Odyssey 2016
21-24 Jun 2016, Bilbao, Spain

A Low-Power Text-Dependent Speaker Verification System with Narrow-Band Feature Pre-Selection and Weighted Dynamic Time Warping

Qing He[⋆] *Gregory W. Wornell*[⋆] *Wei Ma*[†]

[⋆] EECS & RLE, MIT, Cambridge, MA 02139, USA
[†] Texas Instruments Inc, Santa Clara, CA, 95051, USA

Abstract

To fully enable voice interaction in wearable devices, a system requires low-power, customizable voice-authenticated wake-up. Existing speaker-verification (SV) methods have shortcomings relating to power consumption and noise susceptibility. To meet the application requirements, we propose a low-power, text-dependent SV system comprising a sparse spectral feature extraction front-end showing improved noise robustness and accuracy at low power, and a back-end running an improved dynamic time warping (DTW) algorithm that preserves signal envelope while reducing misalignments. Without background noise, the proposed system achieves an equal-error-rate (EER) of 1.1%, compared to 1.4% with a conventional Mel-frequency cepstral coefficients (MFCC)+DTW system and 2.6% with a Gaussian mixture universal background (GMM-UBM) based system. At 3dB signal-to-noise ratio (SNR), the proposed system achieves an EER of 5.7%, compared to 13% with a conventional MFCC+DTW system and 6.8% with a GMM-UBM based system. The proposed system enables simple, low-power implementation such that the power consumption of the end-to-end system, which includes a voice activity detector, feature extraction front-end, and back-end decision unit, is under 380 μW.

1. Introduction

With the increasing popularity of mobile devices and wearable electronics, it is competitively advantageous to enable full voice interaction beginning with voice-authenticated wake-up. An ideal SV system for such applications requires a combination of security, low power usage, noise resiliency, and customized passphrases. In consideration of these constraints, we develop a novel text-dependent SV system in which the user defines his or her own short passphrase ($<$ 1s in duration) by enrolling a small number of samples. Some characteristics of our system include: (1) low-power sparse spectral feature extraction front-end; (2) DTW with adaptive signal envelope distortion constraints; (3) robustness to noisy environments by adjusting features selection to noise spectral conditions.

Existing techniques for SV can be 'text-independent' [1, 2] or 'text-dependent' [3]. Text-independent SV has the flexibility to recognize a speaker's identity without constraints on the speech (i.e., any word can be uttered during enrollment and testing). However, it usually requires a large amount of speaker-specific enrollment data (typically more than 30s) to extract sufficient useful features to discriminate between speakers. A performance penalty is paid for the high degree of variability in speech contents. On the other hand, text-dependent SV assumes the utterances being tested are the same as, or a subset of, the enrollment lexicon. Therefore, a more specialized model can be built, achieving better accuracy using shorter enrollment (usually less than 8s). Our applications falls into the category of text-dependent SV.

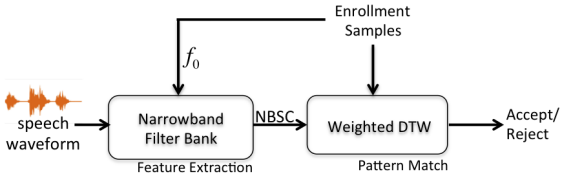

Figure 1: *Block diagram of our proposed system including the feature extraction front-end, which consists of K ($\sim$ 10) narrowband filters with fixed bandwidth ($\sim$ 300Hz) centered around multiples of f_0 (estimated from enrollments). All or a subset of the K features are used for decision making depending on the background noise spectrum. The back-end is a weighted-DTW algorithm, in which the adaptive warping constraint is inversely proportional to the temporal total signal energy.*

A successful technique in SV is to leverage speech across a cohort of speakers to train a background model as a prior, which is then used to make speaker-specific refinements, see e.g. Gaussian mixture universal background models (GMM-UBM) [4, 5], i-vectors [6], deep neural networks (DNNs) [7, 8] and hidden-Markov-models (HMMs) [9, 10]. Since these methods require background model training on *a priori* known passphrases, it is not suitable for our application due to lack of training data besides the few samples of user enrollment.

Our system solves SV as a pattern matching problem based on similarity measures between the input signal and the enrollment samples directly. As shown in Figure 1, the process includes two stages: feature extraction and pattern matching on features. We develop novel designs in both stages:

Feature extraction: In speech recognition applications, the Mel-frequency cepstral coefficients (MFCCs) [11, 12] are widely used and have yielded good performance. Nevertheless, the extraction process usually involves fast sampling, a large number of filters (26 to 40) and high-rate processing, that are associated with high computation and power costs. We propose a low-complexity, power-efficient feature extraction front-end that completes feature extraction in two simple steps: (1) filtering the analog speech signal using a handful of ($\sim$ 10) fixed-width narrowband filters, whose center frequencies are chosen according to the fundamental frequency f_0 estimated from enrollments; and (2) taking the logarithm of the filterbank power. We show through analysis and experiments that these sparse features retain essential speech information and offer the benefits of low-power implementation, high verification accuracy and noise robustness by automatically discarding features with

doi: 10.21437/Odyssey.2016-1

high noise occupancy. The low-dimensionality of the features also reduces the back-end computation since the complexity of the back-end SV algorithm is proportional to the feature dimension.

Similarity measure: Speech pattern matching is often performed with DTW [13, 14]. Variations of the DTW algorithm are developed to constrain the warping path [14, 15], to add weightings to the feature vector based on the intraspeaker variability for each feature element [13] or to add weighting based on temporal characteristics of the warping path [14]. One common issue associated with applying these methods to our application is they either apply too much warping that distorts the signal characteristics or insufficient warping to compensate the long pauses between words. We propose a modified version of the DTW algorithm that adaptively adjusts warping constraints based on the signal's total energy envelope, thus restricting excessive distortion on the main signal envelope while still allowing sufficient time warping to take care of long pauses between words and speaking speed variations.

We describe and analyze the narrow-band feature extraction scheme in Section 2 and introduce the weighted-DTW algorithm in Section 3. In Section 4, we compare our system performance with the conventional constrained DTW with MFCC features approach and with the widely used fixed-text SV method based on GMM-UBM.

2. Speaker-dependent Narrowband Feature Extraction

In this section, we describe our narrowband feature extraction front-end and justify its design.

2.1. Description of the front-end

Our front-end consists of the following steps:

1. Estimate the average fundamental frequency f_0 from each enrollment sample. The final f_0 is determined as the mean of the f_0 estimation from all enrollments. (This step is only performed once during initialization. As a result, even though it uses the full spectrum, it does not affect power consumption for prediction.)

2. Let B be the bandwidth of the speech signal and K, the number of filterbanks. The center frequencies of the filterbanks are at

$$k \times f_0 \times \left\lfloor \frac{B}{f_0 K} \right\rfloor, \quad 1 \leq k \leq K.$$

   These K filters are evenly spaced across the frequency spectrum and B/K is approximately the spacing between adjacent filters. K is selected such that B/K is smaller than a threshold parameter θ_h (i.e., $K \geq B/\theta_h$). The parameter θ_h is dependent on the property of speech (defined in Section 2.2.2). Usually, $\theta_h = 2$ cycle/kHz, is sufficient. For example, with B = 4 kHz and $\theta_h = 2$ cycle/kHz, $K \geq 8$. The bandwidth of the filters is narrow ($\sim$ 200 Hz) and Section 2.2.2 discusses the effects of the filter bandwidth.

3. The logarithms of the narrowband filter energies are aggregated and framed to form the narrowband spectral coefficients (NBSCs). The dimension of the feature vector is equal to the number of bands K.

4. Assuming knowledge of the noise spectrum, a subset of the NBSCs, where the SNR is the highest, is retained as features. The remaining bands are discarded.

2.2. Analysis

The purpose of a speech-processing front-end can be thought of as dimension reduction of a high-dimensional speech sample to a few representational coefficients. In speech applications, cepstral domain features such as MFCCs are widely used [11, 12] because speech information is sparse in the cepstral domain [16, 17, 18]. Cepstral coefficients are Fourier duals of the logarithm of the power spectrum density (PSD) of a time-domain speech segment.

Given that a major driver of power consumption in cepstral domain feature extraction is high-rate sampling and preprocessing required to transform the signal to the cepstral domain, our aim here is to show that substantially the same features can be extracted using a set of narrowband filters directly from the raw speech waveform by exploiting certain properties of speech and the desired application, thus reducing power consumption. We begin by reviewing the process by which cepstral domain features such as MFCCs are extracted.

2.2.1. Cepstral analysis of speech

Speech signals, denoted by $s(t)$, can be modeled as a time-domain convolution between the excitation signal $e(t)$ and the vocal tract modulation function $h(t)$:

$$s(t) = e(t) * h(t). \tag{1}$$

For voiced sounds, $e(t)$ is a periodic glottal pulse with fundamental frequency f_0. For unvoiced sounds, $e(t)$ can be modeled as a stochastic noise sequence. It is understood that most of the speech information is embedded in the time-varying vocal tract function $h(t)$ [19, 20, 12].

The convolution relationship in (1) becomes multiplication in the frequency domain:

$$S(f) = E(f) \cdot H(f). \tag{2}$$

Taking the logarithm of the PSD, the multiplication operation is converted to summation:

$$\hat{S}(f) = \hat{E}(f) + \hat{H}(f), \tag{3}$$

where $\hat{S}(f)$, $\hat{E}(f)$ and $\hat{H}(f)$ denote $\log|S(f)|$, $\log|E(f)|$ and $\log|H(f)|$, respectively. By taking the inverse Fourier-transform (IFT) of the logarithm of the PSD, the signal is transformed to the cepstral domain. Let us use $\hat{s}(\tau)$, $\hat{e}(\tau)$ and $\hat{h}(\tau)$ to denote IFT($\hat{S}(f)$), IFT($\hat{E}(f)$) and IFT($\hat{H}(f)$), respectively. Then, it follows from the linearity of IFT and (3) that:

$$\hat{s}(\tau) = \hat{e}(\tau) + \hat{h}(\tau). \tag{4}$$

Figure 2 illustrates the process of cepstral analysis. In Figure 2-(b) and (c), the narrow spikes (solid lines) are due to the excitation component $\hat{E}(f)$ and the signal envelopes (dashed lines) correspond to the modulation function $\hat{H}(f)$ and high frequency falloff of speech. When transformed to the cepstral domain, the speech signal becomes sparse (Figure 2-(d) and (e)). The low-quefrency component corresponds to vocal-tract modulation: $\hat{h}(\tau)$, and the higher-quefrency component corresponds to the excitation signal, $\hat{e}(\tau)$. Usually, the location of

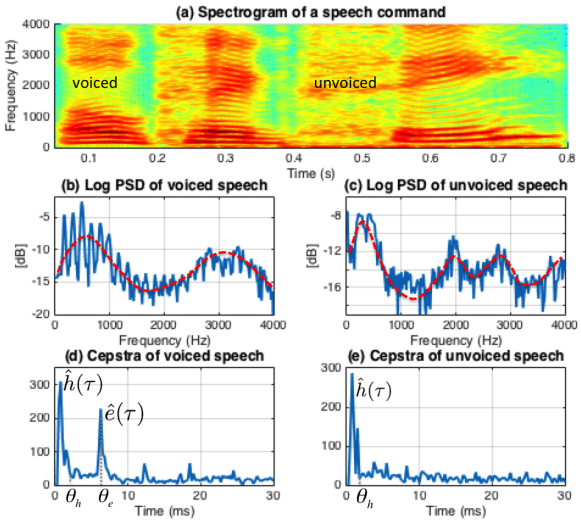

Figure 2: *Cepstral analysis of a speech sample. (a): the spectrogram of a speech command. (b) and (c) show the logarithm of the PSD of the speech segment (solid line). (d): the signal cepstrum is sparse and consists of two components: $\hat{h}(\tau)$ and $\hat{e}(\tau)$. (e) shows the cepstra of the unvoiced frame, where only the $\hat{h}(\tau)$ component is present.*

the excitation component θ_e is much higher than the cutoff frequency of $\hat{H}(f)$, denoted by θ_h, and $\theta_e = 1/f_0$ s (i.e., cycle/Hz). The low-quefrency components contain most of the information for speech recognition and are extracted as features [16, 17, 18].

In implementation, the conventional approach of cepstral coefficients extraction involves the following steps: (1) sampling time-domain speech signal frames; (2) computing short-time Fourier transform of each frame; (3) rescaling the frequency axis based on the Mel-scale (in the case of MFCCs); (4) computing the filter-bank energies; (4) transforming to the cepstral domain and (5) performing liftering (i.e., low-pass filtering in the cepstral domain) to obtain the low-quefrency components of the cepstral coefficients. Due to sampling and processing of the high-dimensional raw speech signal and the large number of steps involved in feature extraction, it is highly desirable to seek an alternative when power consumption is a constraint.

2.2.2. *Proposed narrowband feature extraction*

Referring to the description in Section 2.1, we propose a simple feature extraction method that performs dimension reduction directly on the time-domain signal, using a small number of narrowband filters.

Figure 3-(a) plots the logarithm of the PSD of a typical speech frame. The fast fluctuation corresponds to the glottal pulse excitation $\hat{E}(f)$ at the fundamental frequency f_0 and its harmonics, and the envelope (dashed line in Figure 3-(c)) outlines $\hat{H}(f)$, the vocal tract modulation function. The cepstral domain also shows these two components: $\hat{e}(\tau)$ represented by a delta function at θ_e and $\hat{h}(\tau)$ represented by a narrow triangle (Figure 3-(b)). Since the most essential information of $\hat{h}(\tau)$ is concentrated at the low-quefrencies (typically under 2-3 cycle/kHz [17, 19]), the $\hat{h}(\tau)$ component is shown to have a cutoff at θ_h in Figure 3-(b).

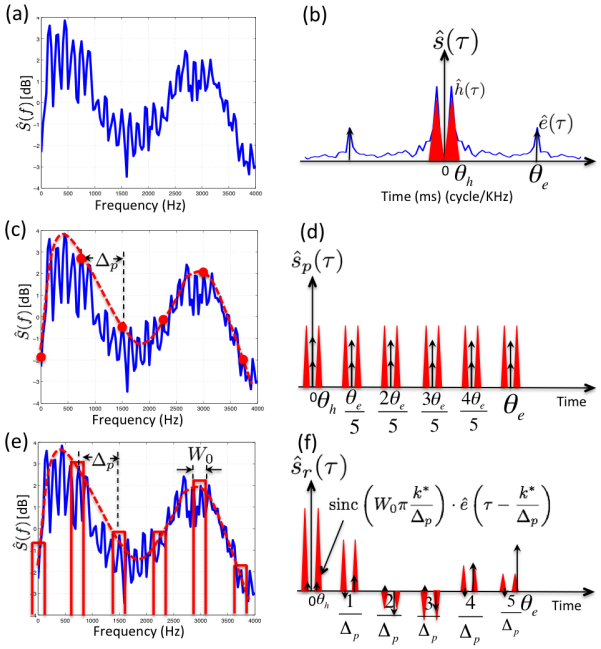

Figure 3: *Narrowband feature extraction: (a) and (b) show the PSD and the cepstrum of a speech segment. The cepstrum is simplified as the summation of $\hat{h}(\tau)$ (triangle shape) and $\hat{e}(\tau)$ (delta function). In (c), $\hat{S}(f)$ is measured at evenly spaced points (denoted by $\hat{S}_p(f)$). Δ_p is an integer multiple of f_0. In (d), $\hat{s}_p(\tau)$ (cepstrum of $\hat{S}_p(f)$) is an aliased version of $\hat{s}(\tau)$. In (e), $\hat{S}(f)$ is measured with evenly spaced rectangular functions with arbitrary spacing, Δ_p. Aliasing between $\hat{h}(\tau)$ and $\hat{e}(\tau)$ occurs in (f) and $\hat{e}(\tau)$ is attenuated with the sinc function.*

The constraint that $\hat{h}(\tau)$ is assumed to be (cepstrally) band-limited to low quefrencies allows the opportunity to 'under-sample the spectral domain signal. Consider the case where we sample $\hat{S}(f)$ at a set of evenly-spaced points (dots in Figure 3(c)). The point sampling function is defined by $P(f)$:

$$P(f) = \sum_{k\in\mathbb{Z}} \delta(f - k\Delta_p), \tag{5}$$

where $\Delta_p = \beta f_0$ is an integer multiple of the fundamental frequency. In the example in Figure 3(c), $\beta = 5$.

The sampled PSD, $\hat{S}_p(f)$, can be expressed as the product of $\hat{S}(f)$ and the sampling function $P(f)$:

$$\hat{S}_p(f) = \hat{S}(f) \times \sum_{k\in\mathbb{Z}} \delta(f - k\Delta_p). \tag{6}$$

The cepstrum of $P(f)$ is another set of delta functions spaced by $1/\Delta_p$. Since multiplication becomes convolution in the cepstral domain, the cepstrum of $\hat{S}_p(f)$, denoted by $\hat{s}_p(\tau)$, is an aliased version of $\hat{s}(\tau)$ (Figure 3(d)):

$$\hat{s}_p(\tau) = \sum_{k\in\mathbb{Z}} (\hat{e}(\tau - \frac{k}{\Delta_p}) + \hat{h}(\tau - \frac{k}{\Delta_p})). \tag{7}$$

As long as we choose $\Delta_p < \frac{1}{2\theta_h}$, repetitions of $\hat{h}(\tau)$ and $\hat{e}(\tau)$ will not overlap. With $\Delta_p = \beta f_0 = \beta/\theta_e$, copies of $\hat{e}(\tau)$ occur

at 0 and multiples of θ_e/β (Figure 3-(d)). Hence, the vocal tract modulation components, $\hat{h}(\tau)$, are not corrupted by aliasing and are preserved in the 'sampled' spectrum $S_p(f)$.

What this implies is that if we have a good estimation of the fundamental frequency, f_0, a few judiciously selected points from the signal PSD can capture most of the essential speech information $\hat{h}(\tau)$. What if the estimation of the fundamental frequency f_0 is not accurate? In this case, $\hat{e}(\tau)$ is not centered around 0 and may be aliased with $\hat{h}(\tau)$. This problem can be mitigated by capturing $\hat{S}(f)$ with rectangular windows instead of a point function. As shown in Figure 3-(e), we measure $\hat{S}(f)$ using a set of evenly spaced rectangular windows (implemented as a set of narrowband filters), which can be expressed as the convolution of the point sampling function $S_p(f)$ and a scaled rectangular function of width W_0:

$$\begin{aligned} G(f) &= P(f) * \text{rect}_{W_0}(f), \\ &= \sum_{k\in\mathbb{Z}} \text{rect}_{W_0}(f - k\Delta_p), \end{aligned} \tag{8}$$

where,

$$\text{rect}_{W_0}(f) = \begin{cases} \frac{1}{W_0}, & \text{if } \frac{-W_0}{2} < f < \frac{W_0}{2} \\ 0, & \text{otherwise.} \end{cases}$$

Since the cepstrum of the rectangular function is a sinc function and convolution in the frequency domain becomes multiplication in the cepstral domain, the cepstrum of $G(f)$ is an impulse train whose amplitudes are scaled by the sinc function:

$$\hat{g}(\tau) = \sum_{k\in\mathbb{Z}} \text{sinc}(W_0\pi\frac{k}{\Delta_p})\delta(\tau - \frac{k}{\Delta_p}). \tag{9}$$

Therefore, the filtered spectrum, $\hat{s}_r(\tau)$, is an aliased version of $\hat{s}(\tau)$ where the amplitudes of the aliased copies are scaled by the amplitude of a sinc function as follows:

$$\begin{aligned} \hat{s}_r &= (\hat{h}(\tau) + \hat{e}(\tau)) * \hat{g}(\tau), \\ &= (\hat{h}(\tau) + \hat{e}(\tau)) * \left(\sum_{k\in\mathbb{Z}} \text{sinc}(W_0\pi\frac{k}{\Delta_p})\delta(\tau - \frac{k}{\Delta_p})\right), \\ &= \sum_{k\in\mathbb{Z}} \text{sinc}(W_0\pi\frac{k}{\Delta_p})\left(\hat{e}(\tau - \frac{k}{\Delta_p}) + \hat{h}(\tau - \frac{k}{\Delta_p})\right). \end{aligned} \tag{10}$$

This is illustrated in Figure 3-(e-f). The modulation function $\hat{h}(\tau)$ is now aliased with $\hat{e}(\tau - k^*/\Delta_p)$, where

$$k^* = \left\lfloor \frac{\theta_e}{1/\Delta_p} \right\rfloor, \tag{11}$$

and the location of aliasing is offset from 0 at $(\theta_e - k^*/\Delta_p)$. When $\Delta_p = \beta f_0$, this offset is equal to 0. As indicated in Figure 3-(f), the amplitude of the aliasing component is scaled by a sinc function:

$$\text{sinc}\left(W_0\pi\frac{k^*}{\Delta_p}\right) \cdot \hat{e}\left(\tau - \frac{k^*}{\Delta_p}\right). \tag{12}$$

As a result, the wider the filter bandwidth W_0, the more attenuation there is on $\hat{e}(\tau - k^*/\Delta_p)$, and hence, the less $\hat{h}(\tau)$ will suffer from aliasing with $\hat{e}(\tau)$. As long as $\Delta_p < \frac{1}{2\theta_h}$ is still satisfied, $\hat{h}(\tau)$ will not be corrupted by its own aliases.

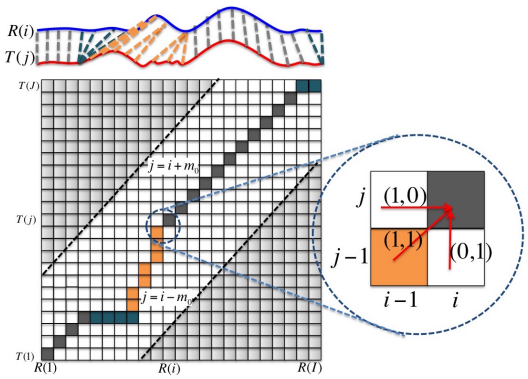

Figure 4: *Illustration of the DTW algorithm. The warping path is represented by the highlighted line. The warping window (window length m_0) is represented by the unshaded area. At each point, there are three candidate movements:* $(1, 0)$, $(1, 1)$ *and* $(0, 1)$.

For example, with filter bank spacing of $\Delta_p = 800\text{Hz} = 0.8\text{kHz}$, filter bandwidth $W_0 = 0.2\text{kHz}$ and speech fundamental frequency $f_0 = 100\text{Hz} = 0.1\text{kHz}$, the low quefrency corruption from the component of $\hat{e}(\tau)$ is $\approx -0.04\hat{e}(\tau - k^*)$.

We have shown that by filtering the signal with a set of narrowband filters, centered around the harmonics of the speech signal and evenly spaced across the frequency spectrum, essential speech information for speech recognition is preserved. The inaccuracy in fundamental frequency estimation can be compensated by increasing the bandwidth of the narrowband filters.

This approach is beneficial because, unlike the conventional approach of using $26 - 40$ filters with varying bandwidths (e.g. in MFCC feature extraction), we only need to use a handful (~ 10) of fixed bandwidth filters. In addition, dimension reduction and feature extraction are done directly on the time-domain signal without transformation to the cepstral domain, which reduces processing complexity. Moreover, if f_0 is estimated well, and the bandpass filters are narrowly centered around the true harmonics where signal energy is concentrated, we have the opportunity to achieve higher in-band SNR than using the general Mel-frequency band filters.

It is important to point out that even though the narrowband features are extracted around the harmonics, which uses the information of f_0, the exact value of f_0 may be lost. For example, if two speakers have very similar vocal tract characteristics $\hat{h}(\tau)$, but one person's fundamental frequency is an exact multiple of the other person's, narrowband features from these two speakers may be indistinguishable.

3. The weighted-DTW algorithm

The back-end operates by comparing features of a trial utterance with features from each of the enrollment samples. In this section, we describe the weighted-DTW algorithm that forms our back-end. We begin with the classical DTW algorithm, then describe our modification.

3.1. Classical DTW

Let the enrollment signal, R, and the input signal, T, each represent a sequence of feature vectors,

$$\begin{aligned} R &= [R(1), R(2), \ldots, R(i), \ldots, R(I)]; \\ T &= [T(1), T(2), \ldots, T(j), \ldots, T(J)]; \end{aligned}$$

where $R(i)$ and $T(j)$ are feature vectors with dimension K, and I and J are the number of temporal frames in R and T, respectively. We would like to measure the similarity between R and T to determine whether T is generated by the target speaker. Due to temporal variations such as speaking speed differences and pauses in the speech utterance (e.g., pauses between words), the similarity between the input features and the enrollment features cannot be directly compared frame-by-frame. There are however standard algorithms such as the DTW algorithm [13, 14, 20], that are designed to mitigate the problem of signal misalignment by applying a warping function coupling two sequences so that they can be directly compared. The warping function, W, can be represented as a sequence of index pairs that provide a mapping between the frames of R and T. More specifically,

$$W = [W(1), W(2), \ldots, W(m), \ldots, W(M)],$$

where $W(m) = (i(m), j(m))$, and i and j are warping indexes corresponding to R and T, respectively.

As shown in Figure 4, the warping function forms a path on the $i-j$ plane and M corresponds to the length of the path. Due to conditions of the DTW algorithm[14], two consecutive points on the warping path can only be connected by three candidate movements:

$$W(m) = \begin{cases} W(m-1) + (0,1), & \text{move up} \\ W(m-1) + (1,1), & \text{diagonal} \\ W(m-1) + (1,0). & \text{move right} \end{cases} \quad (13)$$

The optimal warping path is obtained by first filling up the accumulative distance matrix $D_{I\times J}$ under a chosen distance measure (denoted by 'dist'), and then traversing back the matrix along the entries that yielded the minimum overall distance. More specifically:

$$\begin{aligned} D(i,1) &= \text{dist}(R(i), T(1)), \\ D(1,j) &= \text{dist}(R(1), T(j)), \\ D(i,j) &= \text{dist}(R(i), T(j)) + \min\{D(i-1,j), \\ &\qquad D(i-1,j-1), D(i,j-1)\}. \end{aligned} \quad (14)$$

In order to restrict the total amount of warping and save computation, a warping constraint can be added. A widely used warping constraint is the Sakoe-Chuba window constraint as shown with the unshaded region in Figure 4 (i.e., $|i(m) - j(m)| \leq m_0$). Details of the DTW algorithm can be found in [14]. Subsequent variations of the DTW algorithm have also been developed to add weightings to different frames along the time-axis[14], or to add weightings to different features of each frame [13] .

3.2. Weighted-DTW

For our SV application, the passphrase is defined by the user and could contain long gaps between words. The major challenge associated with using the classical DTW algorithm for our SV application is how to apply sufficient warping to realign the words while still preserving the temporal characteristics of the signal. Existing variants of the DTW algorithm do not address this issue properly. If too much warping is allowed (e.g., m_0 is large), it often results in excessive signal mutation such that details of the signal characteristics are lost, which leads to a large number of false-positive decisions. On the other hand, if the warping constraints are too strict (e.g., m_0 is small), it results in insufficient warping to take care of the long pauses between words and thus results in mis-detections. In order to overcome this issue, we propose a modified version of the DTW algorithm that penalizes excessive warping according to the following factors:

- the penalty scales linearly with the number of consecutive warps of the same type (i.e., 'move up', 'diagonal' or 'move right');
- the penalty scales linearly with the amplitude of the total power envelope
  - more penalty when the signal amplitude is high in order to retain the shape of the signal envelope;
  - less penalty when the signal amplitude is low, which is an indication of possible pauses.

More specifically, the warping function is found as follows: we define a movement matrix M ($M \in \{(1,0),(0,0),(0,1)\}^{I\times J}$) that records the type of movement taken to arrive at each point (i,j). We then define a step counter matrix C ($C \in \mathbb{N}^{I\times J}$) that records the number of accumulative same-type movement to arrive at each point. For example, if we take three consecutive horizontal steps (i.e., $(1,0)$) to arrive at (i,j) , then $C(i,j) = 3$. The counter restarts whenever the previous step and the current step are not the same type. In order to limit mutation to the signal envelope at each step, we use the total energy of the two signals (E_R and E_T) as a weighting function to determine the penalty of taking a certain step:

$$\begin{aligned} D(i,j) =& \text{dist}(R(i), T(j)) + \\ & \min_S\{D((i,j) - S) + P((i,j), S)\}, \end{aligned} \quad (15)$$

where

$$S \in \{(1,0),(1,1),(0,1)\},$$

and

$$\begin{aligned} P((i,j),S) = &\, \mathbb{1}\{M(i-1,j) = S\}C(i-1,j)|\text{E}_T(j)| + \\ &\mathbb{1}\{M(i,j-1) = S\}C(i,j-1)|\text{E}_R(i)|. \end{aligned}$$

Eq.(15) replaces (14) of the conventional DTW algorithm. To save computation, we use the L_1 norm as our distance measure and normalize it over the feature dimension K:

$$\text{dist}(R(i), T(j)) = \frac{1}{K}\sum_{k=1}^{K} |R(i)[k] - T(j)[k]|. \quad (16)$$

The matrices M and C are initialized with

$$M(1,1) = (0,0) \quad \text{and} \quad C(1,1) = 0;$$

and are updated with S^* that yields the minimum $D(i,j)$ (Eq. 15) at each step that

$$\begin{aligned} M(i,j) &= S^*, \\ C(i,j) &= (C((i,j) - S^*) + 1)\mathbb{1}\{M((i,j) - S^*) = S^*\}. \end{aligned}$$

Without the penalty term in (15), the weighted-DTW algorithm would yield the same path as the classical DTW algorithm.

For the classical DTW algorithm, the distance between R and T is equal to $D(I,J)$. That is not the case for the weighted-DTW algorithm due to the additional penalty term. The final

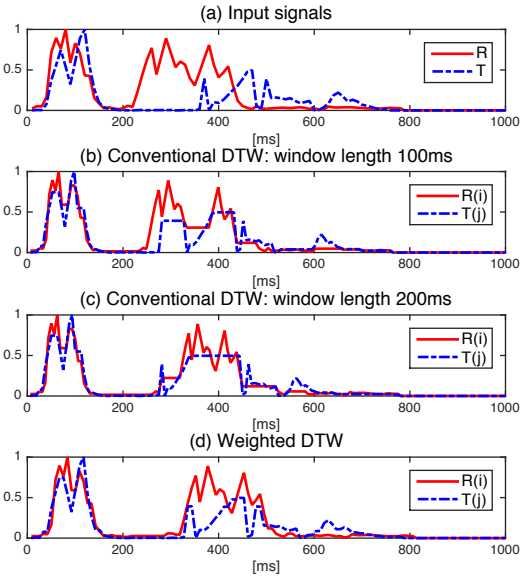

Figure 5: *Warping of two signals R and T. (a) R and T before warping. There is a long pause in the signal T. (b) With window length* 100 *ms, the classical DTW fails to realign the envelopes of the two signals. (c) With window length* 200 *ms, even though the main bulk of the two signals are aligned, the temporal envelope of T is heavily mutated. (d) The weighted-DTW algorithm properly aligns the main bulk of the signals without excessive mutation on the shape of the signal envelopes.*

similarity measure between R and T is re-computed after obtaining the warping path. We normalize the total distance such that the average distance is invariant of the warping path length:

$$D_{\text{norm}} = \frac{1}{M} \sum_{m=1}^{M} \text{dist}(R(i(m)), T(j(m))). \qquad (17)$$

Fig. 5 illustrates the difference between the weighted-DTW algorithm and the conventional DTW algorithm[14]. The simulations demonstrate that the weighted-DTW algorithm is capable of applying sufficiently large amounts of warping in the case of misalignments, while refraining from excessively mutating the signal envelope.

4. Experiment and Results

Using a data set collected at Texas Instruments Kilby Labs, we compare the practical performance of our front- and back-end to baseline systems and under different noise conditions. Collectively, the results demonstrate that equivalent or better performance can be obtained at much lower power with the proposed system compared to conventional systems.

4.1. Proposed and baseline systems

The proposed system comprises a front-end extracting NBSC features (Section 2) and a back-end implementing weighted-DTW (Section 3).

The baseline systems substitute either the front-end or the back-end, or both. For the front-end, the baseline substitutes are conventional features including MFCC and the more primitive Mel-frequency spectral coefficients (MFSCs), which are spectral domain features representing the power in different Mel-frequency scale bands. The MFSCs has demonstrated some success in recent speech recognition research [21]. For the back-end, the baseline substitutes include the conventional DTW algorithm[14] and the GMM-UBM based system [5].

The detailed parameters are as follows. For the front-end:

- NBSC (Proposed features): We use the following parameter settings: B = 6 kHz (cutoff frequency of speech signal), W_0 = 200 Hz (bandwidth of narrowbands) and K= 6, 8, 10, 12. Fundamental frequency f_0 is estimated using the auto-correlation method [22].
- MFCC/MFSC (Baseline features): The MFCC features have 13 dimensions extracted from the 40-dim Mel frequency filterbank. We experiment on two sets of MFSC features: the 26 bands and the 13 bands Mel-frequency filterbank. All features are extracted with frame duration of 25 ms and frame rate of 10 ms.

For the back-end:

- weighted-DTW (Proposed back-end): as described in Section 3.2 with a window length of 250 ms.
- DTW (Baseline back-end): the conventional DTW [14] with the same window length of 250 ms.
- GMM-UBM (Baseline back-end): The GMM-UBM based SV system [5], which requires background model training and assumes prescribed passphrase. We vary the number of Gaussian mixtures and take the parameter that yields the best result.

4.2. Experimental set-up

4.2.1. Data set

The primary dataset consists of audio from three different passphrases (two in English and one in Chinese) spoken by 30 to 40 speakers with $20 - 40$ repetitions per speaker per passphrase (Table 1). The data set was collected in multiple sessions and about $2/3$ of all speakers are male and $1/3$ are female. Each passphrase is limited to a duration of 1s and was sampled at 16 KHz.

Table 1: *Experiment dataset*

Passphrase	# of speakers	# of repetitions
Hi Galaxy	40	40
Ok Glass	40	20
Ok Hua Wei	30	20

A secondary dataset of out-of-vocabulary (OOV) utterances, consisting of 5000 samples (1s duration) of short commands, speech clips from conversations and audio books, is also used. Finally, noisy samples are generated by adding wind noise or car noise to each clean sample such that the total SNR within the 1s speech segment is equal to 3dB.

4.2.2. Evaluation and decision threshold

Given a passphrase, every speaker is chosen as the target speaker once. We take 3 utterances from this speaker as enrollment samples and the rest are used as positive (authentic) test samples. The same passphrase from all other users are used as negative (impostor) samples for SV evaluation, while all samples of the OOV dataset are used as negative samples during

false-trigger evaluation. For experiments involving noisy samples, the enrollment samples are clean.

The minimum of the distances between a test sample and the enrollment samples is compared with a threshold to make the final verification decision. The threshold is chosen *a posteriori* such that the false-positive and false-negative rates are equal (unless otherwise indicated), which gives the equal-error rate (EER).

For the GMM-UBM model training, the speakers are divided into two halves. The first half is used for background model training and the other half for evaluation. Each user from the evaluation set is chosen as the target speaker once and 4 utterances are used as enrollment samples for speaker specific model adaptation.

4.2.3. Feature adaptation under background noise

When noise is present, we assume a coarse estimate of the noise spectrum is known. The energy of the wind and car noises is concentrated in the low-frequency domain under 2kHz. Therefore, we simple discard spectral features below 2kHz and use the remaining features for NBSC and MFSC front-ends; for MFCC this is not feasible, so there is no feature adaptation.

4.3. Experiment results

4.3.1. Front-end and back-end combinations

The first set of experiments compare NBSC (proposed) and MFCC front-ends combined with weighted-DTW (proposed), DTW, or GMM-UBM back-ends, for both noiseless and noisy conditions.

Table 2: *EER [%] for combinations of features and algorithms.*

	Clean		Noisy (3dB)	
Features / Algorithm	MFCC	NBSC	MFCC	NBSC
weighted-DTW	0.9	1.1	10.5	5.7
DTW	1.4	1.5	13	6.7
GMM-UBM	2.6	N/A	6.8	N/A

Table 2 shows the EER for systems with different feature and verification algorithms. Without background noise, all of the 12 bands are used as features. With background noise, only the bands above 2kHz are active. The weighted-DTW algorithm yields better accuracy than the standard DTW algorithm for both the MFCC and the NBSC features. Without background noise, the 12-band NBSC yields slightly worse accuracy than the MFCC features. Under 3dB SNR, the NBSC yields much better performance than the MFCC features as a result of spectral domain feature selection. Without the need for background model training, the proposed system outperforms the GMM-UBM based system in both clean and noisy conditions. Note that the GMM-UBM requires background model training with a large number of training samples (usually generated from a pool of speakers). Since the NBSCs are extracted based on the fundamental frequency of the target speaker, different sets of features are used for different speakers. Hence, the NBSCs are generally not applicable to algorithms that require model training with a large number of training samples, such as the GMM-UBM.

In contrast to Table 2, which evaluates the systems' SV accuracies (same passphrase produced by different speakers), Table 3 shows the systems' false-positive rates against the OOV data set (to evaluate the false-triggering rate as a wake-up application). The back-end is fixed to the weighted-DTW algorithm and the decision threshold is the same as that yielded the EER in Table 2. Without background noise, the false-trigger rate is 0 for both the MFCC and NBSC features. Under 3dB SNR, the NBSC yields a false-trigger rate of 0.6%, much lower than the MFCC features with a false-trigger rate of 1.4%.

Table 3: *False-positive rates [%] with OOV dataset. Decision threshold is taken from the EER threshold obtained with the weighted-DTW algorithm in Table 2.*

	Clean		Noisy (3dB)	
Features / Algorithm	MFCC	NBSC	MFCC	NBSC
weighted-DTW	0	0	1.4	0.6

4.3.2. Spectral domain features: NBSC vs. MFSC

The second set of experiments fixes a weighted-DTW (proposed) back-end and compares accuracies of NBSC (proposed) vs. MFSC front-ends at various filter-band counts.

Table 4 shows that, in general, accuracy improves as the number of bands increases. With fewer bands than what is commonly used in MFSC-based front-ends, the NBSC performance is better than that of the MFSC. When there is background noise, the accuracy improves significantly by using fewer features (i.e., band selection).

Table 4: *EER [%] for NBSC and MFSC features with the weighted-DTW algorithm, under quiet condition and 3dB wind and car noise.*

Features	NBSC				MFSC	
# of filters	6	8	10	12	13	26
Clean	1.99	1.9	1.54	1.1	1.95	1.83
Noisy (band selection)	6.8	6.6	6.3	5.7	16.4	17.2
Noisy (all bands)	15.5	15	15	14.5	33.4	33.9

4.4. Power consumption in hardware

The total system power consumption is evaluated as the sum of the front-end and back-end power consumption. The front-end power consumption is estimated from Texas Instruments's chip design, which consists of a voice-activity-detector (VAD) with power consumption of 150 μW and a filterbank with an additional power cost of 10 μW per band. The back-end algorithm is implemented on the Cortex-M0 processor. The firmware implementation for the algorithm and data occupies less than 40kB memory. Recall the computation complexity of the back-end algorithm is proportional to the feature dimension (i.e., the number of bands) and the total number of frames in the utterance. Given a fixed frame rate of 10ms, the back-end power consumption is proportional to the number of bands and it is slightly under 9 μW per band. With 12 active bands, the end-to-end system power consumption is kept under 380 μW assuming 100% duty cycle.

5. Conclusions and future work

In this paper, we proposed a low-power system for text-dependent SV allowing the enrollment of *a priori* unknown passphrases. The front-end consists of a set of narrow-band filters that are centered around the harmonics of the fundamental frequency f_0 and evenly spaced across the frequency spectrum. We show through analysis that essential speech information is retained by capturing information within a few narrowbands. Unlike the MFCC features, which require a large number of filters and high-rate processing, this method offers the benefits of simple and low-power implementation. The back-end is an improved weighted-DTW algorithm. It penalizes signal mutation at where the signal amplitude is high while allowing the remaining parts to align. Compared to conventional DTW, the amount of signal envelope mutation is reduced.

The proposed system delivers improved performance over the baseline while consuming less power ($< 380\mu$W). In quiet conditions, the proposed system achieves comparable performance as the MFCC+DTW and GMM-UBM systems. The highlight however is in the performance under noisy conditions, where the proposed system has much improved accuracy. The gain in accuracy is due to the effects of spectral domain feature selection based on the noise spectrum.

As a next step, more advanced feature selection algorithms can be developed for general background noises.

6. Acknowledgment

The authors would like to thank Kilby Labs at Texas Instruments for supporting this research work. This work was supported in part by Texas Instruments, and by NSF under Grant No. CCF-1319828.

7. References

[1] Douglas A Reynolds and William M Campbell, "Text-independent speaker recognition," in *Springer Handbook of Speech Processing*, pp. 763–782. Springer, 2008.

[2] Tomi Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[3] Matthieu Hébert, "Text-dependent speaker recognition," in *Springer handbook of speech processing*, pp. 743–762. Springer, 2008.

[4] Douglas E Sturim, Douglas A Reynolds, Robert B Dunn, and Thomas F Quatieri, "Speaker verification using text-constrained gaussian mixture models," in *IEEE Int Conf on Acoustics, Speech, Signal Processing*, Orlando, USA, May 2002, pp. 677–680.

[5] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[6] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans on Acoustics, Speech, Signal Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[7] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Jorge Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE Int Conf on Acoustics, Speech, Signal Processing*, Florence, 2014, pp. 4052–4056.

[8] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[9] Tomoko Matsui and Sadaoki Furui, "Concatenated phoneme models for text-variable speaker recognition," in *IEEE Int Conf on Acoustics, Speech, Signal Processing*, Minneapolis, USA, 1993, pp. 391–394.

[10] Aaron E Rosenberg, Chin-Hui Lee, and Frank K Soong, "Sub-word unit talker verification using hidden markov models," in *IEEE Int Conf on Acoustics, Speech, Signal Processing*, Albuquerque, USA, April 1990, pp. 269–272.

[11] Steven B Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans on Acoustics, Speech, Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[12] Ben Gold, Nelson Morgan, and Dan Ellis, *Speech and audio signal processing: processing and perception of speech and music*, John Wiley & Sons, 2011.

[13] Sadaoki Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans on Acoustics, Speech, Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[14] Hiroaki Sakoe and Seibi Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans on Acoustics, Speech, Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[15] Fumitada Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans on Acoustics, Speech, Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[16] Alan V Oppenheim and Ronald W Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 95–106, 2004.

[17] Ronald W Schafer, "Homomorphic systems and cepstrum analysis of speech," in *Springer Handbook of Speech Processing*, pp. 161–180. Springer, 2008.

[18] Lawrence R Rabiner and Ronald W Schafer, *Theory and application of digital speech processing*, Prentice hall, 2009.

[19] Taffeta M Elliott and Frédéric E Theunissen, "The modulation transfer function for speech intelligibility," *PLoS comput biol*, vol. 5, no. 3, 2009.

[20] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, Prentice hall, 1993.

[21] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoffrey Zweig, Xiaodong He, Julia Williams, et al., "Recent advances in deep learning for speech research at microsoft," in *IEEE Int Conf on Acoustics, Speech, Signal Processing*, Vancouver, CA, 2013.

[22] Lawrence Rabiner, Michel J Cheng, Aaron E Rosenberg, and Carol A McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Trans on Acoustics, Speech and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.