Overview of Front-end Features for Robust Speaker Recognition

Qin Jin[†] and Thomas Fang Zheng[*]
[†]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA USA
E-mail: qjin@cs.cmu.edu Tel: +00-1-4122685479
[*]Center for Speech and Language, Tsinghua University, Beijing
E-mail: fzheng@tsinghua.edu.cn Tel: +86-10-62796393

***Abstract*— This paper provides an overview of automatic speaker recognition technologies, with an emphasis on front-end features for robust speaker recognition. We categorize the front-end features into low-level features and high-level features. We discussed in detail several selected low-level and high-level features including their motivations, mechanism, reported improvements, and limitations, etc. The goal of this paper is to help beginners in speaker recognition research area to catch the overall picture of current front-end features quickly.**

I. Introduction

Speech, as the most natural way for human communication, conveys several types of information. From the speech production point of view, the speech signal conveys linguistic information (e.g., message and language information) and speaker information (e.g., emotional and physiological characteristics) etc. From the speech perception point of view, it conveys information about the environment in which the speech was produced and transmitted. Even though this wide range of information is encoded in a complex form, humans can effortlessly decode most of this information. This human ability has inspired researchers to understand speech production and perception for developing systems that automatically extract and process the richness of information in speech. Automatic speaker recognition is the process of recognizing a person's identity from his or her voice [1, 2]. Speaker recognition technology has wide application areas. It enables systems to use a person's voice to control the access to restricted services (automatic banking services), information (telephone access to financial transactions), or area (government or research facilities). It also allows detection of speakers, for example, voice-based information retrieval within audio archives, recognition of perpetrator in forensic analysis, and personalization of user devices. Speaker diarization [3] attempts to find speakers turn takings in a conversation. It is an extension of the "classical" speaker recognition technologies applied in multiparty conversations.

There have been several very useful survey or tutorial papers for speaker recognition in the past, including Furui's overview [1], Campbell's tutorial [2], Bimbot et al's tutorial [4], and the most recent overview by Kinnunen and Li [6]. This paper presents an overview of speaker recognition technologies with an emphasis on front-end features for speaker recognition, including a few representative features from low-level acoustic features to high-level super-segmental features. The remaining of this paper is organized as follows. Section II provides a general overall overview of speaker recognition technologies. Section III and IV presents the low-level and high-level features for speaker recognition. Section V presents the conclusions.

II. Overview of Speaker Recognition Technologies

Speaker recognition can be categorized into three fundamental tasks [1-3]: speaker identification, speaker verification/detection, and speaker diarization. Speaker identification task determines who is speaking given a set of known voices. In this task, the system uses only the voice (no identity claim is required) to perform recognition of the unknown speaker. There are two modes of operation for speaker identification: in the closed-set mode, the system assumes that the unknown voice must come from the set of known voices; in open-set mode, the speakers that do not belong to the set of known voices are referred to as impostors. An important application of speaker identification technology is forensics, identifying the suspects among a set of known criminals. Speaker verification task is also known as voice verification or authentication, speaker authentication, talker verification or authentication, and speaker detection. Differently from the speaker identification task, the system also requires an identity claim together with the voice sample. This is an open-set task because it also involves rejecting impostors. This task can be used for security applications, such as, to control telephone access to banking services. Speaker diarization task is also known as "who spoken when" or speaker segmentation and clustering. Very differently from both speaker identification and speaker verification, this task applies to multi-party speaker conversation scenarios. The system identifies the speaker turn changes and clusters the segments that belong to the same speaker. This task can be used for spoken document indexing and retrieval, meta data generation etc.

Automatic speaker recognition systems can be further classified according to the speech modality: text-dependent or text-independent. In text-dependent recognition, the user must speak a phrase known to the system, which can be fixed or prompted. The knowledge of a spoken phrase can provide better recognition results. In text-independent recognition, the system does not know the phrase spoken by the user. Despite the unconstrained phrase selection, this makes the system to be more complex. However, text-independent speaker recognition systems have more applications than text-dependent ones in real life.

There are generally two phases [2] in building or using a speaker recognition system. The first phase is called enrollment or training phase, in which a user enrolls by providing voice samples to the system. The system extracts speaker-specific information from the voice samples to build a voice model of the enrolled speaker. The second phase is called the classification or recognition phase, in which a test voice sample is used by the system to measure the similarity of the user's voice to the previously enrolled speaker models to make a decision. In a speaker identification task, the system measures the similarity of the test sample to all stored voice models. In speaker verification task, the similarity is measured only to the model of the claimed identity. The decision also differs across systems. For example, a closed-set identification task outputs the identity of the recognized user; besides the identity, an open-set identification task can also choose to reject the user in case the test sample do not belong to any of the stored voice models; a verification task chooses to accept or reject the identity claim.

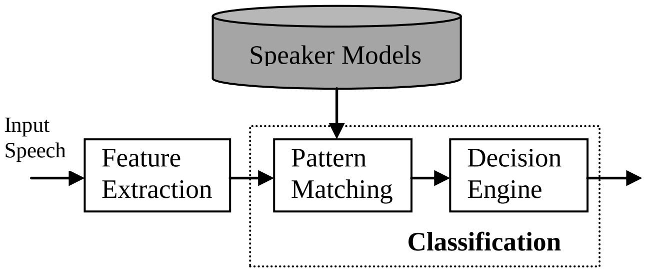

Fig. 1: A General Speaker Recognition System.

Like most pattern recognition problems, a speaker recognition system can be partitioned into two modules: feature extraction and classification. The classification module has two components: pattern matching and decision engine. Fig. 1 shows a general speaker recognition system architecture. The feature extraction module estimates a set of features from the speech signal that represent some speaker-specific information. An ideal feature would have the following characteristics [7, 8]:

- occur naturally and frequently in normal speech,
- be easily measurable,
- have large between speaker variability and small within-speaker variability,
- not change over time or be affected by the speaker's health,
- not be affected by reasonable background noise nor depend on specific transmission characteristics,
- be robust to disguise or mimicry.

In practice, not all of these criteria can be applied to the parameters used by the current systems.

The pattern matching module is responsible for comparing the estimated features to models from the set of known speakers. The speaker models are trained and stored into the system database based on feature vectors extracted from the feature extraction module. There are many types of pattern matching methods and corresponding models used in speaker recognition. In text-dependent speaker recognition, the model is utterance-specific and it contains the temporal dependencies between the feature vectors. Therefore, text-dependent speaker recognition and speech recognition shares similarities in their pattern matching processes and these can also be combined [9, 10]. In text-independent recognition, the training and test utterances are unrestricted; we often model the feature distribution, i.e. the shape of the "feature cluster" rather than the temporal dependencies. Researchers have proposed methods [11-13] to segment the speech signal into phones or broad phonetic classes as a pre-processing step and then do the modeling similarly as in the text-dependent mode. People also use data-driven units instead of the strictly linguistic phonemes as segmentation units [80]. Classical speaker models can also be categorized into nonparametric and parametric models. They are also called template models and stochastic models, respectively. Vector quantization (VQ) [14] and dynamic time warping (DTW) [15] are representative examples of template models for text-independent and text-dependent recognition, respectively. In stochastic models, each speaker is modeled as a probabilistic source with an unknown but fixed probability density function. The training phase is to estimate the parameters of the probability density function from the training data. The likelihood of the test utterance with respect to the model is used for pattern matching. The Gaussian mixture model (GMM) [16, 17] and the hidden Markov model (HMM) [18, 19] are the most popular stochastic models for text-independent and text-dependent speaker recognition, respectively.

Speaker models can also be classified into generative and discriminative models. The generative models such as GMM and VQ estimate the feature distribution within each speaker independently. While the discriminative models such as artificial neural networks (ANNs) [20, 21] and support vector machines (SVMs) [22, 23] model the boundary between speakers.

In open-set applications (speaker verification and open-set speaker identification), the estimated features can also be compared to a model that represents the unknown speakers. In a verification task, the pattern matching module outputs a similarity score between the test sample and the claimed identity. In an identification task, it outputs similarity scores for all stored speaker models.

The decision engine module analyzes the similarity score(s) (statistical or deterministic) to make a decision. The decision process is related to the task. For closed-set identification task, the decision engine can just select the identity associated with the model that is the most similar to the test sample. In open-set applications, the systems can also require a threshold to verify whether the similarity is valid. Since open-set application can also reject speakers, the cost of making an error should also be considered in the decision process. For example, it is more costly for a bank to allow an impostor (false acceptance) to withdraw money, than to reject a true bank customer.

The effectiveness of a speaker recognition system is measured differently for different tasks. Since the output of a closed-set speaker identification system is a speaker identity from a set of known speakers, the identification accuracy is used to measure the performance. For the open-set systems, there are two types of error: false acceptance of an impostor and false rejection of a known speaker. The performance measure can also incorporate the cost associated with each error. The performance of a speaker diarization system is measured by diarization error rate (DER) [3]. It is expressed in terms of the miss (speaker in reference but not in hypothesis), false alarm (speaker in hypothesis but not in reference), and speaker-error (mapped reference speaker is not the same as the hypothesized speaker) rates. The overall DER is the sum of these three components.

In summary, a general speaker recognition system consists of two key modules: feature extraction and classification. The classification module needs the extracted features from the feature extraction module. The feature extraction module is crucial in any speaker recognition systems. In the following sections we will overview different front-end features for robust speaker recognition.

III. Overview of low-level features

Speech signal includes many features that are useful for speaker discrimination. Humans rely on such different types or levels of information in the speech signal to recognize others. We can roughly categorize these features into a hierarchy running from low-level features to high-level features. Low-level features are generally related to physical traits of a speaker's vocal apparatus. Short-term spectral features, as the name suggests, are computed from short frames of about 20-30 milliseconds in duration. They are usually descriptors of the short-term spectral envelope which is an acoustic correlate of timbre, i.e. the "color" of sound, as well as the resonance properties of the supra-laryngeal vocal tract. High-level features are generally related to a speaker's learned habits and style, such as particular word usage or idiolect. While all of these levels appear to convey useful speaker information, automatic speaker recognition systems have relied almost exclusively on low-level information via short-term features related to the speech spectrum until the SuperSID project [24] at the 2002 JHU Summer Workshop. We have discussed the characteristics that an ideal feature should have in previous section. From the practical point of view, the number of dimensions of features should be also relatively low. Traditional statistical models such as the Gaussian mixture model [16, 17] cannot handle high-dimensional data. The number of required training samples for reliable density estimation grows exponentially with the number of features. Normally low-level features also have low dimensionality. The low-level features are still the most popular speaker features in current state-of-the-art speaker recognition systems. In this section, we overview some selected low-level features.

A. MFCC

Mel Frequency Cepstral Coefficients (MFCCs) have been the most popular low-level features for speaker recognition and speech recognition systems. The Mel-Frequency Cepstrum (MFC) is a representation of the short-term power spectrum of a signal, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. MFCCs are coefficients that collectively make up an MFC. The difference between the normal cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of signal, for example, in audio compression. Fig. 2 shows the computation flow chart of MFCCs:

- Take the Fourier transform of a signal in a window.
- Map the power/magnitude spectrum obtained above onto the mel scale, using triangular overlapping windows.
- Take the logs of the powers at each of the mel frequencies.
- Take the discrete cosine transform of the list of mel log powers.
- The final MFCC features are the amplitudes of the resulting spectrum.

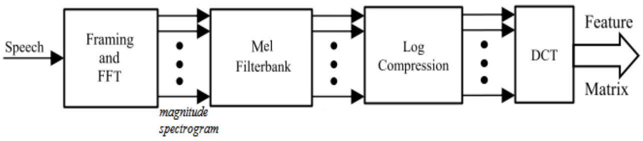

Fig. 2: Computation flow chart of MFCCs

There can be variations on this process, for example, differences in the shape or spacing of the windows used to map the scale [25].

Different types of low-level features have been proposed in the speaker recognition area with the motivations to improve the performance of MFCC baseline systems. Some features aim at improving performance under noisy conditions, such as Minimum Variance Distortionless Response (MVDR) and Mean Hilbert Envelope Coefficients (MHEC) features etc. Some features aim at improving performance on reverberant speech, such as Frequency Domain Linear Prediction (FDLP) features etc. Some features are motivated by providing complementary information that have been missed by MFCC features, such as Spectral Centroid Frequency (SCF), Spectral Centroid Magnitude (SCM), Fundamental Frequency Variation (FFV), Harmonic Structure Cepstral Coefficients (HSCC) features etc. We will overview these representative low-level features in the following sub-sections.

B. *MVDR*

Minimum Variance Distortionless Response (MVDR) [26] is a method for estimating a smoothed version of a signal's power spectrum. It is formulated as a filter design problem. Its warped version, warped MVDR (WMVDR) [27], is used to replace the FFT and filterbank steps in the extraction of MFCCs.

Suppose we have a signal *x[n]*, whose power spectrum is $S\left(e^{jw}\right)$. We want to estimate $S\left(e^{jw}\right)$ at a specific frequency w_{foi}. To do this, we design a *M*-th order moving-average filter, whose coefficients are $h_0, h_1, \cdots, h_M$ (they can be complex), and whose frequency response is $H_{w_{fi}}\left(e^{jw}\right)$. We want this filter to let the signal at frequency w_{foi} pass unchanged, while suppressing other frequencies as much as possible. To be precise, we minimize the energy that gets through when $x[n]$ is fed into the filter (minimum variance), under the constraint that the filter response at the specified frequency $H_{w_{foi}}\left(e^{jw}\right)$ is equal to 1 (Distortionless).

Define two vectors:

$$h = [h_0, h_1, \cdots, h_M]^T \qquad (1)$$

$$V\left(e^{jw}\right) = [1, e^{jw}, \cdots, e^{jMw}]^T \qquad (2)$$

Then the problem can be formulated as to minimize $h^H \Phi h$ subject to

$$V^H\left(e^{jw_{foi}}\right)h = 1 \qquad (3)$$

where Φ is an $(M+1)\times(M+1)$ Toeplitz autocorrelation matrix of the input signal $x[n]$ (we assume that $x[n]$ is real):

$$\Phi(i,j) = R_x\left[i-j\right] = \sum_n x[n-i]x[n-j]\big]$$

Using Lagrangian multipliers, we can solve the coefficients of the optimal filter:

$$h = \frac{\Phi^{-1}V\left(e^{jw_{foi}}\right)}{V^H\left(e^{jw_{foi}}\right)\Phi^{-1}V\left(e^{jw_{foi}}\right)} \qquad (4)$$

By substituting (4) into the objective function in (3), we can get the output energy of the filter centered at w_{foi}:

$$\begin{aligned} E\left(e^{jw_{foi}}\right) &= \int_0^{\pi}\left|H_{w_{foi}}\left(e^{jw}\right)\right|^2 S\left(e^{jw}\right)dw \\ &= \frac{1}{V^H\left(e^{jw_{foi}}\right)\Phi^{-1}V\left(e^{jw_{foi}}\right)} \end{aligned} \qquad (5)$$

The higher the filter order *M*, the better $E\left(e^{jw}\right)$ will approximate $S\left(e^{jw}\right)$, but it is always a "smoothed" version. Alternatively, when the filter order *M* is not very high, we can think of the output energies at a series of frequencies as the output energies of a filterbank. In order for MVDR to replace the spectrum estimation and filterbank steps in the MFCC extraction procedure, however, we still need frequency warping. Frequency warping is achieved with a bilinear transform:

$$\tilde{z}^{-1} = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}} \text{ or } e^{-j\tilde{w}} = \frac{e^{-jw}-\alpha}{1-\alpha e^{-jw}} \qquad (6)$$

The relationship between the original frequency and the warped frequency is shown in Fig. 3. In the case of $\alpha > 0$, if we pick a series of warped frequencies $\tilde{w}_k$ at equal intervals, the distribution of unwarped frequencies w_k will approximate the center frequencies of the Mel filterbank used in MFCC. At a sampling rate of 8 kHz, the warp factor $\alpha = 0.3624$ gives the best approximation of the Mel scale [28].

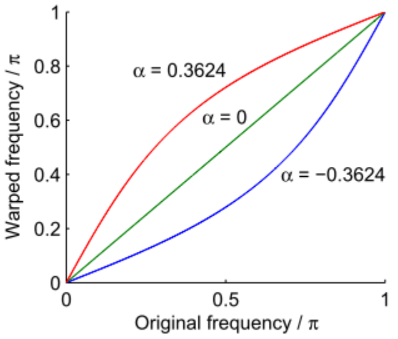

Fig. 3: Relationship between original and warped frequencies for different warp factors

We do not directly estimate $S(w)$ at the frequency w_k, because if we do so, the bandwidths of the filters would be the same, which would not give different resolutions in different frequency bands. Instead, we try to construct a warped signal whose power spectrum is a warped version of that of $x[n]$. Such a signal, however, would be infinite in time; therefore it would be hard to calculate its autocorrelation function accurately. But there's a trick available. The traditional autocorrelation function is defined as:

$$R_x[i] = \sum_n x[n]x[n-i] = \sum_n x[n]D^i x[n] \qquad (7)$$

where D is the delay operator, expressed as z^{-1} in the *z*-domain. We can replace it with a "warped delay operator" $\tilde{D}$, whose *z*-domain expression is given by the bilinear transform (Eq. (6)), to define a "warped autocorrelation function":

$$\tilde{R}_x[i] = \sum_n x[n]\tilde{D}^i x[n] \qquad (8)$$

The time-domain sequence corresponding to this autocorrelation function is hard to solve for, but we have found by experiment that the power spectrum of this signal, $\tilde{S}\left(e^{j\tilde{w}}\right)$, satisfies $\tilde{S}\left(e^{j\tilde{w}}\right)d\tilde{w} = S\left(e^{jw}\right)dw$

If we do the MVDR estimation on the warped frequency scale at a series of $\tilde{w}_k$ at equal intervals, we will have constructed a series of warped filters $\tilde{H}_{\tilde{w}_k}\left(e^{j\tilde{w}}\right)$ with equal bandwidths. Their unwarped counterparts, $H_{w_k}\left(e^{jw}\right) = \tilde{H}_{\tilde{w}_k}\left(e^{j\tilde{w}}\right)$, will have unequal bandwidths exactly like the Mel filterbank used in MFCC extraction. The energy values given by the warped MVDR will be equal to the output energies of the filterbank $H_{w_k}\left(e^{jw}\right)$ on the unwarped scale:

$$\begin{aligned}\tilde{E}\left(e^{j\tilde{w}_k}\right) &= \int_0^{\pi}\left|\tilde{H}_{\tilde{w}_k}\left(e^{j\tilde{w}}\right)\right|^2 \tilde{S}\left(e^{j\tilde{w}}\right)d\tilde{w} \\ &= \int_0^{\pi}\left|H_{w_k}\left(e^{jw}\right)\right|^2 S\left(e^{jw}\right)dw = E\left(e^{jw_k}\right)\end{aligned} \qquad (9)$$

Now the warped MVDR can replace the spectrum estimation and filterbank in MFCCs completely.

Let $\tilde{\Phi}$ be the warped Toeplitz autocorrelation matrix: $\tilde{\Phi}(i, j) = \tilde{R}_x[i-j] = \sum_n x[n]\tilde{D}^{|i-j|}x[n]$. The output energies will be:

$$\tilde{E}\left(e^{j\tilde{w}_k}\right) = \frac{1}{V^H\left(e^{j\tilde{w}_k}\right)\tilde{\Phi}^{-1}V\left(e^{j\tilde{w}_k}\right)} \qquad (10)$$

These values are passed through the log compression and DCT steps, and the resulting feature is called the warped minimum variance distortionless response (WMVDR) feature. Previous experiments [28, 29] have shown that WMVDR features achieved better performance than MFCC features for speaker and speech recognition under noisy conditions.

C. FDLP

Frequency domain linear prediction (FDLP) was proposed in [30] to improve the speech recognition performance on the reverberant speech. It is applied to improve speaker recognition performance when speech is corrupted by reverberation in [31]. Since reverberation is a long-term phenomenon, techniques based on short-term spectra generally result in worse performance as the models trained in clean environments fail to match the reverberant test conditions. A number of feature compensation techniques have been proposed in the past for speaker verification systems (for example, feature warping [32], RASTA processing [33] and cepstral mean subtraction (CMS) [34]). Although these techniques provide good improvements for short-term distortions like telephone channel conditions, they fail to suppress the long-term artifacts caused by room reverberation.

FDLP uses gain normalized temporal trajectories of sub-band energies to compensate for the room reverberation artifacts. Hilbert envelopes of sub-band signals are estimated by applying linear prediction in the frequency domain. For the reverberant speech, the sub-band Hilbert envelopes can be assumed to be a convolution of the sub-band Hilbert envelope of the clean speech with the sub-band Hilbert envelope of the room impulse response [30]. When linear prediction is applied in the frequency domain, the Hilbert envelope convolution model suggests that the artifacts present in reverberant speech affect the gain of the sub-band temporal envelopes. This causes the mismatch between the features trained from clean and reverberant environments. In order to reduce the mismatch, the proposed FDLP technique normalizes the gain of the auto-regressive (AR) models in narrow frequency bands. Gain normalization of the sub-band envelopes provides reasonable suppression of the reverberant artifacts in speech. The gain normalized sub-band envelopes are then integrated into short time frames (for example: 32ms with a shift of 10ms as in [31]) using a Hamming window. The frequency axis of the multiple linear sub-bands (96 sub-bands as in [31]) is warped according to the mel-scale. The output of the integration process provides a gain normalized mel-scale energy representation of speech similar to the mel-spectrogram obtained in conventional MFCC feature extraction. These mel-band energies are converted to cepstral coefficients by using a log operation followed by a DCT. Fig. 4 shows the FDLP feature extract flow chart. The procedure of gain normalization followed by the integration of the sub-band envelopes provides short-term mel-band energies that are similar to the mel-spectrogram in MFCC feature extraction. Thus it can be viewed as a pre-processing mechanism for the MFCC features to improve robustness in reverberant environments.

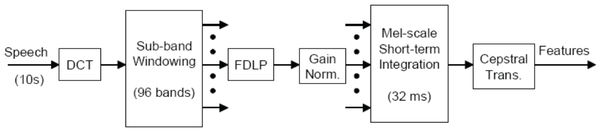

Fig. 4: Computation flow of FDLP [31]

Fig. 5 illustrates the comparison of MFCC features with CMS and FDLP features with gain normalization [31]. It plots C0 for MFCC features and FDLP features. In these plots, MFCC features are processed with CMS and the FDLP features are derived from gain normalized sub-band envelopes. FDLP features show better invariance to telephone distortions as well as reverberant artifacts compared to MFCC features. The automatic speech recognition results in [30] and speaker verification results in [31] have proved that FDLP features can provide significant improvements under reverberant conditions.

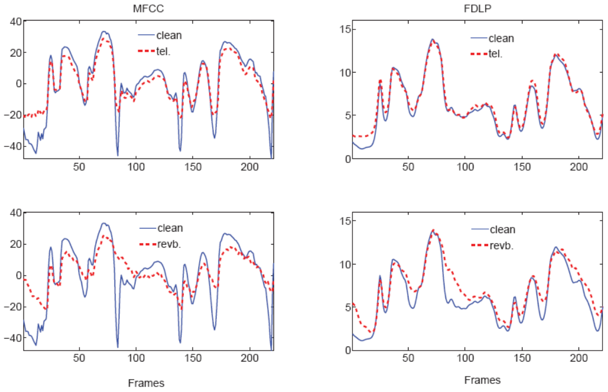

Fig. 5: Comparison of CMS for MFCC and gain normalization for FDLP [31]

D. MHEC

Mean Hilbert Envelope Coefficients (MHECs) feature is a type of new acoustic feature proposed in [35]. They were reported to perform better than traditional MFCC features under noisy conditions with different noise types and noise levels. They even display immunity to car noise: at an SNR of 0 dB, the identification accuracy of MFCC-based system drops to below 10%, while that of MHEC-based system stay above 90% [35].

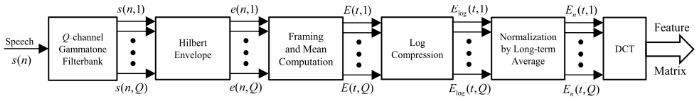

Fig. 6: Computation flow chart of MHECs

The steps to extract MHEC features from a speech signal are shown in Fig. 6. First, the speech signal is passed through a *Q*-channel Gammatone filterbank. The output of this is *Q* channels of time-domain signals. Second, the Hilbert envelope of the signal in each channel is calculated. Let the signal in channel *j* be $s(n, j)$, and its Hilbert transform $\hat{s}(n, j)$, then the analytic signal corresponding to $s(n, j)$ is:

$$s_a(n, j) = s(n, j) + i\hat{s}(n, j) \qquad (11)$$

The Hilbert envelope of $s(n, j)$ is defined as the modulus of this analytic signal:

$$e(n, j) = \sqrt{s^2(n, j) + \hat{s}^2(n, j)} \qquad (12)$$

Next, the signals $e(n, j)$ are blocked into frames, and the mean Hilbert envelope at frame *t* in channel *j* is calculated as:

$$E(t, j) = \frac{1}{N}\sum_{n=0}^{N-1} w(n)s(N_t + n, j) \qquad (13)$$

where N_t is the starting sample of frame *t*, *N* is the frame length, and $w(n)$ is a Hamming window of length *N*. The mean Hilbert envelope values $E(t, j)$ are compressed with logarithm, and normalized by the long-time average in each channel:

$$E_n(t, j) = \frac{E_{\log}(t, j)}{\frac{1}{T}\sum_{t=1}^{T} E_{\log}(t, j)} \qquad (14)$$

Finally, a DCT is applied to the normalized values $E_n(t, j)$ to produce the MHEC features.

Let us analyze why MHEC should outperform MFCC. Let’s look at how their extraction procedures differ. Two steps in their extraction procedures are the same: log compression and DCT. For MHEC, there’s a normalization step (Eq. (14)) in between. We suspect that a subtraction should be used instead of division in Eq. (14), since it doesn’t make sense to divide log values. If we change it to subtraction, this step is interchangeable with the DCT, because DCT is a linear transform. Then we can see that the normalization step is exactly CMS, so we can consider it as a post-processing step and exclude it from the feature extraction procedure itself.

Now the last two steps for extracting MFCCs and MHECs are identical. The input to the identical steps can be considered as an energy map across the frames and the channels, and the difference between MFCC and MHEC is just how this energy map is constructed. Two major differences can be seen immediately:

- MFCC goes into the frequency domain immediately with the STFT, while MHEC stays in the time domain.
- The filterbanks used by MFCC and MHEC are different. Although both the Mel filterbank and the Gammatone filterbank use a warped frequency scale to give lower frequencies higher resolution, the scale used by the Gammatone filterbank is more warped than the Mel scale. Also, since the Gammatone filterbank is implemented in the time domain, the filters are not perfect triangles in the frequency domain, but rather bells with the tails overlapping a lot with neighboring channels.

The last thing to contemplate is the Hilbert transform step in the extraction of MHECs. The Hilbert transform is simply a filter if viewed in the frequency domain, whose transfer function is:

$$H(w) = \begin{cases} -i & if \quad w > 0 \\ i & if \quad w < 0 \end{cases} \qquad (15)$$

Then the filter that produces an analytic signal from a real signal is:

$$H_a(w) = 1 + iH(w) = \begin{cases} 2 & if \quad w > 0 \\ 0 & if \quad w < 0 \end{cases} \qquad (16)$$

This indicates that the analytic signal has exactly twice as much energy as the real signal. The Hilbert transform step now seems useless, because if we simply dropped it, all that would happen is that the energy map would be divided by 2. But this is not accurate since there are other details involved: the mean Hilbert envelope doesn't have a square in its formula, so it's not exactly a representation of the energy; also, the Hilbert transform is applied to the entire signal instead of frame by frame. These facts may be a third factor that contributes to the difference of MHEC and MFCC.

We have identified three differences between MHEC and MFCC, but it is hard to say which one produces the benefit in MHEC. Very good noise robustness of MHEC features were reported in [35]. The authors also extended this type of feature for dealing with robust speaker recognition under reverberant mismatched conditions. More details can be found in the [36].

E. SCF/SCM

Although MFCC is the most popular feature used in speaker recognition systems, it doesn’t capture all the information contained in the acoustic signal. For example, while it integrates the energy in each channel of the filterbank, it ignores the distribution of the energy within a channel. Therefore, the information contained in small oscillations in the fundamental frequency of the speech, which may reflect idiosyncrasies of the speaker, is totally lost. This is usually made up by using frequency modulation (FM) features. However, FM features are computationally expensive. In order to make use of the complementary information and to avoid high computational cost at the same time, a pair of new acoustic features, spectral centroid frequency (SCF) and spectral centroid magnitude (SCM) were proposed in [37].

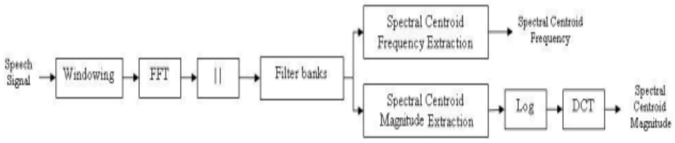
Speech Signal
Windowing
FFT
| |
Filter banks
Spectral Centroid Frequency Extraction
Spectral Centroid Frequency
Spectral Centroid Magnitude Extraction
Log
DCT
Spectral Centroid Magnitude

Fig. 7: Computation flow chart of SCF/SCM

The procedure to extract SCF and SCM are shown in Fig. 7. The steps up to "filterbanks" are identical to MFCC. The next two blocks calculate the "spectral centroid frequency" F_k and "spectral centroid magnitude" M_k in each channel k, frame by frame, as follows:

$$F_k = \frac{\sum_{f=l_k}^{u_k} f|S[f]w_k[f]|}{\sum_{f=l_k}^{u_k} |S[f]w_k[f]|} \qquad (17)$$

$$M_k = \frac{\sum_{f=l_k}^{u_k} f|S[f]w_k[f]|}{\sum_{f=l_k}^{u_k} f} \qquad (18)$$

Here S is the spectrum of the frame, f is the frequency at each point of the spectrum, l_k and u_k are the lower and upper bound of the frequency range of the k-th channel, and w_k is a triangle window spanning from l_k to u_k in the frequency domain. In plain words, F_k is the magnitude-weighted average frequency in the k-th channel, and M_k is the frequency-weighted average magnitude. The spectral centroid frequencies make up the SCF features directly. The spectral centroid magnitudes are then passed through log compression and DCT, just like in MFCC, to produce the SCM features.

SCF features can describe the bias of the distribution of the energy in the channels. This can provide complementary information to MFCC. On the other hand, SCM features are almost the same as MFCC, except that the magnitudes at each frequency point within a channel are weighted with the frequency before being added up. In [37] evaluation on the NIST 2006 database using a fusion of SCM-based and SCF-based subsystems, demonstrated relative improvements of 13% over the performance of an MFCC-only system. This supports the hypothesis that the combination of SCM and SCF carries more information than MFCC alone. SCF was also shown to perform significantly better than the previously proposed sub-band spectral centroid and frame-averaged FM features, such as [38, 39], for speaker recognition.

F. FFV

Fundamental Frequency (F0) features have been applied to the speaker recognition task in one of three main ways. Most commonly, global utterance-level statistics, such as mean and standard deviation, are estimated and compared between two utterances [40]. However, such statistics do not capture the shape of the F0 trajectory in time, a limitation which has been addressed in part through the inclusion of dynamic features in the feature vector [41]. A second approach to modeling F0 for speaker recognition aims to explicitly represent the F0 trajectory in time. Pitch contours between any two renderings of the same lexical content can be compared using dynamic time warping [42]. This approach is limited to text-dependent speaker recognition applications. Its extension to text-independent applications [43], comprising the third approach we mention, relies on the availability of a speech recognition system and requires a considerable amount of training data. In theory, when these requirements are met, the approach allows for the inference of conditional F0 feature densities, given other features such as energy trajectories or specific lexical contexts.

Instantaneous variation in pitch is normally computed by determining a single scalar, the fundamental frequency (F0), at two temporally adjacent instants and forming their difference. F0 represents the frequency of the first harmonic in a spectral representation of a frame of audio, and is undefined for signals without harmonic structure. In the context of speech processing applications, the localization of the first harmonic, and the subsequent differencing of two adjacent estimates, may be suboptimal feature compression and premature inference, since the goal of such applications is not the accurate estimate of pitch. In [44-46], a frame-level vector representation of the instantaneous change in F0, known as the fundamental frequency variation (FFV) spectrum was introduced. In particular, they leverage the fact that all harmonics are spaced equally in each of the two adjacent frames, and use every element of a spectral representation to yield a representation of the F0 delta. Unlike F0, the FFV spectrum remains well defined in the absence of voicing, and eliminates the need to localize a unique peak corresponding to the fundamental frequency, a process which is prone to error. In plain words, FFV spectrum estimates the fundamental frequency variation without estimating F0 directly. Fig. 8 illustrated the FFV spectrum computation paradigm.

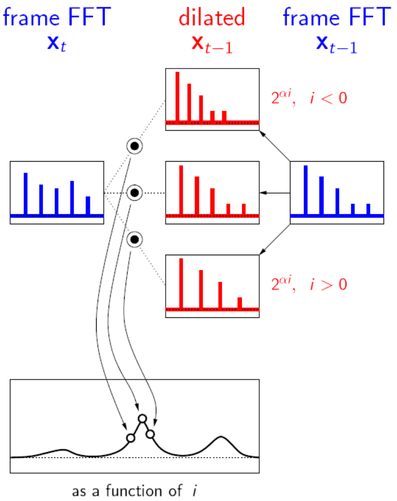

Fig. 8: Illustration of FFV spectrum computation

The experiments presented in [44] showed that FFV spectrum is suitable for standard Gaussian mixture modeling (GMM). Also, for speaker recognition, FFV information is complementary to that in standard frame-level MFCCs. Model-based combination with a GMM-FFV system reduces the classification error rate of the baseline GMM-MFCC system by 40-54%.

G. HSCC

In [47], it is argued that the estimation of pitch, an argmax operation in a transformed domain, does not serve the needs of speaker recognition. Those pitch errors which are considered most egregious may actually be just as speaker-discriminative as is pitch itself, if not more so. From a systems engineering perspective, the result is that the speaker discriminative information which pitch estimators may first compute, but then suppress or discard in the service of a better argmax hypothesis, is never made available to downstream components. As one would expect, that information appears to be mostly unrecoverable even in the face of costly and arcane modeling efforts. So the goal is to reverse all these information.

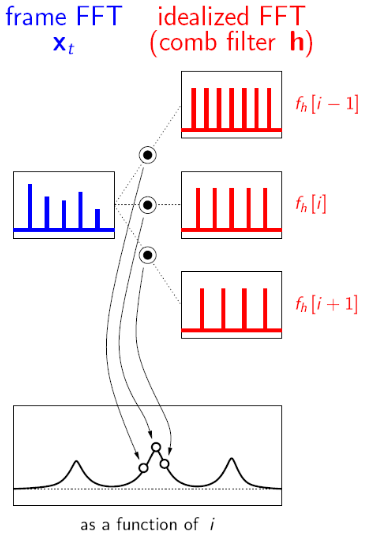

Fig. 9: Illustration of HSCC computation

Conceptually and functionally, the harmonic structure transform is related to the fundamental frequency variation (FFV) spectrum applied to speaker recognition in [44]. The FFV algorithm compares the FFT spectrum to a synthetic FFT spectrum, namely the frequency-dilated version of the true FFT spectrum from the preceding speech frame, over a range of logarithmically-spaced dilation factors. This yields a search space; nominally, its argmax is the relative change in F0 expressed in octaves per second. Here, the Harmonic Structure Cepstral Coefficients (HSCC) computation compares the FFT spectrum to another idealized FFT spectrum, namely the discrete-frequency comb filter with known F0, over a range of linearly-spaced F0 values. This also yields a search space; nominally, its argmax is the absolute F0 expressed in Hertz. For speaker recognition in [47], the entire search space is modeled rather than its argmax. Fig. 9 illustrates the computation paradigm of HSCC.

Experimental results in [47] have shown that HSCC features achieve comparable performance to an MFCC baseline under matched-channel and matched-multisession nearfield conditions. In contrast to prosodic features elsewhere, HSCCs are simple to compute, simple to model, and appear to require neither segmentation nor large quantities of training material. The HSCC feature space offers a paradigmatic shift in the processing of prosody for speaker recognition, and possibly for other speech processing tasks. The effectiveness of HSCC features under mismatched far-field conditions are to be investigated.

H. Multitaper MFCC

Multitaper is a technique for robust estimation of the power spectrum of a frame of acoustic signal [48]. In the standard procedure of MFCC extraction, the power spectrum is estimated by taking the square of the DFT spectrum of a windowed frame. Usually the Hamming window is used to suppress the sidelobes. Although the power spectrum obtained this way well preserves the information contained in the frame of signal, it also captures the variance. Therefore such power spectra usually look peaky, and vary greatly from frame to frame even when the acoustic signal stays relatively stationary. The multitaper technique in [49] addresses this problem by taking the average of multiple power spectra estimated with different window functions (also called “tapers”). This is inspired by the simple statistical fact that if a random variable X has a variance of σ^2, the average of n independent samples of X only has a variance of σ^2/n. The price paid for smaller variance is decreased frequency resolution, but for MFCC extraction, the frequency resolution is less important because the spectrum will be integrated by subbands anyway.

Fig. 10 compares the multitaper spectrum estimation for analysis of speech under additive factory noise corruption with the traditional single window/taper method [49]. The left panel shows spectrum estimate using the conventional single-taper (Hamming window) method whereas the right panel shows spectrum estimate using multipeak tapers with k = 6 tapers. The single-taper spectrum contains more details and shows large difference between the clean and the noisy frame. The multitaper spectra, in turn, are smooth and look visually more similar between the clean and the noisy version. This shows that the multitaper method achieves smaller variance.

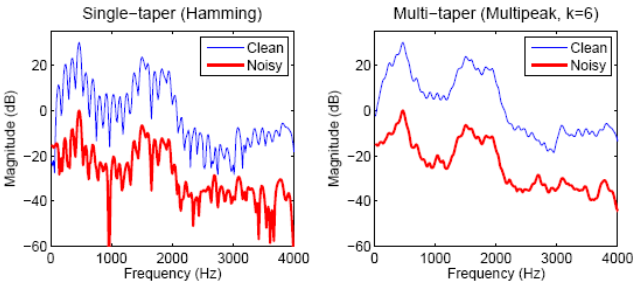

Fig. 10: Comparison of single window and multitaper methods under additive noise [49]

There exist several types of multitapers, such as Thompson, Multipeak, and SWCE (sine-weighted cepstral estimator), optimized for different criteria. In [49], the performances of these types of multitapers, as well as the optimal number of tapers, are investigated in a speaker verification task. The conclusion is that Multipeak and SWCE tapers are better than Thompson tapers, and the optimal number of tapers lies between 4 and 8. In summary, Multitapers are simple and robust alternative for the conventional single-window methods.

IV. Overview of high-level features

The speaker-specific information is the result of complex transformations occurring at different levels of the speech production: semantic, linguistic, articulatory, and acoustic [2, 50]. Humans rely on these several different types or levels of information in the speech signal to recognize others from voice alone. These can be the obvious nasality of a voice, a unique laughter, or the particular repeated word usage etc. In the previous section we overviewed the low-level features, which extract information at the acoustic level that deals with the spectral properties of the speech signal. For example, the dimensions of the vocal tract, or length and mass of vocal folds will define in some sense the fundamental and resonant frequencies, respectively. High level features capture speaker-specific linguistic and behavioral aspects not reflected at the cepstral level. While there is a long tradition of exploring higher-level features for speaker recognition – especially the use of pitch and other prosodic markers [51-53]) – systems incorporating them generally require significantly more data for adequate training or impose other constraints, such as text-dependency. In 2001, in response to growing interest in the use of higher-level features, NIST introduced the Extended Data task [54] based on the Switchboard-I corpus of conversational telephone speech. Unlike the traditional speaker recognition tasks, the Extended Data task provided multiple whole conversation sides for speaker training (for up to about 45 minutes of speech) and tested on whole conversation sides, thus enabling research on larger-scale features. The SuperSID team [24] at the Johns Hopkins 2002 Summer Workshops assembled to systematically explore a wide range of features for speaker recognition. These new sources of information have shown the promise not only for improvement in basic recognition accuracy by adding complementary knowledge, but also the possibility for robustness to acoustic degradations from channel and noise effects.

In this section we will overview several types of high-level features including prosodic features, phone features, lexical features, and cepstral derived features.

A. Prosodic Features

Prosodic features have been used for speaker recognition for a long history [41, 42]. With the continual advancement of tools (such as phone and speech recognition systems), increasingly large amounts of speech from a speaker, researchers have explored diverse collection of prosodic features for speaker recognition spanning many types of pitch, energy, duration, and the combination of them all.

In [55], the authors proposed two approaches for exploring prosodic features: 1) Pitch and Energy Distributions, a feature vector consisting of per-frame log pitch, log energy, and their first derivatives was used for speaker verification; 2) Pitch and Energy Track Dynamics, the aim was to learn pitch and energy gestures by modeling the joint slope dynamics of pitch and energy contours. A sequence of symbols describing the pitch and energy slope states (rising, falling), segment duration, and phoneme or word context is used to train an n-gram classifier. This combined well with absolute pitch and energy distributions in 1), indicating it is capturing new information about the pitch and energy features.

In [56], a collection of (19) prosodic statistics from duration and pitch related features, such as mean and variance of pause durations and F0 values per word, were extracted from each conversation side. These feature vectors were used in a k-nearest neighbor classifier for speaker verification and the experimental results again indicate that prosodic features are useful and provide new information for speaker recognition.

In [57], a more advanced paradigm for the extraction of prosodic features from speech was presented. Syllables are estimated automatically using the output of an automatic speech recognition (ASR) system, and more than a hundred measurements based on pitch and energy signals, along with the duration of the syllable and its constituents (onset, nucleus, and coda) are extracted over each syllable. These features are called syllable-based NERFs (non-uniform extraction region features), or SNERFs. The extracted features have some particular characteristics that make them harder to model than the standard spectral features: they have mixed continuous/discrete distributions, they are much sparser than low-level features, and they have undefined values. A system based on these features has been the best performing prosodic system on NIST speaker recognition evaluation (SRE) data published in the speaker recognition literature, since its introduction in 2005. Despite its success, these syllable-based features have not been widely used in the community, mainly because they are not simple to extract. They require ASR output and, even though they are all basically simple measurements over the pitch, energy and duration patterns, their implementation is laborious. In [58], a simplified version was proposed which uses ASR-independent regions based on the valleys found in the energy signal and uses polynomial approximations of the pitch and energy signals, along with the length of the regions. The features are not only simpler in extraction, they are also simpler to model since they are all continuous and do not contain undefined values. This makes the Joint Factor Analysis (JFA) modeling of these features possible.

B. Phone Features

Phone features are also known as acoustic tokenization features or phonetic features [59]. It generally employs unconstrained phone recognition essentially as a means by which to discretize the acoustic space and enable acoustic sequence modeling. In [60], an alternative acoustic tokenization approach using GMM-generated events was proposed. The “phonetic” speaker models capture an assortment of speaker-dependent factors, including spectral characteristics, pronunciation idiosyncrasies, and lexical preferences, and can therefore be difficult to interpret. The basic approach builds the speaker specific and universal background phone N-gram models based on the best phone decoding hypothesis and then evaluates likelihood ratios of speaker specific and universal phone N-gram models [61]. In this approach, the time sequence of phones coming from a bank of open-loop phone recognizers is used to capture some information about speaker-dependent pronunciations. Results can be improved by running several language-dependent or gender-dependent phone recognizers. Multiple phone streams are scored independently and fused at the score level. A refinement approach in [62] improved this approach by replacing phone N-gram with binary decision tree. With a binary tree, it is possible to use large context without exponential memory expansion. It is also easier to run some adaptation and recursive smoothing techniques on the binary decision tree that are important for sparse data sets. Improved approach in [63] examines capturing cross-stream information from the multiple phone streams simultaneously. Improvements can also be obtained by modeling not just the top hypothesized phone sequence from the recognizer, but rather the expected phone N-gram frequencies extracted from phone recognition lattices [64]. In [65], lattice-based phone N-gram frequency modeling is combined with word conditioning. The phone N-grams occurring in specific words and frequent phrases are tallied and assembled into a more detailed feature vector that is modeled by SVMs. In [66], a unique combination of phone- and word-based modeling is described, trying to learn speaker-dependent pronunciations. The output of an unconstrained phone recognizer is time-aligned with the phone sequence from a word recognizer, and the conditional probabilities of the former given the latter are modeled. Another important advance in high-level phonetic speaker recognition was the use of SVMs instead of likelihood models to model phone N-gram frequencies [67]. Although approaches based on unconstrained phone recognition get about 2 to 3 times the EER of the best cepstral systems, they can provide substantial gains when combined with the low-level cepstral features.

C. Lexical Features

Lexical features (a speaker's preference of word usages) are one of the earliest types of higher-level features explored for speaker recognition. Early work in authorship authentication tried to use lexical N-gram statistics to discriminate different authors. The approach did not produce a significant gain at the time, presumably because of the brief training and test samples used in task definitions at the time.

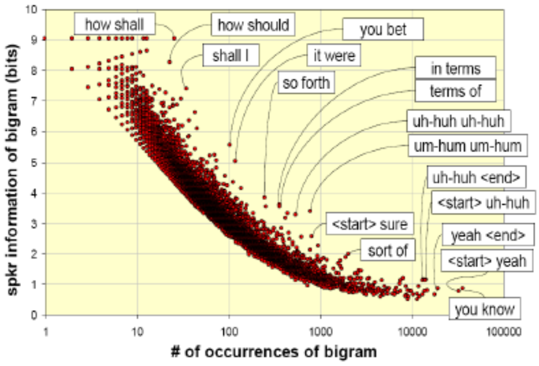

Fig. 11: Illustration of speaker information captured by bigrams [68]

However, under the extended data condition, it was found in [68] that rates of idiosyncratic word N-grams (for example, "how shall", as shown in Fig. 11) could be used to help discriminate speakers.

In the improved approach presented in [69], the relative frequencies of frequent word unigrams, bigrams, and trigrams are obtained and assembled into a feature vector that is modeled by SVMs. More recently, the approach has been extended to encode the duration (slow/fast) of frequent word types as part of the N-gram frequencies [70]. This technique explicitly models both N-gram frequencies and word durations. It can be considered to capture lexical, pronunciation, and prosodic characteristics of the speaker simultaneously. Lexical features have also proved to provide additional gains when combined with cepstral features.

D. Cepstral-derived Features

Speaker recognition systems based on low-level cepstral features are usually the best performing systems. In [71], the MLLR approach was proposed. This approach is based on Cepstral-derived features. It uses speaker-specific model adaptation transforms from a speech recognizer (either phone or word level) as features, modeled by a support vector machine (SVM). Instead of cepstral features, it uses the difference between speaker-adapted Gaussian means and corresponding speaker-independent means as features. This difference is expressed as the coefficients of an affine transform that rotates and shifts the speaker-independent model to obtain a speaker-dependent model, computed with maximum likelihood linear regression. Furthermore, the Gaussian models used in this approach are not unstructured GMMs but the detailed context-dependent phone models used in a speech recognizer, making the resulting features text independent. This approach has the advantage that features are text independent while being shared among all instances of a given phone, thus avoiding the data fragmentation implied by the conditioning on words. Transforms specific to different phone classes are combined for greater representational detail. Experimental results have shown that although the MLLR transform features cannot perform better than the conventional Cepstral features, they can provide substantial gains when combined with the latter [72]. This shows that cepstral-derived high-level features can provide complementary information.

V. CONCLUSIONS

What is it in the speech signal that conveys speaker identity? This is one of the central questions addressed by automatic speaker recognition research. Speaker feature extraction is certainly one of the most important components in any speaker recognition systems. It is pretty obvious that we (humans) rely on several different types or levels of information in the speech signal to recognize others from voice alone. We can roughly categorize the speaker features into low-level features, related to physical traits of the vocal apparatus, and high-level features, related to learned habits and style. We have overviewed several types of low-level features and high-level features in this paper. Which features should one use? There are no standard rules for choosing among different features. It depends on the intended application, computing resources, amount of speech data available (for both development purposes and in run-time) and whether the speakers are cooperative or not. For beginners in speaker recognition research area, the short-term spectral features would be a good choice since they are easy to compute and yield good performance. High-level features are believed to be more robust because they are related to speaker's learned habits and style, which normally not be affected by noises. However, High-level features also require considerably more complex front-end, such as automatic speech recognizer, and thus more training data to build a reliable system.

In conclusion, there does not exist globally "best" feature for speaker recognition yet, but the choice is a trade-off between speaker discrimination, robustness, and practicality. Fusion of multiple features often provides additional gains.

REFERENCES

[1] S. Furui, "Recent Advances in Speaker Recognition," *Pattern Recognition Letters, 18, (1997), 859-872*.

[2] J. P. Campbell, "Speaker Recognition: A Tutorial," *Proceedings of the IEEE, 85, 9, (1997), 1437-1462*.

[3] S. Tranter, and D.A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech and Language Processing 14, 5 (September 2006), 1557–1565*.

[4] F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D.A. Reynolds, "A Tutorial on Text-Independent Speaker Verification," *EURASIP Journal on Applied Signal Processing 2004, 4 (2004), 430–451*.

[5] D.A. Reynolds, "An Overview of Automatic Speaker Recognition Technology," *In Proceedings of ICASSP '2002 (Orlando, Florida, 2002), pp. 4072-4075*.

[6] T. Kinnunen and H. Li, "An Overview of Text-Independent Speaker Recognition: From Features to Supervectors," *Speech Communication 2010, 52 (2010), 12-42*.

[7] J. J. Wolf, "Efficient Acoustic Parameters for Speaker Recognition," *JASA, 51, 6, (1972), 2044-2056*.

[8] F. Nolan, "The Phonetic Bases of Speaker Recognition," *1st ed. New York: Cambridge University Press, 1983*.

[9] M. BenZeghiba, and H. Bourland, "On the combination of speech and speaker recognition," *In Proc. Eurospeech, Geneva, Switzerland, September 2003, pp. 1361–1364*.

[10] L. Heck, and D. Genoud, "Combining speaker and speech recognition systems," *In Proc. ICSLP, Denver, Colorado, USA, September 2002, pp. 1369–1372*.

[11] E. Hansen, R. Slyh, and T. Anderson, "Speaker recognition using phoneme-specific GMMs," *In Proc. Speaker Odyssey: the Speaker Recognition Workshop, Toledo, Spain, May 2004, pp. 179–184*.

[12] S. Kajarekar, and H. Hermansky, "Speaker verification based on broad phonetic categories," *In Proc. Speaker Odyssey: the Speaker Recognition Workshop, Crete, Greece, June 2001, pp. 201–206*.

[13] A. Park, and T. Hazen, "ASR dependent techniques for speaker identification," *In Proc. ICSLP, Denver, Colorado, USA, September 2002, pp. 1337–1340*.

[14] L. R. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," *Prentice-Hall, NJ, 1993*.

[15] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing 29, 2 (April 1981), 254–272*.

[16] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing 10, 1, January 2000, 19–41*.

[17] D. Reynolds, and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing 3, January 1995, 72–83*.

[18] J. Naik, L. Netsch, and G. Doddington, "Speaker verification over long distance telephone lines," *In Proc. ICASSP, Glasgow, May 1989, pp. 524–527*.

[19] M. BenZeghiba, and H. Bourland, "User-customized password speaker verification using multiple reference and background models," *Speech Communication 48, 9, September 2006, pp. 1200–1213*.

[20] L. Heck, Y. Konig, M. Sonmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," *Speech Communication 31, June 2000, 181–192*.

[21] B. Yegnanarayana, and S. Kishore, "AANN: an alternative to GMM for pattern recognition," *Neural Networks 15, April 2002, pp. 459–469*.

[22] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language 20, 2-3 April 2006, 210–229*.

[23] V. Wan and W. Campbell, "Support vector machines for speaker verification and identification," *Proceedings of the 2000 IEEE Signal Processing Society Workshop, vol.2, no., pp.775-784 vol.2, 2000, doi: 10.1109/NNSP.2000.890157*.

[24] D. Reynolds, W. Andrews, J. P. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "The SuperSID Project: Exploiting High-level Information for High-accuracy Speaker Recognition," *In Proceedings of ICASSP '2003 (Hong Kong, 2003), pp. 784-787*.

[25] F. Zheng, G. Zhang and Z. Song, "Comparison of Different Implementations of MFCC," J. Computer Science & Technology, 16(6): 582–589, 2001.

[26] M. Murthi, B. Rao, "Minimum variance distortionless response (MVDR) modeling of voiced speech," *In Proc. ICASSP, vol.3, no., pp.1687-1690 vol.3, 21-24 Apr 1997*.

[27] M. Wolfel, J. McDonough, and A. Waibel, "Minimum variance distortionless response on a warped frequency scale," *In EUROSPEECH-2003, 1021-1024*.

[28] M. Wolfel, "Robust Automatic Transcription of Lectures," *Ph.D. thesis, Universität Fridericiana zu Karlsruhe, 2009*.

[29] Q. Jin, R. Li, Q. Yang, K. Laskowski, and T. Schultz, "Speaker identification with distant microphone speech," *IEEE ICASSP, pp. 4518-4521, 2010*.

[30] Thomas, S., Ganapathy, S. and Hermansky, H., "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Proc. Letters, Vol. 15, pp. 681-684, 2008*.

[31] S. Ganapathy, J. Pelecanos and M. K. Omar, "Feature normalization for speaker verification in room reverberation," *IEEE ICASSP, pp. 4836-4839, 2011*.

[32] J. Pelecanos, and S. Sridharan, "Feature warping for robust speaker verification", *Proc. Speaker Odyssey 2001 Speaker Recognition Workshop, Greece, pp. 213-218, 2001*.

[33] H. Hermansky, and N. Morgan, "RASTA processing of speech," *IEEE Trans. on Speech and Audio Process., Vol. 2, pp. 578-589, 1994*.

[34] S. Furui, "Cepstral analysis technique for automatic speaker verification", *IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 29, pp. 254-272, 1981*.

[35] S. O. Sadjadi and J. H. L. Hansen, "Assessment of single-channel speech enhancement techniques for speaker identification under mismatched conditions," *ISCA Interspeech, pp. 2138-2141, 2010*.

[36] S. O. Sadjadi and J. H. L. Hansen, "Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions," *In Proc. ICASSP 2011, pp. 5448-5451*.

[37] J. M. K. Kua, T. Thiruvaran1, M. Nosratighods, E. Ambikairajah, J. Epps, "Investigation of spectral centroid magnitude and frequency for speaker recognition," *In Proc. Speaker Odyssey: the Speaker Recognition Workshop, Brno, Czech Republic,2010, pp. 34-39*.

[38] T. Thiruvaran, E. Ambikairajah, and J. Epps, "Analysis of band structures for speaker-specific information in fm feature extraction," *Proc. INTERSPEECH, 2009*.

[39] P. Maragos, J. F. Kaiser, and T. F. Quatieri, "Energy separation in signal modulations with application to speech analysis," *IEEE Transactions on Signal Processing, vol. 41, no. 10, pp. 3024–51, 1993*.

[40] M. Carey, E. Parris, H. Lloyd-Thomsa, and S. Bennett, "Robust prosodic features for speaker identification," *in Proc. ICSLP, Philadelphia PA, USA, 1996, pp. 1800–1803*.

[41] K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," *in Proc. ICSLP, Sydney, Australia, 1998, pp. 3189–3192*.

[42] B. Atal, "Automatic speaker recognition based on pitch contours," *in J. ASA, 1972, vol. 52, pp. 1687–1697*.

[43] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey, "Modeling prosodic dynamics for speaker recognition," *in Proc. ICASSP, Hong Kong, China, 2003, pp. 19–41*.

[44] K. Laskowski and Q. Jin, "Modeling instantaneous intonation for speaker identification using the fundamental frequency variation spectrum," *in Proc. ICASSP 2009*.

[45] K. Laskowski, J. Edlund, and M. Heldner, "Learning prosodic sequences using the fundamental frequency variation spectrum," *in Proc. SPEECH PROSODY, Campinas, Brazil, 2008*.

[46] K. Laskowski, J. Edlund, and M. Heldner, "An instantaneous vector representation of delta pitch for speaker-change prediction in conversational dialogue systems," *in Proc. ICASSP, Las Vegas NV, USA, 2008, pp. 5041–5044*.

[47] K. Laskowski and Q. Jin, "Modeling Prosody for Speaker Recognition: Why Estimating Pitch May Be a Red Herring," *in Proc. Speaker Odyssey: the Speaker Recognition Workshop, Brno, Czech Republic, 2010.*

[48] M. Hansson and G. Salomonsson, "A multiple window method for estimation of peaked spectra," *IEEE Trans. on Sign. Proc., vol. 45, no. 3, pp. 778–781, Mar. 1997*.

[49] T. Kinnunen, R. Saeidi, J. Sandberg, and M. Hansson-Sandsten, "What else is new than the Hamming window? Robust MFCCs for speaker recognition via multitapering," *in Proc. InterSpeech 2010*.

[50] B. S. Atal, "Automatic Recognition of Speakers from Their Voices," *Proceedings of the IEEE, 64, 4, (1976), 460-475*.

[51] B.S. Atal, "Automatic speaker recognition based on pitch contours," *JASA, vol. 52, pp.1687-1697, 1972*.

[52] M.J. Carey, E.S. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," *Proc. ICSLP-96, Philadelphia, Nov 1996*.

[53] K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," *Proc. ICSLP-98, Sydney, Dec 1998*.

[54] NIST 2001 Speaker Recognition Evaluation website: http://www.nist.gov/speech/tests/spk/2001/index.htm.

[55] A. Adami, R. Mihaescu, D. Reynolds, and J. Godfrey, "Modeling Prosodic Dynamics for Speaker Recognition," *ICASSP 2003*.

[56] B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. Reynolds, B. Xiang, "Using Prosodic and Conversational Features for High-performance Speaker Recognition: Report from JHU WS'02," *ICASSP 2003*.

[57] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication, vol. 46, no. 3-4, pp. 455–472, 2005, Special Issue on Quantitative Prosody Modelling for Natural Speech Description and Generation*.

[58] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing, vol. 15, no. 7, pp. 2095–2103, Sept. 2007*.

[59] E. E. Shriberg, "Higher Level Features in Speaker Recognition," *In C. Müller(Ed.) Speaker Classification I.Volume 4343 of Lect ure Notes in Computer Science / Artificial Intelligence. Springer : Heidelberg / Berlin / New York, pp. 241-259*.

[60] N. Scheffer, J. Bonastre, "Speaker Detection using Acoustic Event Sequences," *In Proc. Eurospeech 2005*.

[61] W. Andrews, M. Kohler, J. Campbell, J. Godfrey, and J. Hernandez-Cordero, "Gender-dependent Phonetic Refraction For Speaker Recognition," *ICASSP 2002*.

[62] J. Navratil, Q. Jin, W. Andrews, and J. Campbell, "Phonetic Speaker Recognition Using Maximum Likelihood Binary Decision Tree Models," *ICASSP 2003*.

[63] Q. Jin, J. Navratil, D. Reynolds, J. Campbell, W. Andrews, J. Abramson, "Combining Cross-Stream And Time Dimensions In Phonetic Speaker Recognition," *ICASSP 2003*.

[64] A. Hatch, B. Peskin, A. Stolcke, "Improved Phonetic Speaker Recognition Using Lattice Decoding," *In Proc. ICASSP. 2005*.

[65] H. Lei, N. Mirghafori, "Word-Conditioned Phone N-Grams for Speaker Recognition," *In Proc. ICASSP, 2007*.

[66] D. Klusacek, J. Navratil, D. Reynolds, J. Campbell, "Conditional Pronunciation Modeling in Speaker Detection," *In Proc. ICASSP 2003*.

[67] W. Campbell, J. Campbell, D. Reynolds, D. Jones, and T. Leek, "Phonetic Speaker Recognition with Support Vector Machines," *Advances in Neural Information Processing Systems 16, 1377–1384, 2004*.

[68] G. Doddington, "Speaker Recognition based on Idiolectal Differences between Speakers," *Eurospeech, Vol. 4, pp. 2517-2520, 2001*.

[69] S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and J. Zheng, "SRI's 2004, NIST Speaker Recognition Evaluation System," *In Proc. ICASSP. 2005*.

[70] G. Tur, E. Shriberg, A. Stolcke, S. Kajarekar, "Duration and Pronunciation Conditioned Lexical Modeling for Speaker Verification," *In Proc. of Interspeech, 2007*.

[71] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, A. Venkataraman, "MLLR Transforms as Features in Speaker Recognition," *In Proc. Interspeech, 2005*.

[72] K. Boakye, B. Peskin, "Text-Constrained Speaker Recognition on a Text-Independent Task," *In Proc. Odyssey: Speaker and Language Recognition Workshop, Toledo, Spain, 2004*.