular modern clustering algorithms. It is simple to implement, can be

Manuscript received April 22, 2012; revised August 23, 2012; accepted February 05, 2013. Date of publication February 22, 2013; date of current version March 13, 2013. This work was supported by the Israel Science Foundation under Grant 1130/11. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Nobutaka Ono.

The authors are with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: smzadeh@tx.technion.ac.il; icohen@ee.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2248717

solved efficiently by standard linear algebra software, and very often outperforms the traditional clustering algorithms such as the k-means algorithm. Recently, spectral clustering has been utilized by several authors in signal processing applications such as image segmentation [17], [18], speech separation [19], and clustering of biological sequence data [20] just to name a few.

In this paper, we present a novel voice activity detection algorithm using spectral clustering. In particular, we use a normalized spectral clustering algorithm [21] to cluster the Mel-frequency cepstrum coefficients (MFCC) of the received signal into two different clusters, i.e., speech presence and speech absence. The clustering problem can be done using GMM. However, fitting a GMM to high dimensional data generally requires a great amount of training data, and as the number of Gaussian mixtures is increased, we need more and more training data to fit the GMM to high dimensional data. The fact that the distribution of natural data, like speech and transient noise is non-uniform and concentrates around low-dimensional structures [22], motivates us to exploit the shape (geometry) of the distribution for efficient learning. Among different dimensionality reduction techniques, "kernel eigenmap methods" such as local linear embedding [23], Laplacian eigenmaps [24], Hessian eigenmaps [25], and diffusion maps [26] (just to name a few) have recently attracted much research attention. These algorithms exhibit two major advantages over classical dimensionality reduction methods (such as principal component analysis or classical multidimensional scaling): They are nonlinear, and they preserve local structures. The first aspect is essential as most of the time, in their original form, the data points do not lie on linear manifolds. The second point is related to the fact that in many applications, distances of points that are far apart are meaningless, and therefore need not to be preserved. The main idea of these methods is to use the dominant eigenvectors of Laplacian of the similarity matrix as the new lower dimension representation of the data.

Our proposed algorithm is a supervised learning algorithm. One must train the system before it can be used. Training data is used for estimating the parameters of the kernel used in computation of the similarity matrix. The data is also used in finding two Gaussian mixture models for modeling the first two eigenvectors of the Laplacian of the similarity matrix corresponding to the first two leading eigenvalues of normalized Laplacian matrix. This means that we model the low dimensional representation of the original data (i.e., MFCC) using two different GMMs, one for each cluster. Upon receiving new unlabeled data, the optimum parameters of the kernel are utilized to find the similarity between the new data and the training set in order to find the low dimensional representation of new data. Using the GMMs obtained in the training step, the likelihood ratio is computed, and the final VAD is obtained by comparing that likelihood ratio to a threshold.

The rest of this paper is organized as follows. In Section II, we formulate our problem and introduce a novel VAD for online processing. Simulation results and performance comparison are presented in Section III. Finally, we conclude the paper in Section IV.

II. Problem Formulation

In this section, we propose our voice activity detection method, which is based on spectral clustering. The basic idea behind spectral clustering method is to use several eigenvectors of the normalized Laplacian of the similarity matrix as a new low dimension representation of the high dimension data points. Clustering is generally performed on this new representation of the data points using a conventional (weighted) k-means algorithm [19]. Here we introduce a novel technique for clustering the data based on GMM modeling of the eigenvectors of the normalized Laplacian of the similarity matrix.

Every clustering problem consists of the following three main stages: selecting an appropriate feature space, choosing a metric as a notion of similarity between data-points, and selecting the clustering algorithm. In what follows, we discuss each of these stages for voice activity detection in more detail.

A. Feature Selection

Let $x_{\rm sp}(n)$ denote a speech signal and let $x_{\rm tr}(n)$ and $x_{\rm st}(n)$ be the additive contaminating transient and stationary noise signals, respectively. The signal measured by a microphone is given by:

$$y(n) = x_{\rm sp}(n) + x_{\rm tr}(n) + x_{\rm st}(n). \tag{1}$$

The goal is to determine whether there exists speech signal in a given time frame (each approximately 16–20 msec long). Here we choose absolute value of MFCCs and the arithmetic mean of the log-likelihood ratios for the individual frequency bins as our feature space. More specifically, let $\boldsymbol{Y}_m(t,k)(t = 1,2,\ldots,N; k = 1,2,\ldots,K_m)$ and $\boldsymbol{Y}_s(t,k)(t = 1,2,\ldots,N; k = 1,2,\ldots,K_s)$ be the absolute value of the MFCC and the STFT coefficients in a given time frame, respectively. MFCC and the STFT coefficients are computed in K_m and K_s frequency bins, respectively. Then, each frame is represented by a $(K_m + 1)$-dimension column vector defined as follows:

$$\boldsymbol{Y}(:,t) = \begin{bmatrix} \boldsymbol{Y}_m(:,t) \\ \Lambda_t \end{bmatrix} \tag{2}$$

where $\boldsymbol{Y}_m(:,t)$ is the t-th column of $\boldsymbol{Y}_m$, and Λ_t is the arithmetic mean of the log-likelihood ratios for the individual frequency bands in frame t, which is given by:

$$\Lambda_t = \frac{1}{K_s}\sum_{k=1}^{K_s}\left(\frac{\gamma_k(t)\xi_k(t)}{1+\xi_k(t)} - \log\left(1+\xi_k(t)\right)\right) \tag{3}$$

where $\xi_k(t) = \lambda_s(t,k)/\lambda_N(t,k)$ is called *a priori* SNR, which can be estimated using decision-directed method [27], and $\lambda_s(t,k)$ is the variance of speech signal in the k-th frequency bin of the t-th frame; $\gamma_k(t) = |\boldsymbol{Y}_s(t,k)|^2/\lambda_N(t,k)$ is called the *a posteriori* SNR, ϵ is the kernel width obtained during the training phase, and $\lambda_N(t,k)$ is the variance of stationary noise in t-th time frame and k-th frequency bin, which can

be estimated from training data (if there exist sequences consisting of only stationary noise) or by improved minima controlled recursive averaging (IMCRA) [28]. The reason behind choosing this feature space for each frame is as follows. The likelihood ratio has been long exploited as a feature for voice activity detection in presence of stationary noise [6]–[10]. The Mel-frequency cepstrum coefficient is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. MFCCs are commonly used as features in speech recognition systems. Combining these two features appropriately would be a suitable feature space for voice activity detection in presence of transient noise. See Section III for further discussion.

B. Clustering Algorithm

A popular way for representing the data is to build a similarity graph, which is a weighted graph $G = (V, E)$, where V is the set of the vertices and E is the set of the edges of the graph. Each vertex v_i in this graph represents a data point $\boldsymbol{Y}(:, i)$. Each edge e_{ij} between two vertices v_i and v_j carries a non-negative weight $\boldsymbol{W}(i, j) \geq 0$, which is a measure of similarity between the corresponding points. We assume that the graph G is an undirected one (i.e., $\boldsymbol{W}(i, j) = \boldsymbol{W}(j, i)$). A similarity matrix $\boldsymbol{W}$ is a matrix whose (i, j)-th element equals to $\boldsymbol{W}(i, j)$.

Using the concept of subspace comparison, Bach and Jordan [19] proposed a spectral clustering algorithm using the eigenvectors of matrix $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ where $\boldsymbol{D}$ is a diagonal matrix whose i-th diagonal element equals to $\sum_{j=1}^{N} \boldsymbol{W}(i, j)$ (i.e., $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{W}1)$ where $\mathbf{1}$ is a column vector of ones). More specifically, let K be the number of clusters and $\boldsymbol{U}$ be a matrix consisting of the first K eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ corresponding to K's largest eigenvalues of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$. The clustering is done either by running a weighted k-means algorithm on $\boldsymbol{U}$ where each point is represented by a row of $\boldsymbol{U}$ or by running a k-means algorithm on $\boldsymbol{V} = \boldsymbol{D}^{1/2}\boldsymbol{U}\left(\boldsymbol{U}^T\boldsymbol{D}\boldsymbol{U}\right)^{-1}$ where each point is represented by a row of $\boldsymbol{V}$. The most important drawback of this method is that it does not prepare a toll for controlling the tradeoff between probability of false alarm and probability of detection. We will deal with this issue by using GMM modeling of the eigenvectors (see testing algorithm below for further discussion).

The most important part of a spectral clustering algorithm is the calculation of the similarity matrix. Although the definition of the similarity between points is an application and data dependent, a popular way of defining the similarity matrix is to use a Gaussian kernel as follows:

$$\boldsymbol{W}(i, j) = \exp\left(-\frac{\|\boldsymbol{Y}(:, i) - \boldsymbol{Y}(:, j)\|_2^2}{\sigma}\right) \tag{4}$$

where $\boldsymbol{Y}(:, i)$ is the i-th data point. The selection of σ is commonly done manually. Ng *et al.* [29] suggested selecting σ automatically by running their clustering algorithm repeatedly for a number of values of σ and selecting the one that provides the least distorted clusters. In [30] and [31], it was suggested to automatically set the scale by examining a logarithmic scale of the sum of the kernel weights without computing the spectral decomposition of the transition matrix. Zelnik-Manor *et al.* [17] suggested calculating a local scaling parameter σ_n for each data point instead of selecting a single scaling parameter σ. The above mentioned methods are somehow heuristic or hard to implement because of high computational load. Bach and Jordan [19] introduced a method for estimating the parameters of the kernel (not necessarily Gaussian kernel) based on minimization of a cost function that characterizes how close the eigenstructure of the similarity matrix $\boldsymbol{W}$ is to the true partition.

Our voice activity detection algorithm is a supervised learning one. As a consequence, one must utilize training data in order to adjust the parameters of the algorithm and use those parameters for clustering unlabeled data. In the next two subsections, we illustrate how each of these stages works.

1) Learning Algorithm: In this section, we introduce our learning algorithm based on the method presented in [19]. Suppose that we have a database of clean speech signal, a database of transient noise, and a database of stationary noise. We choose L different signals from each database and combine them as follows. Let $x_{\mathrm{sp}}^{\ell}(n)$, $x_{\mathrm{tr}}^{\ell}(n)$, $x_{\mathrm{st}}^{\ell}(n)$ be the ℓ-th speech signal, transient noise, and stationary noise, respectively. Without loss of generality, we assume that all of these signals are of the same length (i.e., N_ℓ). We build the ℓ-th training sequence, $\boldsymbol{Y}^{\ell} \in \mathbb{R}^{K_m+1\times 3N^{\ell}}$, as follows. Let

$$x_1^{\ell}(n) = x_{\mathrm{sp}}^{\ell}(n) + x_{\mathrm{st}}^{\ell}(n), \tag{5}$$

$$x_2^{\ell}(n) = x_{\mathrm{tr}}^{\ell}(n) + x_{\mathrm{st}}^{\ell}(n), \tag{6}$$

$$x_3^{\ell}(n) = x_{\mathrm{sp}}^{\ell}(n) + x_{\mathrm{tr}}^{\ell}(n) + x_{\mathrm{st}}^{\ell}(n), \tag{7}$$

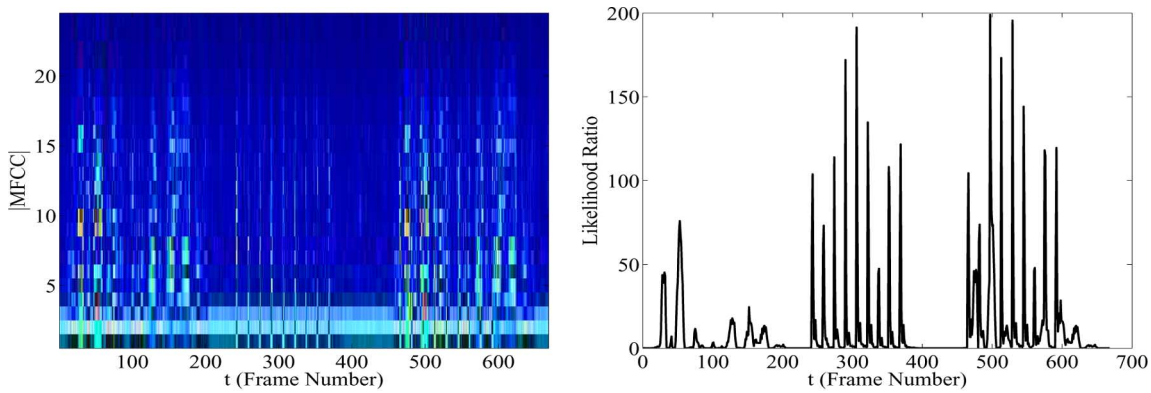
and let $\boldsymbol{Y}_1^{\ell}$, $\boldsymbol{Y}_2^{\ell}$, and $\boldsymbol{Y}_3^{\ell}$ be the feature matrix extracted using (2) and (3) from $x_1^{\ell}(n)$, $x_2^{\ell}(n)$, and $x_3^{\ell}(n)$, respectively. Then, the ℓ-th training data is obtained by concatenating these matrices as follows:

$$\boldsymbol{Y}^{l} = \left[\boldsymbol{Y}_1^{\ell} : \boldsymbol{Y}_2^{\ell} : \boldsymbol{Y}_3^{\ell}\right]. \tag{8}$$

A typical training sequence is depicted in Fig. 1. For each of these training sequences, we compute the indicator matrix of the partitions $\boldsymbol{C}^{\ell} \in \mathbb{R}^{3N^{\ell}\times 4}$ using (9), where $\boldsymbol{C}_{ij}^{\ell}$ is the (i, j)-th element of $\boldsymbol{C}^{\ell}$, $\chi(\cdot)$ is an indicator function that equals to one if its argument is true and zero otherwise, and T_{sp} and T_{tr} are speech and transient noise thresholds and are chosen as the maximum value of threshold such that thresholding the speech or transient noise has no significant auditory effect, $\oplus$ and $\otimes$ are logical **OR** and logical **AND** operators, respectively. $P(\cdot)$ is a power calculation operator defined by:

$$P(\boldsymbol{X}_{\mathrm{sp}}^{\ell}(:, i)) = \frac{1}{K_s}\sum_{k=1}^{K_s} \|\boldsymbol{X}_{\mathrm{sp}}^{\ell}(k, i)\|_2^2 \tag{10}$$

$$P(\boldsymbol{X}_{\mathrm{tr}}^{\ell}(:, i)) = \frac{1}{K_s}\sum_{k=1}^{K_s} \|\boldsymbol{X}_{\mathrm{tr}}^{\ell}(k, i)\|_2^2. \tag{11}$$



(a) (b)

Fig. 1. Typical training sequence consisting of speech signal and keyboard stoke noise (transient noise) corrupted by babble noise (absolute value of MFCC (right) likelihood ratio (left)).

where $\boldsymbol{X}^{\ell}_{\rm sp}(:,i)$ and $\boldsymbol{X}^{\ell}_{\rm tr}(:,i)$ are the STFT coefficients of $x^{\ell}_{\rm sp}(n)$ and $x^{\ell}_{\rm tr}(n)$ in the i-th frame, respectively.

The last and most important object to be defined, in order to use the training algorithm presented in [19], is the parametric similarity matrix, i.e., $\boldsymbol{W}^{\ell}_{\boldsymbol{\theta}}$. For our problem, we define this matrix as follows:

$$\boldsymbol{W}^{\ell}_{\boldsymbol{\theta}}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_p \boldsymbol{Q}(i+p, j+p)\right) \tag{12}$$

$$\boldsymbol{Q}(i,j) = \Bigg\| \boldsymbol{Y}^{\ell}_{m}(:,i)\left(1-\exp\left(\frac{-\Lambda^{\ell}_{i}}{\epsilon}\right)\right) - \boldsymbol{Y}^{\ell}_{m}(:,j)\left(1-\exp\left(\frac{-\Lambda^{\ell}_{j}}{\epsilon}\right)\right)\Bigg\|_2^2 \tag{13}$$

where $\boldsymbol{\theta} = [\epsilon, \alpha_{-P}, \alpha_{-P+1}, \ldots, \alpha_{P-1}, \alpha_P] \in \mathbb{R}^{2P+2}$ is the vector of parameters, $\boldsymbol{Y}^{\ell}_{m}(:,i)$ and Λ^{ℓ}_{i} are the absolute value of the MFCC and the likelihood ratio of the ℓ-th training sequence in the i-th frame, respectively, and $\|\cdot\|_2$ is the Euclidian norm of a vector. The reason behind choosing this weight function is discussed in the following paragraph.

For designing an appropriate weight matrix, we have taken the following two points into consideration. The first one was the similarity between two individual frames, and the second one was the effect of neighboring frames on deciding whether a specific frame contains speech or transient noise. Combining these two features (i.e., MFCC and likelihood ratio) as in (12)–(13), results in a good metric as a similarity notion between two frames for voice activity detection in presence of transient noise. More specifically, if there exists speech signal or transient noise in a specific frame, the value of likelihood ratio is large (see Fig. 1 (right)); hence, the exponential term in (13) approximately equals to zero, and the feature for that frame will be approximately the MFCCs. On the other hand, if a specific frame consists of only stationary noise, then the likelihood ratio will be small, and the exponential term in (2) approximately equals to one. Consequently, the feature vector will approximately be equal to zero vector for those frames that only contain stationary noise. Considering the MFCCs of a typical training sequence as depicted in Fig. 1 (left), it is apparent that there might exist two frames in the speech part (left part of the figure) and the transient noise part (middle part of the figure) that are very similar to each other (in the sense that their Euclidean distance

$$\boldsymbol{C}^{\ell}_{ij} = \begin{cases}
\boldsymbol{C}^{\ell}_{i1} = 1 - \boldsymbol{C}^{\ell}_{i4} & \text{Only stationary noise} & 1 < i \le N^{\ell} \\
\boldsymbol{C}^{\ell}_{i2} = 0 & \text{Transient plus stationary noise} & 1 < i \le N^{\ell} \\
\boldsymbol{C}^{\ell}_{i3} = 0 & \text{Speech plus transient plus stationary noise} & 1 < i \le N^{\ell} \\
\boldsymbol{C}^{\ell}_{i4} = \chi(P(\boldsymbol{X}^{\ell}_{\rm sp}(:,i)) > T_{\rm sp}) & \text{Speech plus stationary noise} & 1 < i \le N^{\ell} \\
\boldsymbol{C}^{\ell}_{i1} = 1 - \boldsymbol{C}^{\ell}_{i2} & \text{Only stationary noise} & N^{\ell} < i \le 2N^{\ell} \\
\boldsymbol{C}^{\ell}_{i2} = \chi(P(\boldsymbol{X}^{\ell}_{\rm tr}(:,i)) > T_{\rm tr}) & \text{Transient plus stationary noise} & N^{\ell} < i \le 2N^{\ell} \\
\boldsymbol{C}^{\ell}_{i3} = 0 & \text{Speech plus transient plus stationary noise} & N^{\ell} < i \le 2N^{\ell} \\
\boldsymbol{C}^{\ell}_{i4} = 0 & \text{Speech plust stationary noise} & N^{\ell} < i \le 2N^{\ell} \\
\boldsymbol{C}^{\ell}_{i1} = 1 - \left(\boldsymbol{C}^{\ell}_{i2} \oplus \boldsymbol{C}^{\ell}_{i3} \oplus \boldsymbol{C}^{\ell}_{i4}\right) & \text{Only stationary noise} & 2N^{\ell} < i \le 3N^{\ell} \\
\boldsymbol{C}^{\ell}_{i2} = \chi(P(\boldsymbol{X}^{\ell}_{\rm tr}(:,i)) > T_{\rm tr}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\rm sp}(:,i)) < T_{\rm sp}) & \text{Transient plus stationary noise} & 2N^{\ell} < i \le 3N^{\ell} \\
\boldsymbol{C}^{\ell}_{i3} = \chi(P(\boldsymbol{X}^{\ell}_{\rm tr}(:,i)) > T_{\rm tr}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\rm sp}(:,i)) > T_{\rm sp}) & \text{Speech plus transient plus stationary noise} & 2N^{\ell} < i \le 3N^{\ell} \\
\boldsymbol{C}^{\ell}_{i4} = \chi(P(\boldsymbol{X}^{\ell}_{\rm tr}(:,i)) < T_{\rm tr}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\rm sp}(:,i)) < T_{\rm sp}) & \text{Speech plus stationary noise} & 2N^{\ell} < i \le 3N^{\ell}
\end{cases} \tag{9}$$

is small) but belong to two different clusters. The characteristic that distinguishes the frames containing speech from those frames containing transient noise is that the neighboring frames of a specific speech frame are almost the same, which is not true for transient noise. Choosing the weight function as in (12)–(13), guarantees small similarity between two frames from different classes (speech and transient noise) even if they are very similar to each other (in the Euclidean sense), because of the large distance between neighboring frames. Upon defining the parametric weight function, the parameters can be obtained by solving the following optimization problem [19]:

$$\boldsymbol{\theta}^{opt} = \arg\min_{\theta} \frac{1}{L}\sum_{\ell=1}^{L} F(\boldsymbol{W}_{\boldsymbol{\theta}}^{\ell}, \boldsymbol{C}^{\ell}) \tag{14}$$

$$F(\boldsymbol{W},\boldsymbol{C}) = \frac{1}{2}\left\| \boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^{T} - \boldsymbol{D}^{1/2}\boldsymbol{C}(\boldsymbol{C}^{T}\boldsymbol{D}\boldsymbol{C})^{-1}\boldsymbol{C}^{T}\boldsymbol{D}^{1/2}\right\|_{F}^{2} \tag{15}$$

where L is the number of training sequence, $(\cdot)^T$ denotes transpose of a vector or a matrix, and $\boldsymbol{\Upsilon}$ is an approximate orthonormal basis of the projections on the second principal subspace of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ obtained by classical *orthogonal iteration* [32]. In practice, we use the gradient method (e.g., *fminunc* or *fmincon* functions in Matlab®) to solve this minimization problem.

2) Testing Algorithm: A testing algorithm aims to cluster the unlabeled data. The most straightforward way to perform clustering using spectral methods into K disjoint clusters is to use the parameters obtained by the learning algorithm, construct the similarity matrix $\boldsymbol{W}$, compute the K eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ corresponding to the first K largest eigenvalues (denoted by $\boldsymbol{U}$), and run weighted k-means algorithm on $\boldsymbol{U}$ or k-means algorithm on $\boldsymbol{V} = \boldsymbol{D}^{1/2}\boldsymbol{U}\left(\boldsymbol{U}^{T}\boldsymbol{D}\boldsymbol{U}\right)^{-1}$ [19]. This method has two major drawbacks. First, this method can only be used for batch processing (offline processing) of data. The second and more important one is that, this method does not allow the user to control the tradeoff between the probability of false alarm and the probability of detection. Every detection algorithm must be equipped with a tool such that one can increase the probability of detection (probably) by increasing the probability of false alarm. In order to overcome these two shortcomings, we utilize the extension method proposed in [33] based on the fact that two test points are similar if they see the training data similarly, and the likelihood ratio test as our decision rule. In order to compute the likelihood ratio, we use GMM to model the eigenvectors of normalized Laplacian matrix. In what follows, we discuss these two issues in more detail.

Let $\boldsymbol{W}_{\boldsymbol{\theta}^{opt}}^{\ell}$ be the similarity matrix of ℓ-th training sequence and $\boldsymbol{U}^{\ell} \in \mathbb{R}^{3N^{\ell}\times 2}$ be a matrix consisting of the two eigenvectors of $\boldsymbol{D}^{\ell\,-1/2}\boldsymbol{W}^{\ell}\boldsymbol{D}^{\ell\,-1/2}$ corresponding to the first two largest eigenvalues. Let the column concatenation of $\boldsymbol{U}^1$ through $\boldsymbol{U}^L$ be

$$\boldsymbol{U} = \left[(\boldsymbol{E}^{1}\odot\boldsymbol{U}^{1})^{T},\ldots,(\boldsymbol{E}^{\ell}\odot\boldsymbol{U}^{\ell})^{T},\ldots,(\boldsymbol{E}_{L}\odot\boldsymbol{U}^{L})^{T}\right]^{T} \tag{16}$$

$$\boldsymbol{E}^{\ell} = \sqrt{\boldsymbol{C}^{\ell}\mathrm{diag}\left(\mathbf{1}_{1\times N^{\ell}}\boldsymbol{C}^{\ell}\right)\mathbf{1}_{4\times 2}};\quad \ell = 1,2,\ldots,L \tag{17}$$

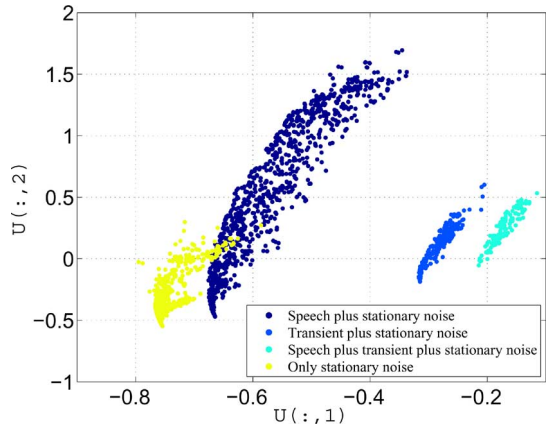


Fig. 2. Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR = 5 dB) obtained by concatenating all training sequences.

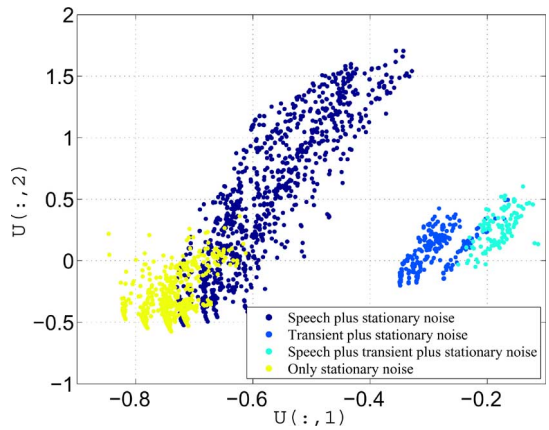


Fig. 3. Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR = 5 dB) obtained by (16)–(17).

where $\odot$ is term by term multiplication, $\mathrm{diag}(\boldsymbol{a})$, is a diagonal matrix whose diagonal is vector $\boldsymbol{a}$ and $\mathbf{1}_{m\times n}$ is an m by n matrix of ones. This normalization of the matrices $\boldsymbol{U}^1$ through $\boldsymbol{U}^L$ is due to a possible different number of points in the same cluster of different training sequences. Because of sign ambiguity in computation of eigenvectors, each of these eigenvectors is computed such that the mean of each cluster (noise only cluster or speech cluster) is as close as possible to the mean of each cluster of the first training sequence. More specifically, we compute the mean of low dimensional representation of each of the two clusters in the first training sequence and choose the sign of the eigenvectors corresponding to the remaining training sequence, such that their means are close to the means of the clusters in the first training sequence. We have selected this approach instead of combining all training sequences as a single training sequence because of computational load and memory usage. Combining all training sequence as a single sequence leads to a very large similarity matrix that cannot be handled computationally. This method is in some sense equivalent to ignoring the similarity between each training sequence, which is a widespread approach for sparsifying the similarity matrix [19]. Figs. 2 and 3 show the scatter plot of the new representation of a typical learning sequence. These figures are obtained from five training sequences each approximately 10

seconds long. The similarity matrix is calculated using optimum parameters obtained by solving the optimization problem in (14). These figures are obtained in the case where the stationary noise was colored Gaussian noise, and the transient noise was a door knock. Fig. 2 is the low dimensional representation of the training data obtained by concatenating the training sequences to a large sequence. Fig. 3 is the low dimensional representation of the training data obtained by the aforementioned method (i.e., (16)–(17)). These figures are experimental justification for the proposed approximation in computation of eigenvectors. It is also apparent that there exist four separated clusters (i.e., only stationary noise, transient plus stationary noise, speech plus transient plus stationary noise, and speech plus stationary noise), two of which contain speech signal.

Once the matrix $\boldsymbol{U}$, a new representation of the training data, is obtained, we use Gaussian mixture modeling to model each cluster (i.e., speech presence or absence) with a different GMM. A mixture model is a probabilistic model that assumes the underlying data belongs to a mixture distribution. In a mixture distribution, the density function is a convex combination of other probability density functions. The most common mixture distribution is the Gaussian density function, where each of the mixture components has a Gaussian distribution. This model has been utilized in many machine learning and speech processing applications such as speaker verification [34], texture retrieval [35], and handwriting recognition [36] just to name a few. Parameters of the GMM can be estimated by the Expectation-Maximization (EM) method [37], and the number of Gaussian component to be used can be selected by the Akaike information criterion (AIC) or Bayesian information criterion (BIC). The procedure to obtain the GMM for each cluster is as follows. For each cluster (i.e., speech presence or absence), we find the rows of the matrix $\boldsymbol{U}$ corresponding to that cluster by using the indicator matrix. Then, by exploiting the EM algorithm and AIC or BIC criterion, we fit a GMM to the new data representation in that cluster. Since the matrix $\boldsymbol{U}$ only depends on the training data, the GMM model for each of the two hypotheses (i.e., speech presence or absence) is obtained during the training phase.

Now suppose we are given T frames of unlabeled data, and we want to decide whether each of these frames belongs to the speech presence or speech absence clusters. For each of these frames, we first extract the feature vector using (2) and (3). Let

$$\boldsymbol{Z}(:,t) = \begin{bmatrix} \boldsymbol{Z}_m(:,t) \\ \Lambda_t^Z \end{bmatrix} \quad t = 1, 2, \ldots, T \tag{18}$$

be the feature vector extracted from unlabeled data, where $\boldsymbol{Z}_m(:,t)$ is the absolute value of the MFCC of the t-th frame, and Λ_t^Z is the likelihood ratio of t-th unlabeled frame obtained by (3). The similarity matrix between the new data and training data is computed as follows:

$$\boldsymbol{B} = \left[(\boldsymbol{B}^1_{\boldsymbol{\theta}^{opt}})^T, (\boldsymbol{B}^2_{\boldsymbol{\theta}^{opt}})^T, \ldots, (\boldsymbol{B}^L_{\boldsymbol{\theta}^{opt}})^T\right]^T \tag{19}$$

$$\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_p^{opt} \boldsymbol{Q}^\ell(i+p, j+p)\right) \tag{20}$$

$$\boldsymbol{Q}^\ell(i,j) = \left\| \boldsymbol{Y}^\ell_m(:,i)\left(1 - \exp\left(\frac{-\Lambda_i^\ell}{\epsilon^{opt}}\right)\right) - \boldsymbol{Z}_m(:,j)\left(1 - \exp\left(\frac{-\Lambda_j^Z}{\epsilon^{opt}}\right)\right)\right\|_2^2 \tag{21}$$

where $\boldsymbol{\theta}^{opt} = [\epsilon^{opt}, \alpha^{opt}_{-P}, \alpha^{opt}_{-P+1}, \ldots, \alpha^{opt}_{P-1}, \alpha^{opt}_{P}]$ is the optimum kernel parameters vector obtained in learning stage by solving the optimization problem in (14), and $\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}(i,j)(1 \leq i \leq N^\ell; 1 \leq j \leq T)$ is the (i,j)-th element of the matrix $\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}$. Once the similarity matrix between unlabeled data and training data has been computed, the new data representation in terms of eigenvectors of the Laplacian can be easily approximated by the following equation:

$$\tilde{\boldsymbol{U}} = \operatorname{diag}\left((\boldsymbol{1}\boldsymbol{B}_{k_{nn}})^{-1}\right)\boldsymbol{B}^T_{k_{nn}}\boldsymbol{U}, \tag{22}$$

where the i-th column of the matrix $\boldsymbol{B}_{k_{nn}}$ is obtained by setting to zero all elements of the i-th column of $\boldsymbol{B}$, except the k largest elements. The subscript k_{nn} stands for k-nearest neighbor. The last equation means that the low dimensional representation of a given test point is simply the weighted mean of the low representation k-nearest neighbor of that point in the training set. Using this new representation of the unlabeled data, the decision rule can be obtained by a likelihood ratio test as follows. Let $\mathcal{H}_0$ and $\mathcal{H}_1$ be speech absence and presence hypotheses, respectively. Let $f(\cdot;\mathcal{H}_0)$ and $f(\cdot;\mathcal{H}_1)$ be the probability density function of those rows of $\boldsymbol{U}$ corresponding to noise only frames and frames containing speech signal, respectively. These two probability density functions were obtained by GMM modeling in the training stage. The likelihood ratio for a new unlabeled frame is given by:

$$\Gamma_t = \frac{f(\tilde{U}(t,:);\mathcal{H}_1)}{f(\tilde{U}(t,:);\mathcal{H}_0)} \tag{23}$$

where $\tilde{U}(t,:)$ is the t-th row of the matrix $\tilde{U}$. Practical evidence shows that using the information supplied by neighboring frames can improve the performance of VAD algorithms [12]. This is because of the fact that frames containing speech signal are usually followed by a frame that also contains speech signal while the transient signals usually last for a single time frame. Using this fact, the decision rule for an unlabeled time frame is obtained by:

$$[ht]\mathrm{VA}_t = \sum_{j=-J}^{J} \Gamma_{t+j} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h \quad t = 1, 2, \ldots, T \tag{24}$$

where T_h is a threshold which controls the tradeoff between probability of detection and false alarm. Increasing (decreasing) this parameter leads to a decrease (increase) of both the probability of false alarm and the probability of detection. In a practical implementation, a hangover scheme is required to lower the probability of false rejections. The hangover scheme does this by reducing the risk of a low-energy portion of speech at the end of an utterance being falsely rejected, by arbitrarily declaring a period of speech activity after a period of speech activity has already been detected. This is based on the idea that speech

TABLE I
PROPOSED VOICE ACTIVITY DETECTION ALGORITHM BASED ON SPECTRAL CLUSTERING METHOD

Learning algorithm:

1. Construct a training data set as described previously that consists of L training signals $\{\boldsymbol{Y}^{\ell} \in \mathbb{R}^{K_m+1\times 3N^{\ell}}; \ell = 1, ..., L\}$ together with L indicator matrices $\{\boldsymbol{C}^{\ell} \in \mathbb{R}^{3N^{\ell}\times 4}; \ell = 1, ..., L\}$.
2. Solve the optimization problem given in (14), to find the optimum value of the parameters (i.e. $\boldsymbol{\theta}^{opt}$).
3. Construct the matrix $\boldsymbol{U}$ by column concatenation of U^1 through U^L by taking into account the sign ambiguity in computation of eigenvectors (see (16) and the text for further discussions).
4. Utilize the EM algorithm to fit a GMM model to the rows of $\boldsymbol{U}$ for each cluster (i.e. speech presence and speech absence).

Output:
$\boldsymbol{U}$, $f(\cdot;\boldsymbol{\mathcal{H}}_1)$ and $f(\cdot;\boldsymbol{\mathcal{H}}_0)$

Testing Procedure:
Let $z_t(n)$ be the test sequence and $\boldsymbol{Z}_t \in \mathbb{R}^{K_m+1\times N^z}$ the feature matrix extracted from $z(n)$ using (18).
for $t = 0 : T : N^z - T$ $(T \ll N^z)$

1. $\boldsymbol{Z} = \boldsymbol{Z}_t(:, t+1 : t+P)$.
2. Compute $\boldsymbol{B}_{k_{nn}}$ using (19)-(21).
3. Compute the new representation of the unlabeled data using
$$\tilde{\boldsymbol{U}} = \text{diag}\left((\mathbf{1}\boldsymbol{B}_{k_{nn}})^{-1}\right)\boldsymbol{B}_{k_{nn}}^{T}\boldsymbol{U}.$$
4. Compute the likelihood ratio for a new unlabeled frame by
$$\Gamma_t = \frac{f(\tilde{U}(t,:);\boldsymbol{\mathcal{H}}_1)}{f(\tilde{U}(t,:);\boldsymbol{\mathcal{H}}_0)}.$$
5. The decision rule for an unlabeled time frame is given by
$$\text{VA}_t = \sum_{j=-J}^{J}\Gamma_{t+j} \underset{\boldsymbol{\mathcal{H}}_0}{\overset{\boldsymbol{\mathcal{H}}_1}{\gtrless}} T_h\,.$$
6. Use VA_t and the hangover technique introduced in [9] to obtain the final VAD decision.

end

occurrences are highly correlated with time. We use the hangover technique introduced by Davis *et al.* in [9]. More specifically, the quantity VA_t is the input of the hangover procedure, and a final VAD decision is obtained from hangover scheme. Our overall voice activity detection algorithm is summarized in Table I.

III. SIMULATION RESULTS AND PERFORMANCE EVALUATION

In this section, we examine the performance of the proposed method using several simulations. We also compare the performance of our method with that of conventional statistical model-based methods presented in [6]–[8], [11], [12] and two standard VAD's: G.729 [3] and AMR2 [38]. The simulation setup is as follows.

We perform our simulation for different types of stationary and transient noise for different SNR situations. The SNR is defined as the ratio of the speech energy to the energy of stationary noise. The stationary noise energy is computed in those frames where speech signal is present. All speech and transient noise signals are sampled at 16 kHz (although the same performance was obtained at 8 kHz sampling rate) and normalized to have unity as their maximum. Since the duration of transient noise is small with respect to speech, defining SNR for transient noise is not useful. Instead, we normalize the transient noise and speech signal to have the same maximum amplitude, which is a very challenging case to treat [39]. Each signal (speech or transient noise) is approximately 3 sec long. The training and testing sequences are constructed using the procedure introduced in (6) and (7). Speech signals are taken from the TIMIT database [40]. Transient noises are taken from [41]. In the training step, we use $M = 50$ different speech utterances (different speakers, half male and half female) and transient noise. In the testing step, we use $M = 50$ different speech utterances (different speakers from the training set, half male and half female) and transient noise (different from the training sequences) each approximately 3 sec long (the length of the testing signal is approximately 500 sec, with sixty percent of total frames containing speech). We use windowed STFT with a hamming window of $K_s = 512$ samples long and 50% overlap between consecutive frames. We compute the MFCC in $K_m = 24$ Mel frequency bands. To solve the optimization problem (14) in the training stage, we use the $fmincon$ function in Matlab®. We solve this optimization problem under the constraint that all estimated parameters are strictly positive. This constraint results in an appropriate similarity matrix. The parameter k in computing the matrix $\boldsymbol{B}_{k_{nn}}$ ((22)) is set to 10.

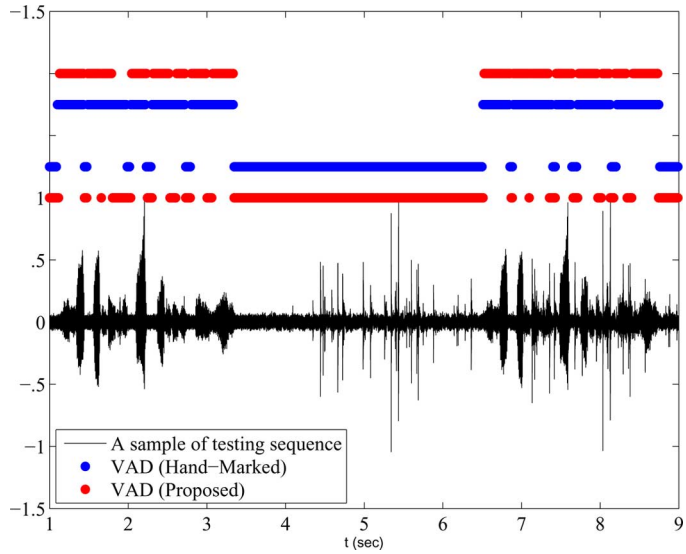


Fig. 4. A realization of the test signal (speech signal plus keyboard typing plus babble noise) together with hand-marked and proposed VAD.

In order to compare our method to the conventional statistical based method, we introduce two different kinds of false alarm probabilities. The first type denoted by Pfa, is defined as the probability that a speech free frame (i.e., consisting of only stationary noise or stationary noise with transient noise) is detected as a speech frame (i.e., exactly the same as probability of false alarm defined in conventional methods). The second type, denoted by Pfa_{tr}, is defined as the probability that a frame consisting of stationary and transient noise is detected as a speech frame. We need these two concepts to show the advantage of the proposed method over conventional statistical model-based methods. The number of frames that contain transient noise (which are mostly detected as speech in statistical model-based methods) is small with respect to the total number of frames. Such frames do not affect the probability of false alarm significantly if it is defined as the probability that a noise frame is detected as a speech frame. For our comparison to be fair, we

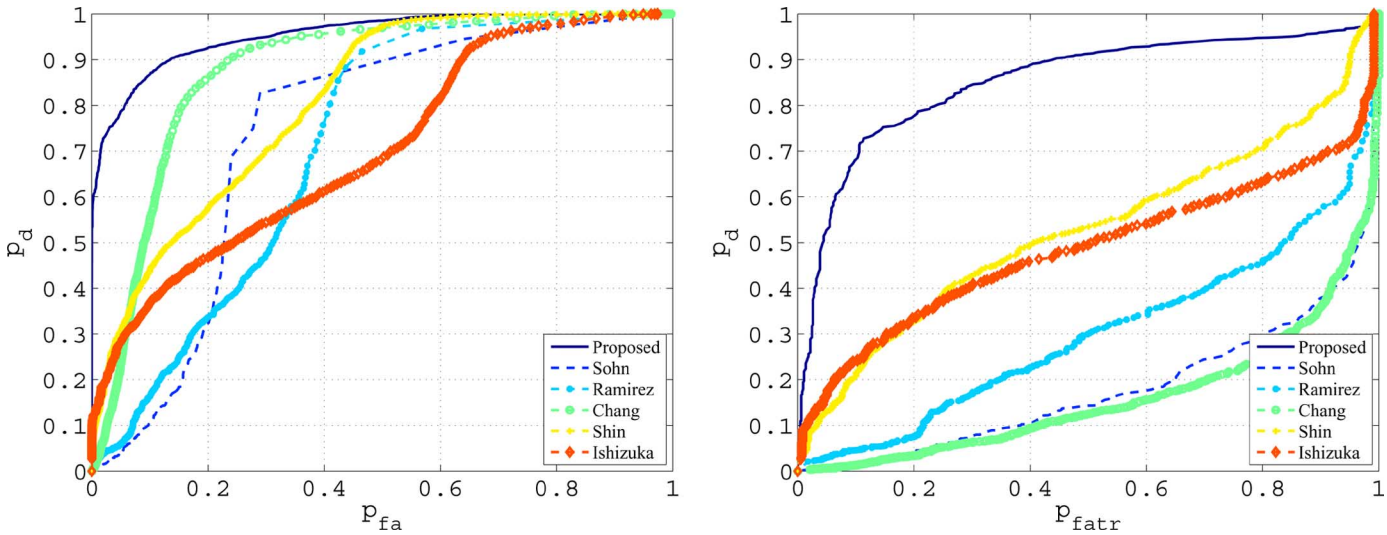


Fig. 5. Probability of detection versus probability of false alarm (Training: SNR $=$ 10 dB; Stationary Noise: Babble; Transient Noise: Metronome; Testing: SNR $=$ 10 dB; Stationary Noise: Babble; Transient Noise: Metronome).

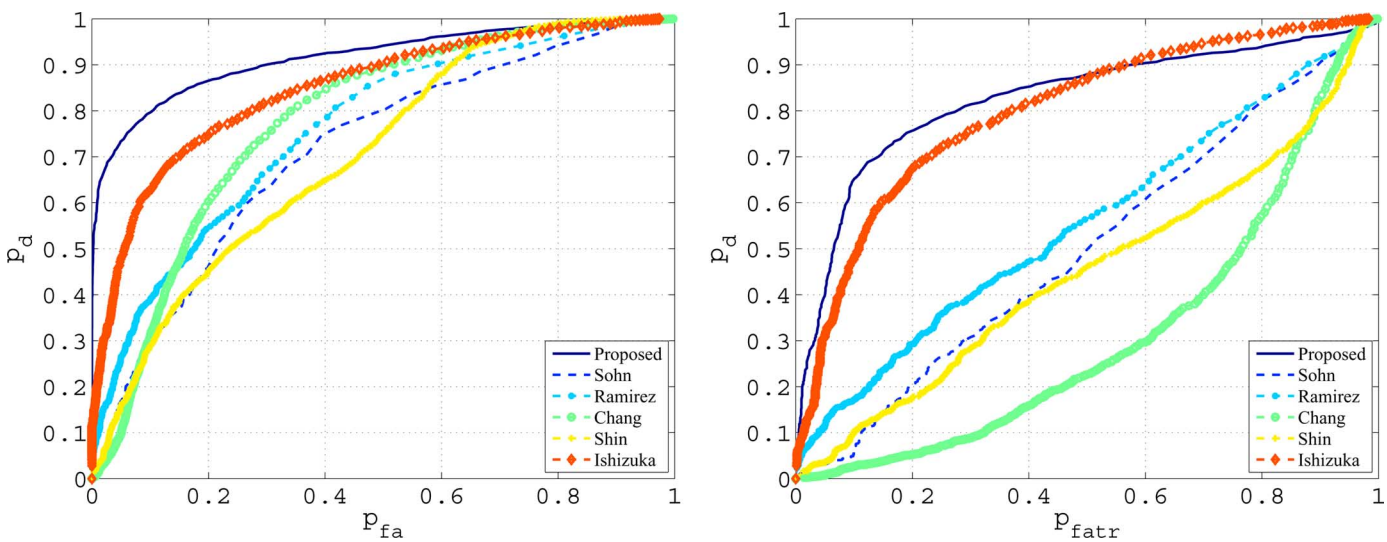


Fig. 6. Probability of detection versus probability of false alarm (Training: SNR $=$ 5 dB, Stationary Noise: Babble; Transient Noise: Keyboard Typing; Testing: SNR $=$ 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing).

assume that the stationary noise statistics are known in conventional methods. The noise statistics are estimated using a realization of stationary noise. In what follows, we investigate the performance of the proposed method in several situations. For our comparison to be more insightful, we also use the following six parameters defined in [9] to indicate the VAD performance:

- **FEC** (front end clipping): Clipping due to speech being misclassified as noise in passing from noise to speech activity.
- **MSC** (mid speech clipping): Clipping due to speech misclassified as noise during an utterance.
- **BEC** (back end clipping): Clipping due to speech being misclassified as noise in passing from speech activity to noise.
- **Over** (over hang): Noise interpreted as speech due to the VAD flag remaining active in passing from speech activity to noise.
- **NDS** (noise detected as speech): Noise interpreted as speech within a silent period.
- **Correct** (correct VAD decision): Correct decisions made by the VAD.

The results of simulations are depicted in Figs. 4–8 and Tables III–VII. Fig. 4 shows a speech signal corrupted by keyboard typing and babble noise with SNR $=$ 20 dB together with hand-marked VAD and the result of the proposed algorithm. For this simulation, the detection threshold is chosen such that $\mathrm{Pfa}_{\mathrm{tr}}$ be zero. The results of Tables III–VII are obtained by setting the threshold such that the probability of detection (**Correct**) be at least ninety-five percent. In Tables III–VII, the best performance (i.e., the lowest probability of false alarm **NDS**) is identified in bold number. As can be seen from Figs. 5–8 and Tables III–VII, although different statistical model-based methods have different performances in different situations, the proposed method is superior in all simulations

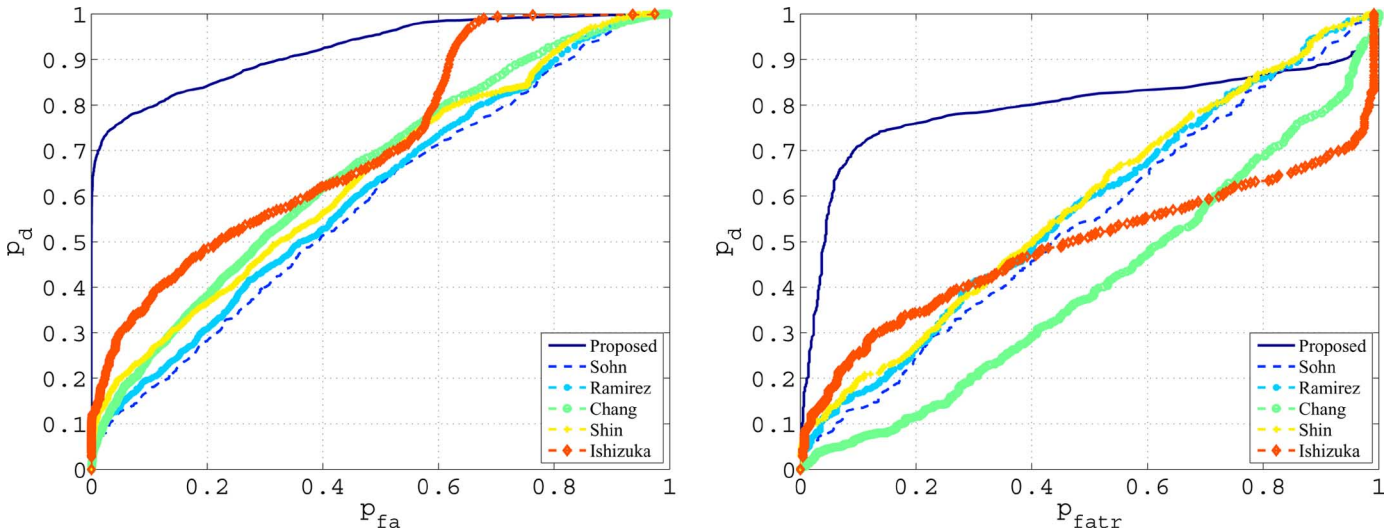


Fig. 7. Probability of detection versus probability of false alarm (Training: SNR $= 5$ dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; Testing: SNR $= 5$ dB; Stationary Noise: White Gaussian; Transient Noise: Metronome).

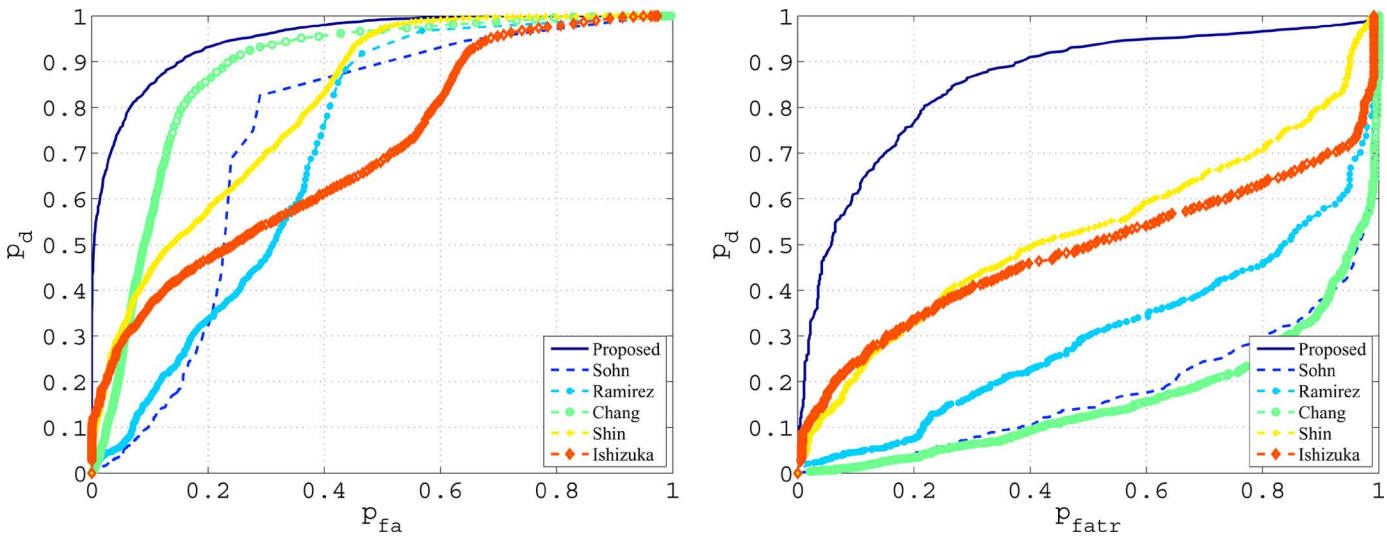


Fig. 8. Probability of detection versus probability of false alarm (Training: SNR $= 10$ dB; Stationary Noise: Colored Gaussian; Transient Noise: Door Knock; Testing: SNR $= 10$ dB; Stationary Noise: Babble; Transient Noise: Metronome).

TABLE II
ELAPSED TIME IN SECONDS FOR DIFFERENT METHODS

Proposed	Sohn	Ramirez	Chang
50.41	2.19	2.28	2.30
Shin	**Ishizuka**	**G729**	**AMR2**
2.42	4.08	1.51	1.31

TABLE III
VOICE ACTIVITY DETECTION PERFORMANCE COMPARISON (**Training**:SNR $= 10$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: METRONOME; **Testing**:SNR $= 10$ dB; STATIONARY NOISE: BABBLE NOISE; TRANSIENT NOISE: METRONOME)

	Correct	FEC	MSC	BEC	NDS	Over
Proposed	95.04	0.50	3.37	1.09	**29.25**	1.15
Sohn	99.23	0.37	0.17	0.23	78.06	0.31
Ramirez	98.20	0.26	0.99	0.55	63.95	0.95
Chang	96.09	0.59	1.53	1.79	33.02	0.57
Shin	95.95	0.54	2.47	1.04	68.77	1.44
Ishizuka	95.95	0.54	2.47	1.04	68.77	1.44
G.729	99.94	0.00	0.06	0.00	96.09	0.16
AMR2	99.59	0.24	0.17	0.00	95.4	0.26

over the compared statistical model-based methods, especially for low false alarm rates. The proposed method outperforms the statistical model-based methods even in the case that the training and testing do not match (Figs. 7 and 8 and Tables VI and VII).

Simulation results indicate that the matrix $\boldsymbol{U}$ is sensitive to change in SNR and stationary or transient noise type. Hence, for different SNR or stationary or transient noise type, the matrix $\boldsymbol{U}$ must be recomputed (even though simulation results reveal the effectiveness of the proposed method even if there exists a mismatch between training and testing, see Figs. 7 and 8 and Tables V–VII). In all of the following simulations we set the parameter vector to $\boldsymbol{\theta} = .001 \times [300\ 0.4\ .75\ 1\ 0.75\ 0.4]$. In the testing algorithm we chose $T = 10$, which means that our algorithm has 160 msec delay (each frame is $K_s/(2f_s) = 16$ msec long, where the factor of 2 in the denominator is due to fifty percent overlap). Choosing this parameter is a tradeoff between induced delay and computational load. Increasing T leads to a

TABLE IV
VOICE ACTIVITY DETECTION PERFORMANCE COMPARISON ($\mathbf{Training}$:SNR $= 5$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: KEYBOARD TYPING; $\mathbf{Testing}$:SNR $= 5$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: KEYBOARD TYPING)

	Correct	FEC	MSC	BEC	NDS	Over
Proposed	95.16	0.61	3.46	0.77	**53.44**	1.46
Sohn	98.79	0.36	0.52	0.32	82.00	0.25
Ramirez	97.90	0.42	1.20	0.47	68.55	0.89
Chang	95.71	0.81	2.22	1.26	63.18	0.76
Shin	95.51	0.64	2.52	1.33	66.11	0.88
Ishizuka	95.65	0.83	2.51	1.01	59.23	1.43
G.729	99.95	0.05	0.00	0.00	96.85	0.22
AMR2	99.43	0.40	0.16	0.97	96.03	0.85

TABLE V
VOICE ACTIVITY DETECTION PERFORMANCE COMPARISON ($\mathbf{Training}$:SNR $= 5$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: KEYBOARD TYPING; $\mathbf{Testing}$:SNR $= 5$ dB; STATIONARY NOISE: WHITE GAUSSIAN; TRANSIENT NOISE: METRONOME)

	Correct	FEC	MSC	BEC	NDS	Over
Proposed	95.05	0.95	3.04	0.95	**45.39**	2.16
Sohn	96.35	0.90	1.59	1.17	87.12	0.66
Ramirez	95.94	0.60	2.69	0.77	83.53	1.11
Chang	95.97	0.88	2.06	1.09	77.79	0.81
Shin	95.79	0.85	1.67	1.69	70.56	0.83
Ishizuka	96.11	0.65	1.88	1.36	62.51	1.67
G.729	99.91	0.04	0.02	0.02	95.33	0.16
AMR2	98.96	0.07	0.96	0.88	94.82	0.33

TABLE VI
VOICE ACTIVITY DETECTION PERFORMANCE COMPARISON ($\mathbf{Training}$:SNR $= 10$ dB; STATIONARY NOISE: COLORED GAUSSIAN; TRANSIENT NOISE: DOOR KNOCK; $\mathbf{Testing}$:SNR $= 10$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: METRONOME)

	Correct	FEC	MSC	BEC	NDS	Over
Proposed	95.03	0.55	3.01	1.42	**24.04**	1.40
Sohn	99.23	0.37	0.17	0.23	78.06	0.31
Ramirez	98.20	0.26	0.99	0.55	63.95	0.95
Chang	96.09	0.59	1.53	1.79	33.02	0.57
Shin	95.85	0.61	1.70	1.84	45.04	0.88
Ishizuka	95.95	0.54	2.47	1.04	68.77	1.44
G.729	99.94	0.00	0.06	0.00	96.09	0.16
AMR2	99.05	0.94	0.01	0.06	95.95	0.62

TABLE VII
VOICE ACTIVITY DETECTION PERFORMANCE COMPARISON ($\mathbf{Training}$:SNR $= 20$ dB; STATIONARY NOISE: BABBLE; TRANSIENT NOISE: DOOR KNOCK; $\mathbf{Testing}$:SNR $= 5$ dB; STATIONARY NOISE: COLORED GAUSSIAN; TRANSIENT NOISE: KEYBOARD TYPING)

	Correct	FEC	MSC	BEC	NDS	Over
Proposed	95.03	0.88	2.83	1.26	**31.47**	1.29
Sohn	98.01	0.37	0.79	0.83	76.01	0.86
Ramirez	97.23	0.45	1.75	0.57	74.28	0.98
Chang	96.29	0.59	1.52	1.60	60.84	0.76
Shin	95.68	0.43	2.68	1.20	66.51	0.82
Ishizuka	95.82	0.57	2.34	1.27	22.13	1.57
G.729	99.98	0.00	0.02	0.00	96.57	0.16
AMR2	99.51	0.49	0.10	0.94	95.93	0.21

lower computational load but increases the delay. The decrease in computational load by increasing the parameter T is due to the fact that there exist efficient algorithms for computing the matrix B ((19)), and it can be shown that the computational load is of the order $O(T + 1/T)$. The most time consuming part of the proposed algorithm is solving the optimization problem, which is done offline and is not of great importance in practice. The excess computational load in the testing stage compared to statistical model-based methods is computation of the matrix $\boldsymbol{B}_{k_{nn}}$. The elapsed time for processing a 500 second sequence sampled at 16 kHz for the proposed method and other competing methods are depicted in Table II. Although the computational load of the proposed algorithm is relatively higher than other methods, there exist efficient algorithms for decreasing the computational load [42].

IV. CONCLUSIONS

We have proposed a novel voice activity detector based on spectral clustering method. Our main concern was dealing with transient noise, which is very challenging to handle. Almost all conventional methods fail in this situation. Our VAD is a supervised learning algorithm that requires some training data in order to estimate the parameters of the kernel used for computation of a similarity matrix. We used GMM to model the eigenvectors of the similarity matrix. In the testing stage, we used eigenvector extension and proposed a VAD which can be used for online processing of the data with a small delay. Simulation results have demonstrated the high performance of the proposed method, particularly its advantage in treating transient noises.

ACKNOWLEDGMENT

The authors thank the associate editor and the anonymous reviewers for their constructive comments and useful suggestions. The first author would also like to thank Rahmat and Marisa for their help in preparing this paper.

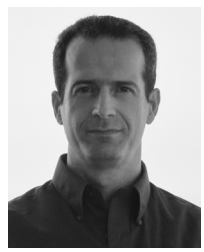
REFERENCES

[1] L. Karray and A. Martin, "Toward improving speech detection robustness for speech recognition in adverse environments," *Speech Commun.*, vol. 3, pp. 261–276, 2003.
[2] J. Ramrez, J. Segura, C. Bentez, A. de la Torre, and A. Rubio, "A new adaptive longterm spectral estimation voice activity detector," in *Proc. EUROSPEECH '03*, Geneva, Switzerland, 2003, pp. 3041–3044.
[3] A. Benyassine, E. Shlomot, H. Su, D. M. C. Lamblin, and J. Petit, "ITU-T recommendation G.729x annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 64–73, Sep. 1997.
[4] A. Sangwan, M. Chiranth, H. Jamadagni, R. Sah, and R. P. V. Gaurav, "VAD techniques for real-time speech transmission on the Internet," in *Proc. IEEE Int. Conf. High-Speed Netw. Multimedia Commun.*, 2002, pp. 46–50.
[5] F. Basbug, K. Swaminathan, and S. Nandkumar, "Noise reduction and echo cancellation front-end for speech codecs," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 1, pp. 1–13, Jan. 2004.
[6] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 16, no. 1, pp. 1–3, Jan. 1999.
[7] J. H. Chang and N. S. Kim, "Voice activity detection based on complex Laplacian model," *Electron. Lett.*, vol. 39, no. 7, pp. 632–634, 2003.
[8] J. W. Shin, J. H. Chang, H. S. Yun, and N. S. Kim, "Voice activity detection based on generalized gamma distribution," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 1, pp. I781–I784.
[9] A. Davis, S. Nordholm, and R. Togneri, "Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 2, pp. 412–424, Mar. 2006.
[10] S. Mousazadeh and I. Cohen, "AR-GARCH in presence of noise: Parameter estimation and its application to voice activity detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 916–926, May 2011.
[11] K. Ishizuka, T. Nakatani, M. Fujimoto, and N. Miyazaki, "Noise robust voice activity detection based on periodic to aperiodic component ratio," *Speech Commun.*, vol. 52, no. 1, pp. 41–60, Jan. 2010.

[12] J. Ramirez and J. C. Segura, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 689–692, Oct. 2005.
[13] I. Volfin and I. Cohen, "Dominant speaker identification for multipoint videoconferencing," *Comput. Speech Lang.*, 2012.
[14] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, *Multimodal Technologies for Perception of Humans*, R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Berlin, Heidelberg, Germany: Springer-Verlag, 2008, ch. HMM-Based Acoustic Event Detection with AdaBoost Feature Selection, pp. 345–353.
[15] P. S. Huang, X. Zhuang, and M. Hasegawa-Johnson, "Improving acoustic event detection using generalizable visual features and multi-modality modeling," in *Proc. ICASSP*, 2011, pp. 349–352.
[16] M. Espi, M. Fujimoto, D. Saito, N. Ono, and S. Sagayama, "A tandem connectionist model using combination of multi-scale spectro-temporal features for acoustic event detection," in *Proc. ICASSP '12*, Mar. 2012, pp. 4293–4296.
[17] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Adv. Neural Inf. Process. Syst.*, vol. 17, pp. 1601–1608, 2005.
[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Patern. Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[19] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.*, vol. 7, pp. 1963–2001, 2006.
[20] W. Pentney and M. Meila, "Spectral clustering of biological sequence data," in *AAAI*, M. M. Veloso and S. Kambhampati, Eds. Cambridge, MA, USA: AAAI Press/The MIT Press, 2005, pp. 845–850.
[21] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, 2007.
[22] A. Jansen and P. Niyogi, "A geometric perspective on speech sounds," Computer Science Dept., Univ. of Chicago, Chicago, IL, USA, 2005, Tech. Rep..
[23] B. Scholkopf, A. Smola, and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, 1996.
[24] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, pp. 1373–1396, 2003.
[25] D. L. Donoho and C. Grimes, "Image manifolds which are isometric to Euclidean space," *J. Math. Imaging Vis.*, vol. 23, no. 1, pp. 5–24, Jul. 2005.
[26] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 5–30, 2006.
[27] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.
[28] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.
[29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing System*. Cambridge, MA, USA: MIT Press, 2001, pp. 849–856.
[30] M. Hein and J. Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in R^d," in *Proc. ICML*, 2005, pp. 289–296.
[31] R. R. Coifman, Y. Shkolnisky, F. J. Sigworth, and A. Singer, "Graph Laplacian tomography from unknown random projections," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1891–1899, Oct. 2008.
[32] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
[33] R. Talmon, D. Kushnir, R. R. Coifman, I. Cohen, and S. Gannot, "Parametrization of linear systems using diffusion kernels," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1159–1173, Mar. 2012.
[34] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, pp. 72–83, 1995.
[35] R. P. Francos, H. Permuter, and J. Francos, "Gaussian mixture models of texture and colour for image database," *Proc. ICASSP*, pp. 25–88, 2003.
[36] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[37] J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models Int. Computer Science Inst., Berkeley, CA, USA, 1998, Tech. Rep.
[38] E. Cornu, H. Sheikhzadeh, R. Brennan, H. Abutalebi, E. Tam, P. Iles, and K. Wong, "ETSI AMR-2 VAD: Evaluation and ultra low-resource implementation," in *Proc. Int. Conf. Multi. Expo, (ICME'03)*, 2003, vol. 2, pp. 841–844.
[39] R. Talmon, I. Cohen, and S. Gannot, "Transient noise reduction using nonlocal diffusion filters," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1584–1599, Aug. 2011.
[40] J. S. Garofolo, *Getting Started With the DARPA TIMIT CD-ROM: An Acoustic-Phonetic Continous Speech Database*. Gaithersburg, MD, USA: National Inst. of Standards and Technology (NIST), 1993.
[41] [Online]. Available: http://www.freesound.org
[42] C. Yu, B. Ooi, K. Tan, and H. Jagadish, "Indexing the distance: An efficient method to knn processing," in *Proc. Int. Conf. Very Large Data Bases*, 2001, pp. 421–430.

Saman Mousazadeh was born in Shiraz, Iran, in 1982. He received the B.S. and M.S. degrees, both in electrical engineering, from Shiraz University, Shiraz, Iran, in 2005 and 2008, respectively.

He is currently pursuing the Ph.D. degree (direct track) in electrical engineering at the Technion—Israel Institute of Technology, Haifa, Israel. Since 2009, he has been a Project Supervisor with the Signal and Image Processing Lab (SIPL), Electrical Engineering Department, Technion. His research interests include speech, image and array signal processing. He received the student challenge award of the Acoustical Society of America (ASA) in 2006.



Israel Cohen (M'01–SM'03) is a Professor of electrical engineering at the Technion—Israel Institute of Technology, Haifa, Israel. He received the B.Sc. (Summa Cum Laude), M.Sc. and Ph.D. degrees in electrical engineering from the Technion—Israel Institute of Technology, in 1990, 1993 and 1998, respectively.

From 1990 to 1998, he was a Research Scientist with RAFAEL Research Laboratories, Haifa, Israel Ministry of Defense. From 1998 to 2001, he was a Postdoctoral Research Associate with the Computer Science Department, Yale University, New Haven, CT. In 2001 he joined the Electrical Engineering Department of the Technion. His research interests are statistical signal processing, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification and adaptive filtering. He is a coeditor of the Multichannel Speech Processing section of the *Springer Handbook of Speech Processing* (Springer, 2008), a coauthor of *Noise Reduction in Speech Processing* (Springer, 2009), a coeditor of *Speech Processing in Modern Communication: Challenges and Perspectives* (Springer, 2010), and a general co-chair of the 2010 International Workshop on Acoustic Echo and Noise Control (IWAENC).

Dr. Cohen is a recipient of the Alexander Goldberg Prize for Excellence in Research, and the Muriel and David Jacknow award for Excellence in Teaching. He serves as a member of the IEEE Audio and Acoustic Signal Processing Technical Committee (AASP TC) and the IEEE Speech and Language Processing Technical Committee (SLTC). He served as Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS, and as Guest Editor of a special issue of the *EURASIP Journal on Advances in Signal Processing* on Advances in Multimicrophone Speech Processing and a special issue of the *Elsevier Speech Communication Journal* on Speech Enhancement.