Title
Robust Speech and Bird Song Processing using Multi-band Correlograms and Sparse Representations

Author
Tan, Lee Ngee

Publication Date
2014

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Robust Speech and Bird Song Processing using Multi-band Correlograms and Sparse Representations

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical Engineering

by

Lee Ngee Tan

2014

Abstract of the Dissertation

Robust Speech and Bird Song Processing using Multi-band Correlograms and Sparse Representations

by

Lee Ngee Tan

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2014

Professor Abeer Alwan, Chair

This dissertation focuses on algorithms for robust speech and bird song processing. Many applications perform well under ideal signal conditions, e.g. noise-free, full bandwidth, sufficient training data. However, a large degradation in performance is generally observed when the input signal condition deviates from these ideal conditions. This dissertation describes robust algorithms for three applications, namely human-pitch detection, automatic speech recognition, and birdsong phrase classification. In the first application, a noise-robust, multi-band summary correlogram (MBSC)-based pitch detector is proposed. Novel signal processing schemes, which include comb-filter channel selection and subband reliability weighting, are designed to enhance the MBSC's peak at the most likely pitch period.

In the second application, a feature enhancement scheme using jointly-sparse reference and estimated soft-mask representations, is developed for noise-robust automatic speech recognition (ASR). Reference and estimated soft-mask exemplar-pairs are extracted from clean and noisy utterance-pairs in the training data.

Using a sparsity-based dictionary learning algorithm, dictionary representations are trained from the exemplar-pairs. The sparse linear combination of estimated soft-mask dictionary representations that best approximates the test utterance's estimated soft-mask is applied to the reference soft-mask dictionary to produce an enhanced soft-mask. This enhanced soft-mask is then used to perform noise suppression on the spectrogram from which features for ASR are extracted.

In the third application, a simple exemplar-based sparse representation (SR) classifier is evaluated on limited data for birdsong phrase classification and verification. Song recordings of the Cassin's Vireo are used for performance evaluation. This study of the SR classifier for bird phrase classification is inspired by a paper that proposed the SR classifier for face recognition and outlier face detection, and reported good performance with only 7 training images per subject. Algorithmic enhancements are subsequently added to the original SR classification framework to improve the classification accuracy of automatically detected and segmented phrases, and phrases sang by bird individuals that are not found in the training set. These algorithmic enhancements include dynamic time warping (DTW) and frame-based feature normalization prior to SR classification. When the class decisions from DTW and first pass SR classification are different, SR classification is repeated with frequency-bin-normalized spectrographic features to resolve the two conflicting decisions.

The dissertation of Lee Ngee Tan is approved.

Charles Taylor

Alan Laub

Kung Yao

Abeer Alwan, Committee Chair

University of California, Los Angeles

2014

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my gratitude to Prof. Abeer Alwan for the guidance and encouragement that she has provided throughout my academic endeavor at UCLA. I greatly appreciate her insightful suggestions and the freedom she has granted me in my research. Besides being a responsible academic advisor, Prof. Alwan is also a caring mentor who understands that there could be dry periods of unfruitful research. She did not burden me with unnecessary pressure when I was struggling to produce good research, while juggling motherhood.

I would like to thank Prof. Kung Yao for his kind appraisal, which together with Prof. Alwan's recommendation, had helped me obtain a UCLA Graduate Division's Dissertation Year Fellowship. I would also like to thank Prof. Charles Taylor for sharing his knowledge in bird songs, and his encouraging comments on my research on bird phrase classification. I would also like to express my appreciation to Prof. Alan Laub for his time and effort to serve in my Ph.D. committee.

I have enjoyed the interaction with current members and former members of the UCLA Speech Processing and Auditory Perception Laboratory (SPAPL), especially Harish Arsikere, Jonas Borgström, Gang Chen, Juan Cortes, Kantapon (Jom) Kaewtip, Eleanor Lee, Hitesh Gupta, Gary Leung, Anirudh Raju, Julien van Hout, and Jie Zhao. I thank them for their companionship and the interesting exchange of ideas.

I like to express my appreciation to the SCENIC team members at SRI for their collaboration in the Defense Advanced Research Projects Agency (DARPA) Robust Automatic Transcription of Speech (RATS) project. I would also like to thank Richard Hedley, George Kossan, and Ni-Chun Wang for their assistance in

the birdsong phrase classification research.

My UCLA M.S and Ph.D. degrees has been mainly funded by a postgraduate scholarship awarded by DSO National Laboratories (DSO) in Singapore. Without this scholarship, I would not have the opportunity to pursue my studies in a prestigious U.S. university, such as UCLA. Thus, I am grateful for the support from my supervisors and the management committee at DSO. My research and conference travels are also funded in part by the DARPA via SRI under Contract No. D10PC20024, and the National Science Foundation (NSF) Award No. 0410438 and IIS-1125423.

Last but not least, I would like to express my deepest thanks to my husband, my parents, and family members for their constant love, support, and patience, during the long years of studies away from home. I am thankful for my husband, Shang Kee, who tolerated my frustrations and journeyed with me as we pursue our Ph.D. together. I am grateful to my mother and parents-in-law who traveled to the United States to take care of my daughter, husband and I. I also want to thank God for providing me with the spiritual strength to persevere, and the awareness of my tendency to take the people who care about me for granted.

Vita

2002	B.E. (Electrical Engineering), National University of Singapore (NUS), Singapore
2002–2007	Member of Technical Staff, DSO National Laboratories, Singapore
2007–2008	Senior Member of Technical Staff, DSO National Laboratories, Singapore
2008–2010	M.S. (Electrical Engineering), University of California, Los Angeles (UCLA)
2010–2013	Graduate Student Researcher, Department of Electrical Engineering, UCLA
2013	Graduate Division Dissertation Year Fellowship Recipient, UCLA

Publications

L. N. Tan and A. Alwan, "Noise-robust F0 estimation using SNR-weighted summary correlograms from multi-band comb filters," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 4464-4467.

L. N. Tan, K. Kaewtip, M. L. Cody, C. E. Taylor, and A. Alwan, "Evaluation of a sparse representation-based classifier for bird phrase classification under limited

data conditions," in *Proc. Interspeech*, 2012, pp. 2522-2525.

L. N. Tan, and A. Alwan, "Multi-band summary correlogram-based pitch detection for noisy speech", *Speech Communication*, Vol. 55, 2013, pp. 841-856.

L. N. Tan, G. Kossan, M. L. Cody, C. E. Taylor, A. Alwan, "A sparse representation-based classifier for in-set bird phrase verification and classification with limited training data," in *Proc. ICASSP*, 2013, pp. 763-767.

K. Kaewtip, L. N. Tan, A. Alwan, C. E.Taylor, "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification," in *Proc. ICASSP*, 2013, pp. 768-772.

K. Kaewtip, L. N. Tan, A. Alwan, "A pitch-based spectral enhancement technique for robust speech processing," in *Proc. Interspeech*, 2013, pp. 3284-3288.

M. Graciarena, A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. H. L. Hansen, A. Janin, B.-S. Lee, Y. Lei, V. Mitra, N. Morgan, S. O. Sadjadi, T. Tsai, N. Scheffer, L. N. Tan, and B. Williams, "All for one: Feature combination for highly channel-degraded speech activity detection," in *Proc. Interspeech*, 2013, pp. 709-713.

L. N. Tan and A. Alwan, "Feature enhancement using sparse reference and estimated soft-mask exemplar-pairs for noisy speech recognition," in *Proc. ICASSP*, 2014, pp. 1729-1733.

CHAPTER 1

Introduction

1.1 Motivation

Since the advent of digital speech processing, many speech applications have been developed. These include speech coding, speech enhancement, speech synthesis, speaker identification, automatic speech recognition (ASR), etc. As the processing power of computers increases, more sophisticated digital speech processing techniques are made possible. Some examples of popular applications are speech synthesis in GPS navigation devices, audio search by humming (query by humming), and speech recognition in many automated telephone surveys and directory services. Most speech applications give reasonably good performance when the input speech is free from distortions or closely matched with the data used in model training. However, the performance of these applications suffers a large degradation when distortions such as environmental noise and transmission channel frequency characteristics are present in the input speech signals. Besides signal distortions, the limitation of data available to statistically train model parameters could also affect performance.

Similarly, the field of bird song detection and classification requires robust signal processing techniques to deal with such signal distortions and limited data conditions. Classifying particular bird calls or song elements becomes especially challenging in those cases where the song repertoire is diverse, comprising large

numbers of variant syllables or phrases. For example, in some species thousands of distinct phrases have been observed in their lexicons [1]. In our experience, the frequencies at which individual bird song elements are observed often resembles a "Zipf curve", where a few phrases are heard many times, but the majority of phrases are rare. Communication in other species, including humans, typically follows this same relation [2]. The amount of training data available can also be limited due to the opportunistic nature of bird vocalization collection in geographical locations of interest. A premium is thus placed on the ability of automated classifiers to correctly classify birdsong phrases based on limited training data. The ability of an automated classifier to detect new phrase types that are not observed in the training set is also important. Accurate phrase verification or outlier detection could potentially reduce the resources for manual annotation on newly collected data, since the human annotators can focus on the detected "outliers" to identify and label potential new phrase classes.

This dissertation focuses on three particular applications, namely noise-robust pitch detection, noise-robust automatic speech recognition, and birdsong phrase classification with limited training data.

1.2 Noise-robust pitch estimation and voiced/unvoiced detection

A speech utterance generally contains both voiced and unvoiced units, known as phonemes. Voiced sounds typically include vocal cord vibration at a rate called the fundamental frequency (F0). In Fig. 1.1, the time- and frequency-domain representations of a voiced and an unvoiced speech frame are plotted. The time waveform of voiced speech frame has a periodic structure with a fundamental

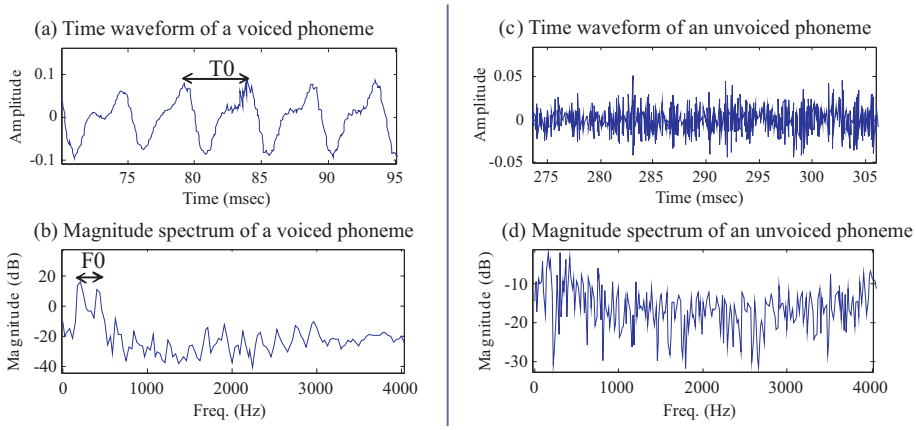

Figure 1.1: Time- and frequency domain representations of a voiced speech frame and an unvoiced speech frame.

period of $\mathrm{T0} = \frac{1}{\mathrm{F0}}$. Its spectrum has a harmonic structure, with spectral harmonic peaks spaced F0 apart, and located near integer multiples of F0. In contrast, such a periodic or harmonic structure is absent in an unvoiced speech frame.

1.2.1 Pitch estimation

Fundamental frequency (F0) detection is important for many speech applications. These applications include speech enhancement, synthesis, coding, source separation, and auditory scene analysis. A pitch detector performs both pitch estimation and voiced/unvoiced (V/UV) detection. In pitch estimation, the rate of vocal-fold vibration is estimated, while in V/UV detection, voiced or quasi-periodic speech frames are distinguished from the rest of the signal. In general, pitch estimation can be performed using (1) time-domain, (2) frequency-domain, or (3) time-frequency-domain signal processing techniques. Time-domain pitch estimation exploits the signal's temporal periodicity by computing a temporal correlation or difference function directly from the signal samples. Some well-known examples of time-domain pitch estimation algorithms are RAPT [3], YIN [4], and

the average magnitude difference function (AMDF) pitch extractor [5], which are known to give accurate pitch estimates for clean speech. Frequency-domain pitch estimation relies on the presence of strong harmonic peaks near integer multiples of F0 in the short-time spectral representation. Some examples of such frequency-domain pitch estimation algorithms are subharmonic-to-harmonic ratio (SHR) [6], dominant harmonics [7], and SWIPE' (SWIPE' is a variant of SWIPE that focuses on harmonics at prime integer multiples of F0) [8]. In time-frequency-domain pitch estimation algorithms, the input signal is typically decomposed into multiple frequency subbands, and time-domain techniques are applied on each subband signal. A popular time-frequency-domain technique is the auditory-model correlogram-based algorithm inspired by Licklider's duplex theory of pitch perception [9], in which frequency decomposition is performed using an auditory filterbank (for which Gammatone filterbanks [10] are widely used), followed by autocorrelation (ACR) computation on each subband signal. The correlogram is formed by vertically stacking all ACR functions to form a 2-D image [11]. Finally, the fundamental period (T0) of the signal is found by locating the ACR delay lag of the maximum peak in the "summary" correlogram (SC), which is typically the averaged ACR function. ACR can be applied either directly on the subband signal or its envelope. The latter is usually performed on mid- and high-frequency subbands only [12,13]. These subbands have sufficiently wide bandwidths to capture at least two consecutive harmonic peaks, such that the resulting filtered signals have an amplitude modulation frequency equal to F0 (a.k.a. beat frequency) [14]. It has been shown that correlogram-based techniques can yield estimates close to human's perceived pitch for difficult signals with missing fundamental, inharmonic complexes and noise tones [15, 16]. Being a multi-band approach, correlogram-based techniques also tend to be more noise-robust than time-domain or frequency-domain algorithms whose parameters are

fixed regardless of the signal's periodicity in the different subbands. This is because additional subband selection or weighting schemes, such as those in [12] and WWB [13], can be implemented to give less emphasis to the noise-dominated subbands. Since the filters in a Gammatone filterbank are narrower and spaced more closely at lower linear frequencies than at higher frequencies [10], the number of filters at lower frequencies (within the first 1 kHz) can be almost equivalent to the number filters in the mid and high frequencies. When the majority of harmonics at the low frequencies are attenuated due to the transmission channel characteristics or masked by strong low-frequency noise interference, it is challenging to design an effective subband selection and weighting scheme to select reliable subband ACRs such that the maximum peak of the resulting summary correlogram yields the true pitch value.

1.2.2 Voiced/unvoiced detection

Voiced/unvoiced (V/UV) detection can be performed by either utilizing the information derived from the pitch estimation module, or using a separate module that is independent of the pitch estimation algorithm. The simplest V/UV detector is one that applies a constant decision threshold on a single degree-of-voicing feature computed by the pitch estimation module, e.g., ACR or cepstral peak amplitudes [17]. To further improve detection accuracy, the initial V/UV decisions are usually smoothed via median filtering [18, 19]. A disadvantage of the constant-threshold scheme is that since the degree-of-voicing feature tends to be dependent on the signal-to-noise ratio (SNR), a threshold level tuned for a particular SNR, generally does not work well at different SNRs. Thus, threshold adaptation techniques have been proposed to improve the noise-robustness of V/UV detectors. Typically, the threshold is adapted based on long-term statis-

tics (min, max, mean, median, etc.) of degree-of-voicing-related features [19,20]. In this case, V/UV detection performance would tend to degrade under a highly non-stationary noise condition. Dynamic programming – a tracking algorithm that integrates V/UV detection with pitch estimation, is another common technique used for pitch detection [3,21,22]. A dynamic programming algorithm finds the least-cost path based on some pre-defined voicing and frequency transition cost functions, leading to performance improvements in both V/UV detection and pitch estimation through utilizing voicing and pitch information from multiple frames. However, when a constant value is used to control the voicing transition cost, such as the voice bias in [3], the V/UV detection performance of these pitch detectors is also dependent on SNRs. Since it is generally difficult to perform noise-robust V/UV detection based on the single degree-of-voicing feature from pitch estimation [23], statistically-trained V/UV classifiers have also been proposed, especially for applications that do not require pitch estimates to be computed (e.g., speaker-independent speech recognition). This latter class of V/UV detectors, which can operate independently from pitch estimation, have reported robust V/UV detection performance, especially if their parameters are learned from noisy speech [24,25]. Since pitch estimation is already part of a pitch detector, the former class of V/UV detectors is of interest in this chapter. There are also algorithms that perform pitch-tracking using models trained on information extracted during pitch estimation. For example, hidden Markov models (HMMs) are used in WWB [13] to form continuous single or dual pitch contours for noisy speech. These data-driven algorithms yield robust voicing/pitch detection performance when the test data has characteristics that are similar to the data used for training the models, but their performance degrades in mismatch train/test data conditions.

1.3 Signal processing using sparse representations

This section reviews the framework of using sparse representations for signal representation and reconstruction, which the proposed algorithms for ASR and birdsong phrase classification are based on. Most of these materials is extracted from [26, 27].

Many natural signals have a sparse or concise representation when expressed in a proper basis or dictionary $\mathbf{A} = [a_1\, a_2\, ...\, a_n] \in \mathbb{R}^{n\times n}$, with a_i as columns. This means that a column vector $b \in \mathbb{R}^n$ can be expressed using a small number of a_i's, as shown in Eq (1.1), i.e. a large fraction of the coefficients in vector x will be small.

$$b = \mathbf{A}x = \sum_{i=1}^{n} x_i\, a_i \tag{1.1}$$

One example of b and $\mathbf{A}$ are an n-pixel image and an orthonormal wavelet basis, respectively. The small coefficients in x can be discarded or set to zero, so as to reconstruct a compressed image $b_S = \mathbf{A}x_S$, where x_S is a sparse vector with all but the largest S coefficients set to zero. This is used in the JPEG-2000 [28] image data compression scheme to generate an compressed image that has a small perceptual difference from the original image.

In the situation where only a subset of the coefficients of b is observable, due to constraints in the data collection equipment or to data loss, we have

$$b_k = \mathbf{A}_k\, x, \;\; k < n \tag{1.2}$$

Eq. (1.2) can be obtained from Eq. (1.1) by multiplying both sides by a sampling matrix, $\Phi \in \mathbb{R}^{k\times n}$ for extracting specific row coordinates, i.e $b_k = \Phi\, b \in \mathbb{R}^k$ is a sub-vector of b that contains the subset of k observable coefficients in b, and $\mathbf{A}_k = \Phi\mathbf{A} \in \mathbb{R}^{k\times n}$ is a submatrix of $\mathbf{A}$ containing the k rows of $\mathbf{A}$ that corresponds to the k coefficients in b. To obtain a sparse x in this under-determined linear

system of equations, an l_1 ($\|x\|_1 \triangleq \sum_i |x_i|$) minimization problem in Eq. (1.3) can be formulated, which can be efficiently solved via convex optimization. Greedy algorithms [29] have also been proposed to recover a sparse solution.

$$\min_{x \in \mathbb{R}^n} \|x\|_1 \text{ subject to } b_k = \mathbf{A}_k \, x \tag{1.3}$$

Besides using a sub-sampling matrix for Φ to obtain an under-determined linear system of equations such that the l_1-minimization problem is meaningful, an under-determined linear system of equations can also be formed using other matrices for $\Phi \in \mathbb{R}^{(k<n)\times m}$, such as a random matrix, to reduce the number of linear equations. On the other hand, if the matrix $\mathbf{A}$ is an over-complete dictionary, or it has a fewer number of rows than columns to begin with, i.e $\mathbf{A} \in \mathbb{R}^{(m \ll n)\times n}$, then dimension reduction may not be necessary to obtain a sparse solution. An example of such an $\mathbf{A}$ matrix is one that is learned from exemplars. Instead of utilizing conventional orthonormal bases such as the wavelet, discrete cosine transform (DCT), or Haar basis, the dictionary is derived from feature vector examples in the training set. Algorithms using exemplar-based dictionaries or dictionaries learned from exemplars have shown superior performance to traditional general-purpose bases in applications ranging from audio/image/video enhancements, source separation and classification in speech and object recognition, and bioinformatic data decoding [27]. In this dissertation, sparse representations obtained or learned from exemplars are utilized in the two applications described in the following subsections, namely (1) feature enhancement for ASR, and (2) birdsong phrase classification.

1.3.1 Noise-robust automatic speech recognition

Automatic speech recognition (ASR) converts speech to text, which facilitates human machine interactions, leading to applications such as automatic call pro-

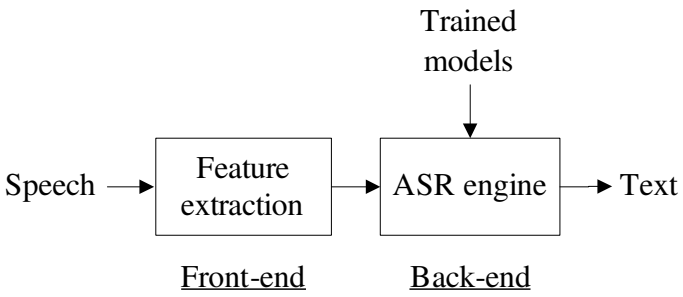

Figure 1.2: A general ASR framework

cessing and query-based information systems. ASR applications are also increasingly popular in portable devices such as smart phones, tablets, and navigation systems, which makes noise-robust speech recognition ever more important.

Generally, ASR involves front-end processing and back-end processing, as shown in Fig. 1.2. During front-end processing, features are extracted from the speech signal. During back-end processing, an ASR engine decodes the most likely sequence of words spoken by matching these features against pre-trained acoustic and language models. Both front-end and back-end processing techniques have been proposed to improve the noise robustness of ASR. Front-end processing includes feature enhancement techniques, such as noise-robust feature extraction and spectral enhancement. Noise-robust features are reliable features that are more invariant in noise. These include speech salient features such as normalized modulation cepstral coefficients (NMCC) [30]) and multi-scale spectro-temporal features [31]. These features are extracted by focusing on speech formants, which are high in energy, and more resilient to noise. Noise-invariant features can also be obtained by reducing the differences between features extracted from clean speech and noisy speech. One example is the peak-isolation method [32] that reduces the variability in spectral energies at the valleys by performing inverse discrete cosine transform (IDCT) on liftered Mel-frequency cepstral co-

efficients (MFCCs), followed by a half-wave rectification on the resulting Mel-frequency spectrum. Noise-invariant cepstral-based features can also be obtained using cepstral mean subtraction, variance normalization, and auto-regressive-mean-averaging (ARMA) filtering [33]. Another feature enhancement technique is the Stereo Piecewise Linear Compensation for Environments (SPLICE) [34]. It learns a joint probability distribution of noisy and clean cepstra from clean and noisy training utterance-pairs (stereo training data), and uses this distribution to estimate cepstra of clean speech from noisy cepstra. In regards to feature enhancement techniques that perform spectral enhancement prior to feature extraction, they improve the noise robustness of ASR features that perform well in a quiet environment (e.g. Mel-frequency cepstral coefficients, MFCCs). Spectral enhancement can be perform via noise subtraction [35] or noise suppression using a statistical model-based spectral amplitude estimator [36], Wiener filtering [37], or a ratio/soft-mask [38,39]. A soft-mask is a time-frequency representation that can take any real values between 0 and 1. An accurate soft-mask would have a value closed to 1 for a speech-dominated time-frequency region, and a value closed to 0 for a noise-dominated region, such that noise is suppressed when the soft-mask is multiplied to the noisy speech spectra. Back-end processing involves adapting or transforming the trained models [40, 41], and modifying the decoding framework in the ASR engine. In model adaptation, model parameters are adapted based on a data subset with similar noise characteristics as the test data, such that the models are more attune in recognizing features extracted from similarly corrupted speech. In [42], acoustic modeling in the ASR engine is improved by using a deep neural network to compute the posterior probabilities over hidden Markov model (HMM) states, instead of using Gaussian mixture models (GMMs). In this dissertation, front-end feature enhancement using spectral enhancement and stereo training data is the research area of interest.

With the successful application of compressive sensing / sparsity-based techniques for image processing, such as robust face recognition [43] and image denoising [44], techniques exploiting sparsity in speech for feature enhancement [45–48] have also been proposed. In [45] and [46], unreliable Mel-spectral components of noisy speech are estimated/imputed using a sparse linear combination of spectral representations in the dictionary. This sparse linear combination is computed from the reliable Mel-spectral components in the noisy speech utterance. The discrete Haar transform basis forms the dictionary used in [45]; while log Mel-spectral exemplars (spanning several time frames) extracted from clean training data form the dictionary used in [46]. Their ASR performances are very sensitive to the accuracy of the binary mask that determines the reliable spectral components. The dependency on the binary mask is avoided in the feature enhancement technique proposed in [47], which uses all Mel-spectral components (both reliable and unreliable), to derive a soft-mask from the sparse linear combination of clean speech and pure noise Mel-spectral dictionary exemplars that best approximates the test Mel-spectra. This soft-mask is used to denoise the test Mel-spectra prior to cepstral feature extraction. This feature enhancement technique achieves higher word recognition accuracies than the imputation method on the Aurora-2 noisy digit recognition task [49]. Including noise exemplars extracted from the test utterance helps to improve recognition accuracy on unseen noise types (not found in the dictionary) [50]. Another sparsity-based feature enhancement scheme is proposed in [48], which uses a joint dictionary containing clean and noisy log-Mel-spectral exemplar-pairs extracted from clean and noisy utterance-pairs in the training set. The sparse linear combination of the noisy log-Mel dictionary exemplars that best approximates the test log-Mel spectra is computed. This sparse activation weighting vector is applied to the corresponding clean log-Mel dictionary exemplars to reconstruct an enhanced log-Mel spectra.

This algorithm outperforms the feature enhancement scheme in [47] on a small vocabulary, in-car noisy speech recognition task [51].

1.3.2 Birdsong phrase classification using limited training data

Bird vocalizations are compositions of short individual units or syllables, each generally lasting less than a second. Typically, longer and more elaborate songs are used for mate attraction and territory declaration; shorter calls are often used for family member contacts and identification, predator announcement or food information communication [1]. Sound recordings of bird vocalizations are helpful in behavioral and population studies [52, 53], especially in dense vegetation environment, where visual identification is difficult. Automated birdsong classification has already proved useful for species identification [54–57], individual recognition [58], and classification of particular syllables or phrases expressed by birds with complex song structures [59–61]. Much of this research has been reviewed by [52] and [53]. Such applications will gain importance with an increasing general interest in "soundscape ecology" [62], and a more refined understanding of bird communication.

The classification of birdsong elements is similar to automatic speech recognition (ASR) or keyword-spotting for speech. Thus, some algorithms that have shown good results for ASR or keyword-spotting have been applied to bird syllable or phrase classification. One of them is the dynamic time warping (DTW) template-matching technique, which was extensively explored in the 1970's for word recognition. The DTW algorithm compares an input sequence with a set of predefined template sequences for each class, by finding an optimal alignment between two sequences through minimizing the cost function. Such DTW template-matching technique has been proposed for segmented bird call classifi-

cation [63], as well as automatic recognition of birdsong syllables from continuous recordings [64]. High classification accuracies were reported when there is little within-class variability. Besides DTW, hidden Markov models (HMMs)-based recognizers [65] which has been predominant for ASR systems for the past two decades, have also been proposed for bird song or song elements recognition. In [59], the advantages and limitations of DTW and HMMs were evaluated. It was found that good performance of the DTW-based technique requires careful selection of templates that may demand expert knowledge, especially in the presence of noise or confusing short-duration calls. On the other hand, equivalent or even better performance can be achieved with HMMs based only on segmentation and labeling of song elements. However, a substantial amount of data is required to statistically train the HMMs' parameters to yield good classification accuracy. Hence, HMM-based techniques are not suitable under limited data conditions.

Besides DTW, support vector machines (SVMs) [66] and the nearest subspace classifier [67] have demonstrated good classification accuracies under limited training data conditions in various applications. The SVM belongs to the class of maximum margin classifiers. It performs binary (2-class) classification by finding a decision surface or hyperplane that has maximum distance to the closest points in the training set, which are termed support vectors [66]. In [68], it is reported that SVM is able to maintain high classification accuracies with a reduced amount of training data, outperforming two other machine learning algorithms – linear discriminant analysis and decision tree. On a content-based image retrieval (CBIR) task [69], and hyperspectral image classification, SVM also performs well when trained on a small number training samples. As for the nearest subspace (NS) classifier [67], it finds the class subspace spanned by the training vectors that best represents the input test vector. In hyperspectral image classification [70] and face recognition [71], the NS classifier has also shown

good classification results given a limited training set.

In addition to the various classification techniques, different features have also been used to represent birdsong segments (containing multiple syllables) or individual birdsong elements. These include using the time-frequency spectrogram explicitly [59, 64, 72, 73], frequency and energy trajectories [54, 74], Mel-frequency cepstral coefficients (MFCCs) [56, 57, 59], and other spectrographic image-based features [75].

1.4 Organization of the dissertation

The organization of this dissertation is as follows:

Chapter 2 proposes a multi-band summary correlogram (MBSC)-based pitch detection algorithm that is robust to a variety of noise conditions (noise-types and SNRs). Noise-robustness is achieved through novel signal processing schemes that enhance the maximum peak in the MBSC. The sensitivity of this degree-of-voicing feature to noise is reduced, such that good V/UV detection performance is achievable with a constant threshold and median filtering scheme. This chapter also includes information on the reference pitch corpora, performance measures and comparative algorithms that are used to evaluate the algorithms' pitch estimation and pitch detection performances.

Chapter 3 proposes the feature enhancement scheme based on jointly-sparse reference and estimated soft-mask representations for noise-robust speech recognition. The algorithm performs feature/spectral enhancement using a joint-dictionary of soft-mask representations. The Aurora-2 and Aurora-4 noisy speech databases are used to evaluate the performance of the proposed algorithm against other sparsity-based feature enhancement schemes. For the Aurora-2 digit ASR

task, exemplars are randomly selected to form the dictionaries. For the Aurora-4 large vocabulary ASR task, a sparsity-based dictionary learning algorithm is applied to the exemplars to build a more complete dictionary.

Chapter 4 proposes the exemplar-based sparse representation classifier for limited data birdsong phrase classification. The performance of a simple SR classifier for birdsong phrase classification and verification is investigated, as a proof of concept of the SR classification technique on bird songs. Dynamic time warping and an additional SR classification stage are subsequently introduced into the original SR classification framework, to improve the robustness of the SR classifier on automatically detected and segmented phrases, and on phrases sang by bird individuals that are different from those in the training set.

Finally, Chapter 5 summarizes this dissertation, and discusses possible future works.

CHAPTER 2

Multi-band Summary Correlogram-based Pitch Detector for Noisy Speech

In this chapter, a multi-band summary correlogram (MBSC)-based pitch detection algorithm (PDA) is proposed. The proposed MBSC PDA performs single-pitch detection of target speech in noise. The technical details of this algorithm are explained in the Section 2.1. Section 2.2 describes the corpora, performance metrics and comparative algorithms used to evaluate pitch estimation and pitch detection performances. Section 2.3 contains the results and a discussion of the pitch *estimation* performance evaluation, while the results and a discussion of the pitch *detection* performance evaluation are found in Section 2.4. Content in this chapter has been published in L. N. Tan, and A. Alwan, "Multi-band summary correlogram-based pitch detection for noisy speech", *Speech Communication*, Vol. 55, 2013, pp. 841-856.

2.1 Proposed MBSC pitch detector

The block diagram in Fig. 2.1 gives an overview of the proposed MBSC pitch detector. It consists of six main signal processing stages which are explained in the following subsections.

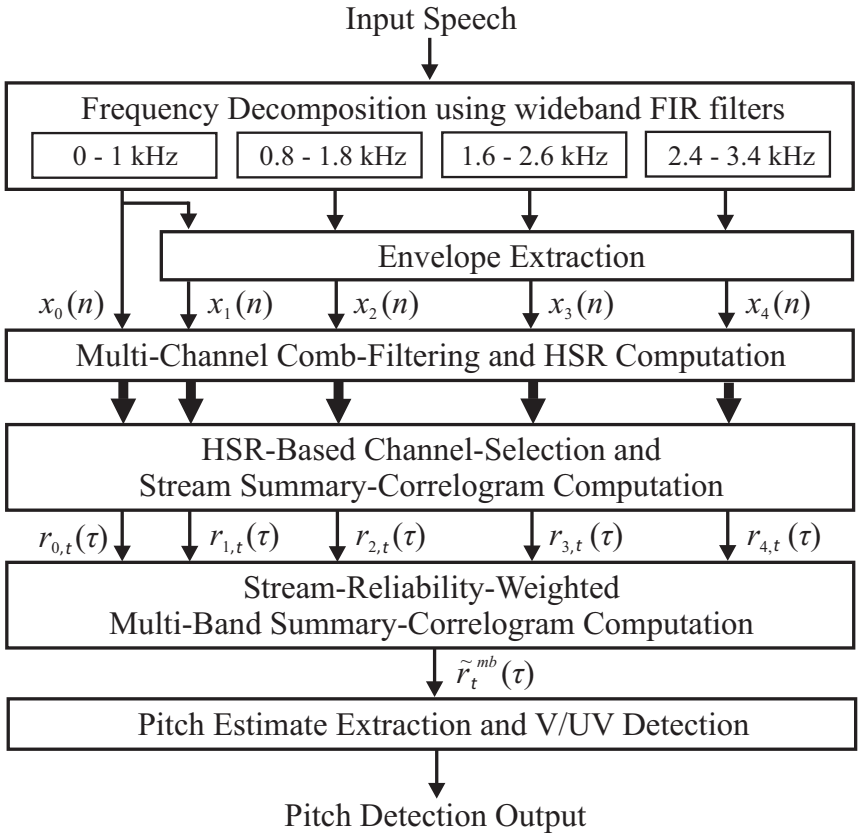

Figure 2.1: Block diagram of the proposed pitch detector. Multi-channel outputs are indicated by thick and bold arrows.

2.1.1 Frequency decomposition using wideband FIR filters

The input speech signal is first decomposed into four subbands using 32-point FIR filters. Filter cut-off frequencies are shown in Fig. 2.1. A 1-kHz filter bandwidth is chosen so that at least two harmonics are captured by each filter, for example, capturing harmonics at 400 Hz and 800 Hz in the first 1 kHz subband for an F0 of 400 Hz – the maximum pitch value of interest in our defined adult pitch range (50 – 400 Hz). When a signal contains more than 1 harmonic of the voiced speech, its envelope would typically oscillate at an amplitude modulation frequency corresponding to the inter-harmonic separation. A 0.2 kHz overlap between adjacent

filters results in a spectrum coverage up to 3.4 kHz with four such filters, which corresponds to the upper cut-off frequency of the G.712-characteristic filter [76]. In our PDA, wideband FIR filters are used instead of a Gammatone filterbank (e.g., in [12] and [13]), so that useful pitch information in the low-frequency envelope can be exploited. In [77], it is shown that inclusion of the low-frequency band signal envelope is especially useful in improving pitch estimation accuracy for speech whose low-frequency harmonics are severely attenuated due to G.712-filtering in telephone speech. In addition, FIR filters that are equally-spaced in the linear frequency domain (instead of a warped frequency scaling) are used, to give equal emphasis to all the harmonics in the available frequency range. The noise-robustness of the algorithm should increase with the number of FIR filters up to a certain limit, but with a higher cost in computational complexity. Four FIR filters are sufficient to give a relatively good pitch detection performance for the development set used.

2.1.2 Envelope extraction per subband

The Hilbert envelope [78] in each subband is extracted by computing the magnitude squared of the analytic signal, which is obtained by applying Hilbert transform on the FIR-filtered outputs. It is noted in [79] that the Hilbert envelope accurately follows the amplitude modulations of a bandpass signal. The Hilbert envelope is mean-normalized on a frame-by-frame basis before subsequent processing. The four mean-normalized envelope streams are denoted as $x_s(n)$, where $s = 1$, 2, 3, and 4 is the stream index, and n refers to the sample index. The lowpass-filtered, non-envelope stream from the first subband, labeled $x_0(n)$, is also used in subsequent processing (see Fig. 2.1). This non-envelope stream contains valuable information for pitch detection, especially when the first harmonic

is not attenuated or noise-corrupted, because the stream tends to be more periodic than the envelopes whose periodicities are more affected by signal energy variations.

2.1.3 Multi-channel comb-filtering per stream

Multi-channel comb-filtering is performed in the frequency domain, as shown in Eq. (2.1), by multiplying the input stream spectrum, $X(f)$, with a comb-function, $c(f)$, represented in the frequency domain. $X_{k,s,t}(f)$ and $Y_{k,s,t}(f)$ denote the complex discrete Fourier transform (DFT) coefficients of $x_s(n)$ in frame t, and its comb-filtered version, respectively. The comb-functions are formulated using raised-cosines, as shown in Eq. (2.2), where $c_k(f)$ is the kth channel's comb-function with an inter-peak frequency of F_k Hz, and f represents frequency. This comb-function enhances spectral harmonics spaced F_k apart, and suppresses the energies at the subharmonics. The raised-cosine function is selected due to its broad spectral peak lobes and smooth peak-to-valley transitions. By selecting a smooth comb-function [8], a slight signal inharmonicity (harmonic frequency perturbation from multiples of F0) would not result in sharp energy attenuation, in comparison to using an impulse-train-like comb-function [6,80], when the signal is filtered with a comb-function whose F_k is close but not exactly equal to the true F0. To reduce the dependency of the maximum autocorrelation (ACR) peak amplitude on the signal's fundamental period (ACR is applied on comb-filtered signals in a subsequent processing stage), N_k, the number of samples in $x_s(n)$ that is used to compute $X_{k,s,t}(f)$ is made proportional to the time periodicity enhanced by $c_k(f)$, i.e., $T_k = f_s/F_k$ samples, where f_s is the sampling frequency. It is found that $N_k = 4T_k$ samples are sufficient to achieve good pitch estimation and V/UV detection performance at low SNRs, but is not too large as to severely

degrade the resolution of V/UV boundary detection at high SNRs. To reduce the number of unique $X_{k,s,t}(f)$ computations, N_k is quantized to the multiple of 40 samples that is closest to $4T_k$.

$$Y_{k,s,t}(f) = \begin{cases} X_{k,s,t}(f)\, c_k^d(f), \text{ if } s = 0 \\ X_{k,s,t}(f)\, c_k(f), \text{ if } s = 1,\ 2,\ 3, \text{ or } 4 \end{cases} \tag{2.1}$$

$$c_k(f) = \begin{cases} \frac{1+\cos(2\pi f/F_k)}{2}, \text{ if } 0.5\ F_k \le f \le 1 \text{ kHz} \\ 0 \qquad\quad , \text{otherwise} \end{cases} \tag{2.2}$$

$$c_k^d(f) = d_k(f)\, c_k(f) \tag{2.3}$$

$$d_k(f) = 1 + \min(1, F_k/f) \tag{2.4}$$

From Eq. (2.1), it can be seen that constant-magnitude comb-functions – $c_k(f)$, are applied on the envelope streams, $s \ge 1$; while decreasing-amplitude comb-functions – $c_k^d(f)$, are used on the non-envelope stream, s=0. Eq. (2.3) obtains $c_k^d(f)$ by multiplying $c_k(f)$ with a decreasing function, $d_k(f)$, which is expressed in Eq. (2.4). In $c_k^d(f)$, the first harmonic has the largest gain of 2, while the gains of the higher harmonics decrease towards 1. Filtering $x_0(n)$ with a $c_k^d(f)$ whose F_k is near the true pitch value would boost the strength of the first harmonic relative to the higher ones. This helps reduce over-estimation errors in the presence of a very strong non-fundamental harmonic, such as in G.712-filtered telephone speech. Boosting the first harmonic in $x_{s\ge1}(n)$ is unnecessary, since the first harmonic of a Hilbert envelope is usually stronger than its higher harmonics. To reduce subharmonic estimation errors, decreasing-amplitude comb-functions have also been used in other comb-filter-based pitch estimation algorithms [8,80]. However, these comb-filter-based algorithms apply only one group of comb-filters (with various peak intervals) that spans the same frequency range. Thus, these algorithms depend heavily on the prominence of the lower frequency harmonics to

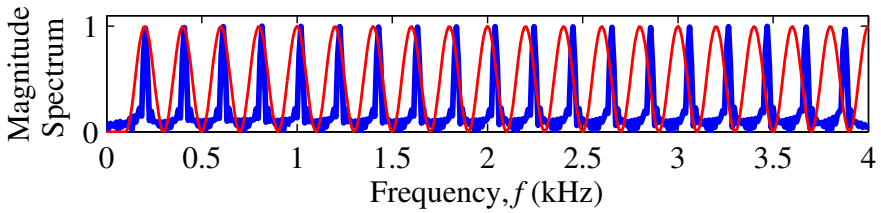

Figure 2.2: Peak deviation at high frequencies between a simulated signal (thick blue line) with an inter-harmonic peak separation of 204 Hz, and a comb-filter (thin red line) with inter-peak separation of 200 Hz.

obtain an accurate pitch estimate, and these harmonics can be masked by noise or attenuated in bandpass-filtered speech (e.g., telephone, hand-held radios [81], etc.). By incorporating separate comb-filters that span different subbands, the pitch detection performance of our PDA is more robust to such distortions.

In the proposed pitch detection algorithm, envelope extraction precedes comb-filtering, in contrast to an earlier pitch estimation algorithm proposed in [77], which performs comb-filtering prior to envelope extraction. One reason for this modification is to retain high-frequency harmonic information of a signal under the limitation of a finite $\mathrm{F0}_k$, as illustrated in Fig. 2.2. The spectrum of a simulated time-domain periodic impulse train (thick blue line) that has an F0 = 204 Hz is plotted with a $c_k(f)$ that has a $\mathrm{F0}_k$ of 200 Hz (thin red line). Although the peak interval of the two functions has a small relative difference of 2%, only the low- and mid-frequency harmonics of the impulse train fall near the spectral peaks of the comb-filter. This would result in undesirable attenuation for the high-frequency harmonics.

Through the process of Hilbert envelope extraction, signals are down-converted to baseband, such that their harmonics are located at the multiples of F0, and the harmonic amplitudes decrease with frequency. This effect is illustrated in

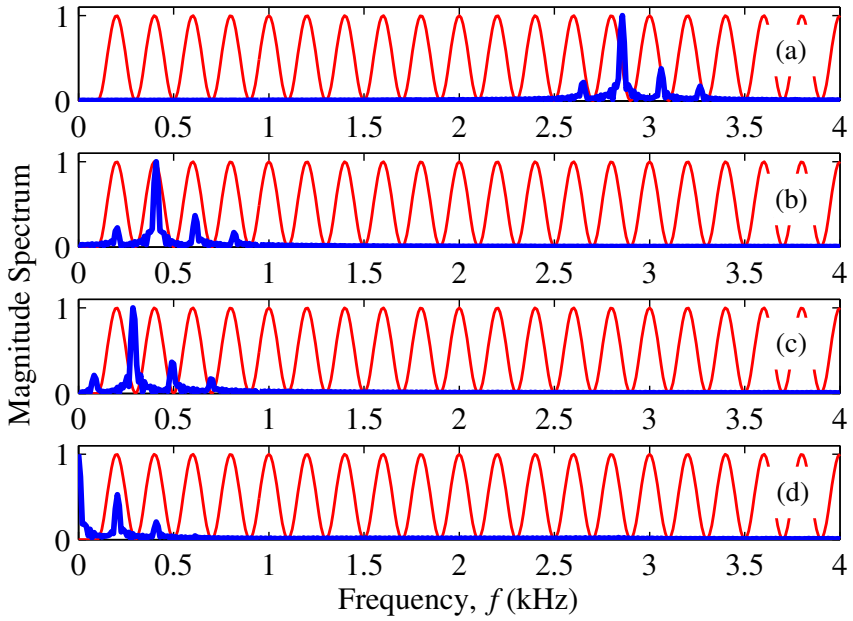

Figure 2.3: (a)-(c) Magnitude spectra (thick blue line) of simulated periodic signals containing a set of spectral harmonic components separated by 204 Hz that are placed at different frequency locations: (a) High-freq subband; (b) Low-freq subband; (c) Same as (b) but shifted down by 120 Hz. (d) Magnitude spectrum of the Hilbert envelope computed from the time-domain signals corresponding to the spectra in (a), (b), and (c). A comb-filter (thin red line) with an inter-peak frequency of 200 Hz is superimposed in (a)-(d).

Fig. 2.3 for three different periodic signals. Following the example in Fig. 2.2, the F0 of the simulated signal is set to 204 Hz. The periodic signals are simulated using a summation of cosine waveforms, i.e. $\sum_i A_i \cos(2\pi f_i t)$. Fig. 2.3a plots the magnitude spectrum (thick blue line) of the signal containing cosine components with f_i corresponding to the 13th to 16th harmonic frequencies in the fourth subband; Fig. 2.3b plots the the magnitude spectrum of the signal containing cosine components with f_i corresponding to the 1st to 4th harmonic

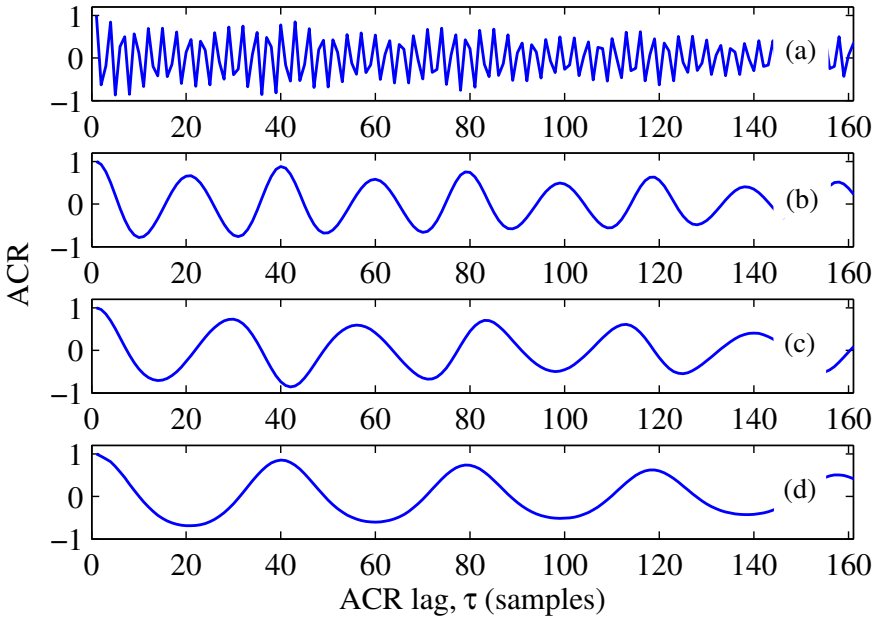

Figure 2.4: (a)-(d) ACR functions of time-domain signals whose magnitude spectra are plotted in Fig. 2.3(a)-(d), respectively.

in the first subband, while Fig. 2.3c plots a shifted version of 2.3b that has been shifted down by 120 Hz, to simulate frequency-shifted data. Fig. 2.3d plots the magnitude spectrum of the Hilbert envelope computed from all three periodic signals represented by 2.3a-c. The magnitude spectra of the Hilbert envelopes computed from all three periodic signals looked identical, thus only 1 version is plotted in 2.3d. The comb-function, $c_k(f)$ with an inter-peak frequency of 200 Hz (thin red line) is also superimposed on all magnitude spectra. In 2.3a and 2.3c, it is clear that comb-filtering will lead to undesirable attenuation of the harmonic components if envelope extraction is not performed prior to comb-filtering.

To highlight the advantage of performing envelope extraction for all subbands (independent of comb-filtering), ACR functions of the time-domain signals represented in Fig. 2.3a-d are plotted in Fig. 2.4a-d, respectively. The true T0 $\approx$ 39

samples ($\because f_s$=8000Hz/F0=204Hz). The maximum ACR peak in Fig. 2.4a-d are respectively located at ACR lags of τ = 42, 39, 29 and 39 samples. Hence, it is evident a more accurate F0 estimate can be obtained from the Hilbert envelope extracted from high-frequency subband signal and frequency-shifted, compared to using the original signals. In 2.4b, although the maximum ACR peak is located at the true T0, this peak is less prominent compared to that observed in 2.4d, due to additional strong peaks that are present at τ = 20, 59, ... samples. Thus, $\tau = 20$ could be erroneously taken as the signal's T0 estimate, if one selects the best T0 estimate by taking the first prominent ACR peak in 2.4b, instead of the maximum ACR peak. Our earlier paper [77] had also reported a significant reduction in F0 estimation errors for G.712-filtered speech, when the Hilbert envelope of the low-frequency band is utilized.

2.1.4 HSR-based comb-channel selection per stream

2.1.4.1 HSR computation

Harmonic-to-subharmonic energy ratio (HSR) is a measure computed to aid the selection of reliable comb-filter channels per stream. Comb-filters defined in Section 2.1.3 capture the harmonic energy of the signal. To capture the subharmonic or inter-harmonic energy, inverted comb-filters, $c_k^-(f)$ and $c_k^{d-}(f)$, are designed to pair with each $c_k(f)$ and $c_k^d(f)$, as shown in Eqs. (2.5) and (2.6). Fig. 2.5 plots the $c_k(f)$, $c_k^-(f)$, $c_k^d(f)$, and $c_k^{d-}(f)$ with an inter-peak frequency, F_k of 200 Hz. The HSR of the kth channel in stream s, denoted by $q_{s,t}(k)$, is computed using Eq. (2.7). For a comb-function whose inter-peak frequency, F_k, is close to

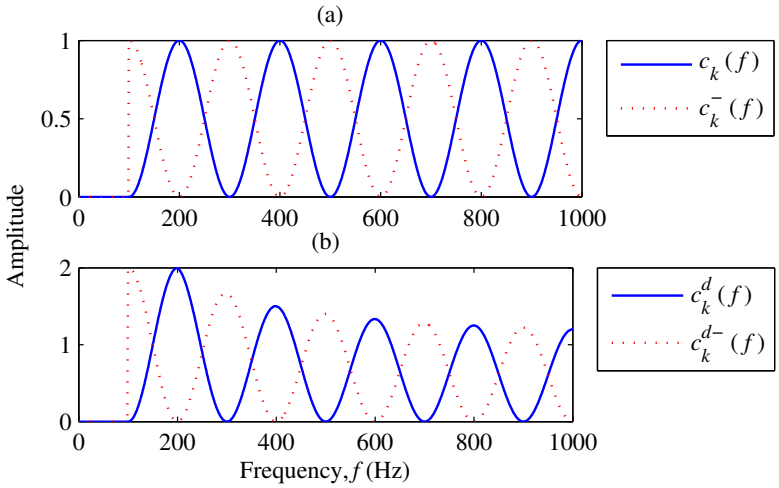

Figure 2.5: Comb-filters with an inter-peak frequency of $F_k = 200$ Hz. (a) Non-decreasing comb-filter, $c_k(f)$ and its inverted counterpart, c_k^-. (b) Decreasing comb-filter, $c_k^d(f)$ and its inverted counterpart, c_k^{d-}.

the input signal's true pitch value, its HSR would be high.

$$c_k^-(f) = \begin{cases} 1 - c_k(f), \text{ if } 0.5\ F_k \leq f \leq 1 \text{ kHz} \\ 0 \qquad , \text{otherwise} \end{cases} \tag{2.5}$$

$$c_k^{d-}(f) = d_k(f)\, c_k^-(f) \tag{2.6}$$

$$q_{s,t}(k) = \begin{cases} \sum_f |X_{k,s,t}(f)\, c_k^d(f)|^2 / \sum_f |X_{k,s,t}(f)\, c_k^{d-}(f)|^2, \text{ if } s = 0 \\ \sum_f |X_{k,s,t}(f)\, c_k(f)|^2 / \sum_f |X_{k,s,t}(f)\ c_k^-(f)|^2\ , \text{ if } s = 1,2,3,4 \end{cases} \tag{2.7}$$

In the presence of noise, one frame of speech might be more corrupted than its neighboring frames, such that inter-frame peak consistency in $q_{s,t}(k)$ is affected. Since the pitch contour of natural speech generally varies smoothly in time, a lowpass IIR filter is applied on $q_{s,t}(k)$ to improve its inter-frame peak consistency, as shown in Eq. (2.8). This time-smoothed HSR is denoted by $\tilde{q}_{s,t}(k)$.

$$\tilde{q}_{s,t}(k) = 0.5\, \tilde{q}_{s,t-1}(k) + 0.5\, q_{s,t}(k) \tag{2.8}$$

2.1.4.2 Tri-stage channel selection

The proposed PDA does not use the computed HSR directly for pitch estimation, because the HSR computed at a subharmonic can sometimes be as high as or higher than the HSR at an F_k near the true F0. Instead, the HSR is used to identify reliable comb-channels in each stream. Channel selection is performed on a per stream basis (i.e., channels selected in stream $s = 0$ are independent of those selected in other streams), using a novel tri-stage selection process designed to improve both pitch estimation and voicing detection performance. The flowchart of this tri-stage channel selection is shown in row (I) of Fig. 2.6, while row (II) shows the HSR, $\tilde{q}_{s,t}(k)$, for a particular $x_{s,t}(n)$ in blue, with the channels selected in each stage indicated by the red asterisks. Row (III) of Fig. 2.6 contains the ACRs computed from the comb-filtered outputs of channels selected in Stage 2. For ease of visualization and comprehension of the selection algorithm described subsequently, F_k instead of channel index k, is used for labeling the horizontal axes in the HSR plots, and a "clean" voiced frame example is used to show a clear subharmonic relationship in the HSR peaks.

In Stage 1, channels corresponding to peaks in $\tilde{q}_{s,t}(k)$ that have an amplitude greater than 1 are selected (Fig. 2.6(IIa)). Taking the peaks in $\tilde{q}_{s,t}(k)$ ensures that only the best-matched comb-filters among their neighbors with similar F_k are selected, while setting an amplitude threshold of 1 selects the channels whose harmonic energy is greater than its subharmonic counterpart.

If $x_s(n)$ has a pitch value of f_0, there should be peaks in $\tilde{q}_{s,t}(k)$ at $F_k \approx f_0$, and its subharmonics, i.e., at $F_k \approx f_0/2$. Hence, in Stage 2, comb-channels with $F_k = f_a$ and $0.5f_a$ are retained (if both are selected in Stage 1, see Fig. 2.6(IIb)). To implement this selection scheme, HSR is also evaluated using $c_k(f)$ with F_k that are half of those defined in the original search array. Since voiced speech

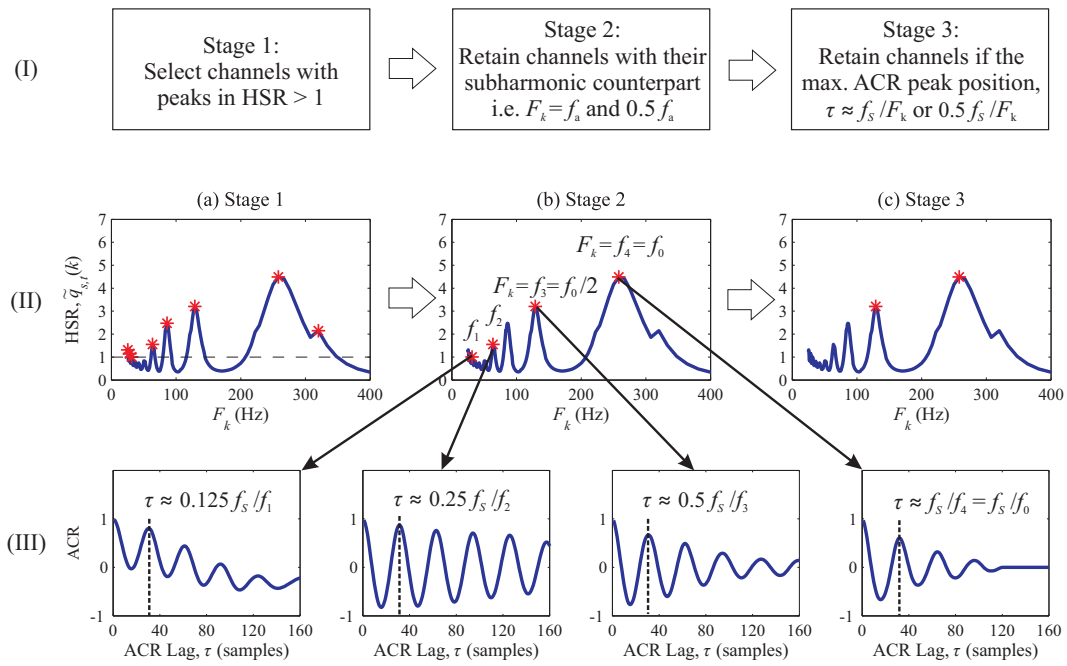

Figure 2.6: (I) Flowchart of the tri-stage channel selection. (II) HSR, $\tilde{q}_{s,t}(k)$ (blue line) and selected channels (red asterisks) in (a) Stage 1, (b) Stage 2, and (c) Stage 3, of a particular $x_s(n)$ with $f_0 \approx 260$ Hz, for a clean voiced frame. (III) ACRs computed from comb-filtered outputs of channels selected in Stage 2. Channels with $F_k \approx 130$ Hz and 260 Hz are selected after this tri-stage selection process in this example.

is generally not perfectly harmonic, especially in noise, the subharmonic channel selection criterion is relaxed to within $\pm 20\%$ of $0.5 f_a$.

From Fig. 2.6(IIb), it can be observed that besides comb-channels with $F_k = f_0$ and $f_0/2$, comb-channels with F_k corresponding to lower subharmonic frequencies (f_0/i, $i > 2$) might also be selected. The number of these lower subharmonic frequencies within the pitch range of interest increases with f_0. Since longer frame lengths (N_k) are used in comb-channels with lower F_k, stopping channel selection at Stage 2 would result in early/late detections of a voicing onset/offset,

especially when f_0 is high. Hence, a third selection stage is implemented to make the number of channels selected less dependent on f_0. An energy-normalized ACR is computed from each comb-filtered, time-domain signal, $y_{k,s,t}(n)$ (inverse-FFT of $Y_{k,s,t}(f)$), of the selected channels in Stage 2. If the pitch value is f_0, the maximum ACR peak should still be located at $\tau = f_s/f_0$ samples, even for the subharmonic comb-channels, as shown in the ACRs plotted in row (III) of Fig. 2.6. To retain comb-channels with $F_k = f_0$ and $f_0/2$, channels whose maximum ACR peak is located within a 20% deviation tolerance of $\tau = f_s/F_k$ or $\tau = 0.5f_s/F_k$ are selected in Stage 3, as shown in Fig. 2.6(IIc). Retaining an additional subharmonic channel with $F_k = f_0/2$ helps improve voicing detection at low SNRs.

2.1.5 SC computation per stream

After channel selection, a correlogram matrix representation, $R_{s,t}(k', \tau)$, is formed per stream by vertically stacking the energy-normalized ACR functions of the selected K_s channels, where K_s is the total number of channels selected in stream s, and $k' = 1, 2, ..., K_s$ represents renumbered indices of the selected channels. Instead of a simple average of the selected channels' ACRs to get each stream's summary correlogram (SC), as done in [12], an HSR-based weighted averaging scheme is performed by multiplying the stream's HSR-based channel-weighting function, $\phi_{s,t}(k')$, to its corresponding correlogram matrix, $R_{s,t}(k', \tau)$, to obtain the stream SC, $r_{s,t}(\tau)$, as shown in Eqs. (2.9) - (2.11). A value of one is subtracted from $\tilde{q}_{s,t}(k')$ in Eq. (2.11) to increase the relative differences of the values in $\phi_{s,t}(k')$, since every value in $\tilde{q}_{s,t}(k')$ is greater than one. The process of computing the HSR-weighted stream SC from the selected channels' ACRs is also illustrated in Fig. 2.7, which is performed on a babble noise-corrupted voiced speech frame

at 5 dB SNR that has a fundamental period ≈ 68 samples. Fig. 2.8 shows the HSR-weighted stream SC computed at each s, for same babble noise-corrupted voiced speech frame.

$$r_{s,t}(\tau) = \sum_{k'} \phi_{s,t}(k') R_{s,t}(k', \tau) \tag{2.9}$$

$$\phi_{s,t}(k') = \frac{Q_{s,t}(k')}{\sum_{k'} Q_{s,t}(k')} \tag{2.10}$$

$$Q_{s,t}(k') = \tilde{q}_{s,t}(k') - 1 \tag{2.11}$$

Through this weighting scheme, ACRs from the more reliable channels (with higher HSR) will have a greater impact on their stream SC, $r_{s,t}(\tau)$, resulting in a more prominent ACR peak at the most likely pitch period of the signal. Peak prominence in each stream SC is also due to the use of harmonically-enhanced comb-filtered signals in the individual channel's ACR computation. This effect is shown in Fig. 2.9 for the same 5 dB SNR babble noise-corrupted voiced speech frame used in Fig. 2.7. Row (I) plots the energy-normalized ACR function of pre-comb-filtered $x_s(n)$. A frame length of 4 times the true fundamental period (T0), is used to compute these ACRs for a fair comparison. Row (II) plots the SCs obtained by using a simple average of selected channels' ACRs in each stream, while row (III) plots the SC, $r_{s,t}(\tau)$, obtained using the proposed HSR-based weighting scheme. In this example, it is evident that the SCs in row (II) for s=1, 3, and 4 have a more prominent peak compared to the respective stream's ACR of pre-comb-filtered $x_s(n)$ in row (I), and the HSR-weighted SCs in row (III) for s=0, 3, and 4 have a more prominent peak at the true T0 compared to the equal-channel-weighted SCs in row (II).

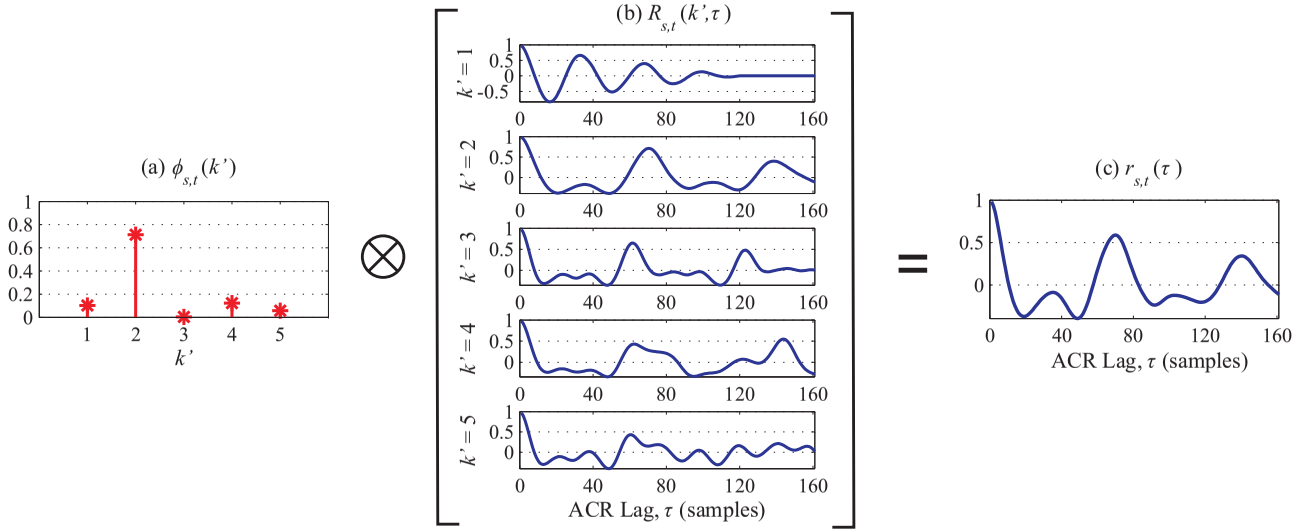

Figure 2.7: (a) HSR-based channel-weighting function, $\phi_{s,t}(k')$, (b) Selected channels' ACRs in the correlogram matrix, $R_{s,t}(k', \tau)$, and (c) HSR-weighted stream summary correlogram, $r_{s,t}(\tau)$ in stream $s = 0$, for a babble noise-corrupted voiced frame with a fundamental period ≈ 68 samples and an SNR of 5 dB.

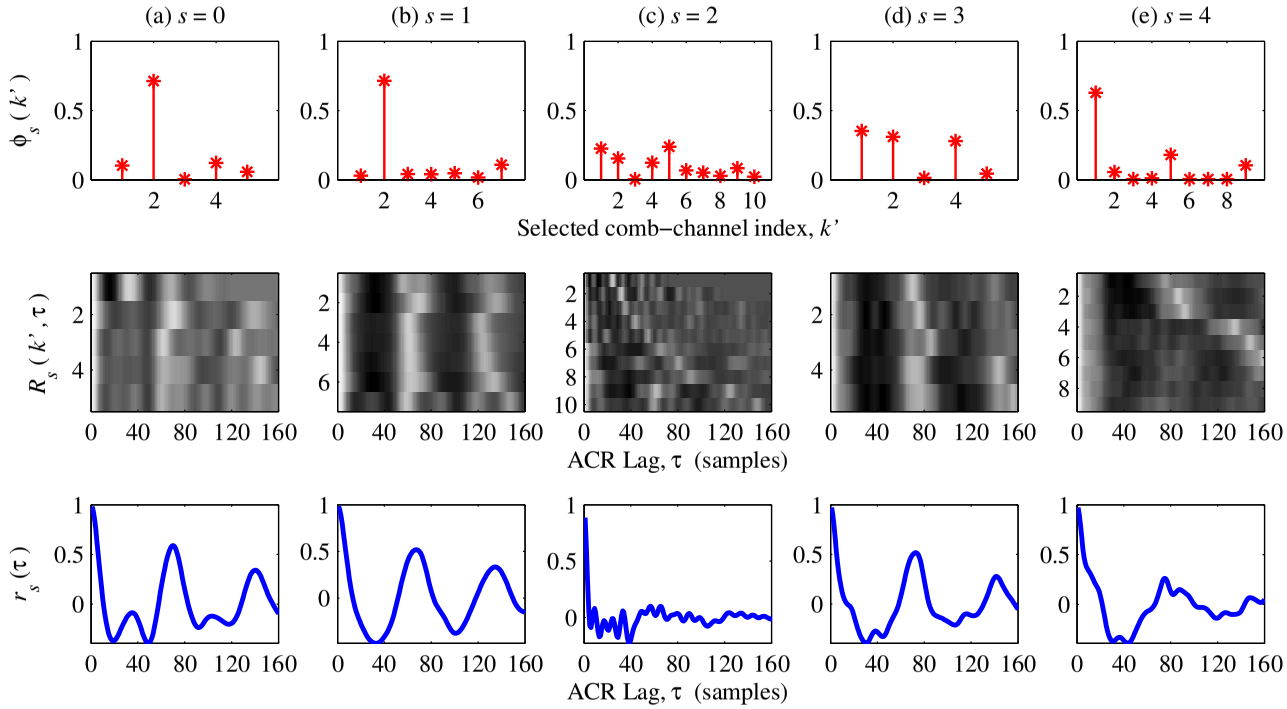

Figure 2.8: Computation of the HSR-weighted stream summary correlograms for all streams, one column for each stream, s. The first row shows the HSR-based weight, $\phi_s(k')$, of each selected comb-channel. The second row shows the correlogram matrices, $R_s(k', \tau)$, and the third row shows the HSR-weighted stream summary correlograms, $r_s(\tau)$, for the same babble noise-corrupted voiced frame in Fig. 2.7

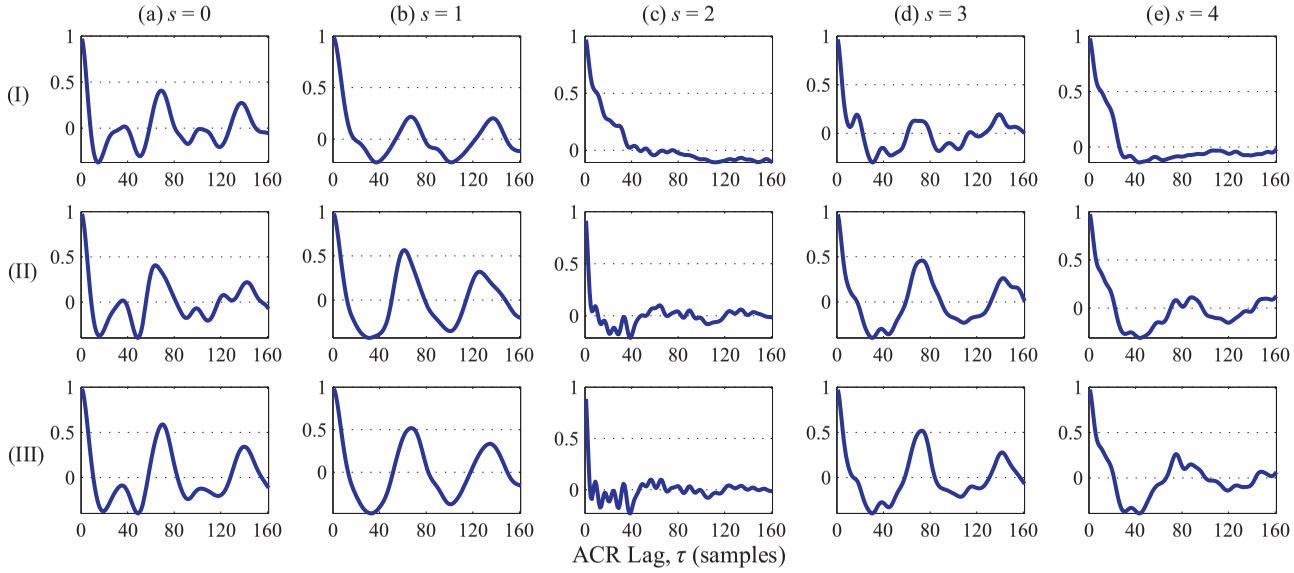

Figure 2.9: (I) ACRs computed from each pre-comb-filtered $x_s(n)$, (II) Stream SCs obtained by averaging the selected channels' ACR, and (III) Stream SC obtained using the proposed HSR-weighting scheme, for $s = 0, 1, ..., 4$. All ACRs and SCs are computed at the same babble noise-corrupted voiced frame as that used in Fig. 2.8.

2.1.6 MBSC computation

The stream SCs are further fused into one multi-band summary correlogram (MBSC), $r_t^{mb}(\tau)$. The contributions of the stream SCs are determined by a stream-reliability-weighting function, $w_t(s)$, as shown in Eq. (2.12). It is observed that the maximum HSRs in the more reliable streams are higher. In addition, reliable $r_{s,t}(\tau)$ tend to have similar peak locations. Hence, the values of $w_t(s)$ in Eq. (2.13) are made dependent on two factors: (1) a within-stream reliability factor, $\alpha_t(s)$, based on the maximum value of $Q_{s,t}(k')$ in each stream, as shown in Eq. (2.14); (2) a between-stream reliability factor, $\beta_t(s)$, that corresponds to the number of $r_{s,t}(\tau)$ whose maximum peak position, $g_t(s)$, falls within 10% of its own, as defined in Eqs. (2.15) - (2.17). A stricter deviation tolerance of 10% is set in Eq. (2.16) compared to the 20% tolerance allowed in channel selection, because there is a high tendency of assigning a large $\beta_t(s)$ to an unreliable $r_{s,t}(\tau)$ when a larger tolerance is set.

$$r_t^{mb}(\tau) = \sum_{s=0}^{4} w_t(s)\ r_{s,t}(\tau) \tag{2.12}$$

$$w_t(s) = \frac{\alpha_t(s)\ \beta_t(s)}{\sum_{s=0}^{4} \alpha_t(s)\ \beta_t(s)} \tag{2.13}$$

$$\alpha_t(s) = \max_{k'}\ Q_{s,t}(k') \tag{2.14}$$

$$\beta_t(s) = \sum_{s'=0}^{4} p_t(s, s') \tag{2.15}$$

$$p_t(s, s') = \begin{cases} 1, \text{ if } |1 - \frac{g_t(s')}{g_t(s)}| < 0.1 \\ 0, \text{ otherwise} \end{cases} \tag{2.16}$$

$$g_t(s) = \arg\max_{20 \le \tau \le 160} \text{peak}\ \ r_{s,t}(\tau) \tag{2.17}$$

Fig. 2.10a and 2.10b show the within- and between- stream-reliability factors, $\alpha(s)$ and $\beta(s)$, respectively, that are computed from the same speech frame example used in Fig. 2.8. Fig. 2.10c shows the stream-reliability-weighting function,

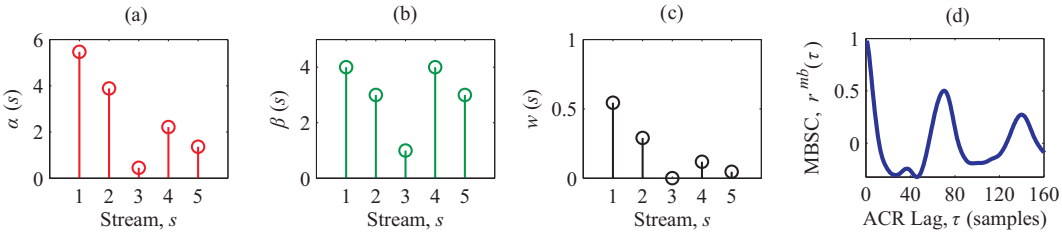

Figure 2.10: (a) Within-stream reliability factor, $\alpha(s)$, (b) between-stream reliability factor, $\beta(s)$, (c) stream-reliability-weighting function, $w(s)$, and (d) the MBSC obtained using the proposed stream-reliability weighting scheme. These are computed at the same babble noise-corrupted voiced frame as that used in Fig. 2.8.

$w(s)$ computed from $\alpha(s)$ and $\beta(s)$, while Fig. 2.10d shows the MBSC obtained when this $w(s)$ is applied to the stream SCs, $r_s(\tau)$, in Fig. 2.8.

Fig. 2.11a plots the MBSC (in blue solid line) obtained by applying an equal-stream-weighting scheme on the stream SCs, $r_{s,t}(\tau)$, in Fig. 2.8. On the other hand, Fig. 2.11b plots the MBSC (in blue solid line) obtained with the proposed stream-reliability-weighting scheme on the same SCs. When these two multi-band SCs are compared to their counterparts (in red dashed lines) calculated from clean speech, it can be observed that difference in MBSC peak amplitudes between the clean and the 5 dB versions are larger with the equal-stream-weighting, compared to the proposed weighting technique. The differences between the two schemes will be larger if there are fewer reliable stream SCs than those present in this example. Thus, the stream-reliability-weighting scheme reduces the variability of the maximum peak amplitude in the MBSC for noisy speech, such that this amplitude becomes a robust indicator of the frame's degree of voicing.

To improve the inter-frame consistency of the MBSC's maximum peak location across a continuous voiced segment, the same lowpass IIR filter in Eq. (2.8)

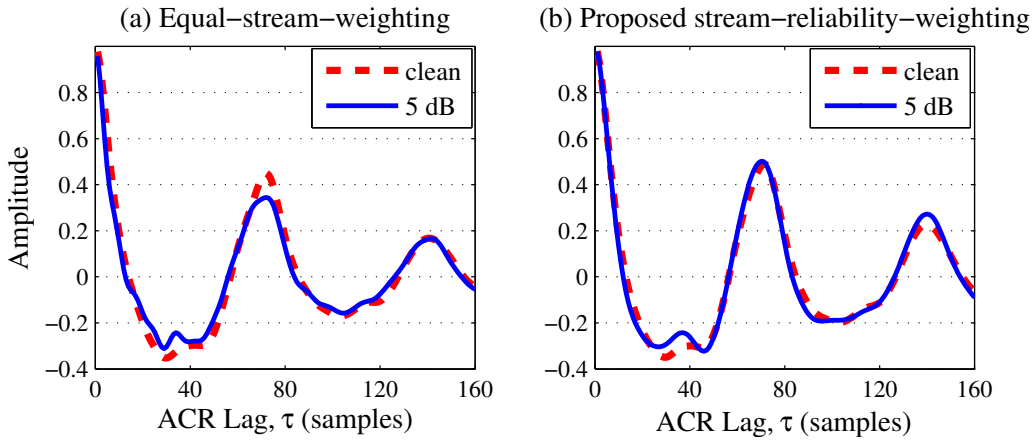

Figure 2.11: MBSCs computed using (a) an equal-stream-weighting scheme, and (b) the proposed stream-reliability weighting scheme on the stream SCs, plotted in row (III) of Fig. 2.9.

is applied on $r_t^{mb}(\tau)$ to obtain a time-smoothed $\tilde{r}_t^{mb}(\tau)$. Time-smoothing also slows down the rate of decrease of peak amplitudes, which in turn improves the detection of weak voiced frames at voicing offsets.

2.1.7 Pitch estimates and V/UV decisions

From $\tilde{r}_t^{mb}(\tau)$, pitch candidates corresponding to the 10 highest peaks with amplitudes greater than 0 are identified. Each peak and its immediate neighbors are fitted by a parabola, and the amplitude and lag position corresponding to the maximum point of this parabola are the refined pitch measurements for the respective pitch candidate [4]. To favor the pitch candidate with the smallest MBSC lag position when peaks of similar amplitudes are present, an empirically-determined, linearly decreasing lag-weighting function, $\lambda(\tau) = 1 - \frac{0.7}{\tau_{\max}}\tau$, is multiplied to the interpolated peak amplitudes [3], where $\tau_{\max} = 160$ is the maximum number of samples in one period (given an 8-kHz signal, and the minimum F0 set at 50 Hz). The lag position of the best pitch candidate (the one with the highest

lag-weighted peak amplitude) gives the estimated pitch period.

As for V/UV detection, a constant threshold is applied on the maximum interpolated peak amplitude (prior to lag-weighting) to obtain the initial binary V/UV decision for each frame. This is followed by a 5-point median filtering in time on these initial decisions to get the final V/UV detection outputs.

2.2 Database, comparative algorithms, and performance metrics

2.2.1 Reference pitch corpora for development and evaluation

The two popular pitch corpora for speech – Keele [82] and CSTR [83] are used to generate narrowband noisy speech, with a sampling rate of 8 kHz (downsampled from the original clean version at 20 kHz). The dataset generated from Keele corpus is the development (dev) set used for algorithmic parameter tuning, while the dataset generated from the CSTR corpus is treated as the evaluation (eval) set. The reference pitch contours (provided in the corpora) are derived from laryngograph signals recorded simultaneously.

2.2.1.1 Dev set – Noise-added Keele

The Keele corpus contains a phonetically balanced story read by five adults from each gender: one file per speaker, each about 30 seconds long. To simulate bandlimited noisy speech, the data is down-sampled and mixed it with three real-world noise types – babble, car (volvo), and machine gun. These are sample noise files from the NOISEX-92 corpus [84] (available at `http://spib.rice.edu/spib/select_noise.html`). They are selected for their different degrees

Table 2.1: Characteristics of the noise-types used to generate noisy speech

Noise type	**Spectral characteristics**
Babble	Non-stationary speech-shaped harmonics
Car	Highly stationary low-frequency noise
Machine gun	Highly non-stationary, containing:
	- No-noise time segments
	- Short bursts of energy covering all frequencies
	- Each burst is followed by a longer low-frequency noise

of stationarity and spectral characteristics as presented in Table 2.1 and Fig. 2.12. The noise files are also downsampled to 8 kHz before adding them to clean speech at 20, 15, 10, 5, and 0 dB SNR using the Filtering-and-Noise-adding-Tool (FaNT) [85]. Speech with fullband or G.712 [76] spectral characteristics is generated by setting the "-m snr_4khz" or "-f g712" option, respectively in FaNT. The ITU G.712 characteristic filter has a flat bandpass response between 300 and 3400 Hz. Since fundamental harmonics below 300 Hz are attenuated, the G.712 dataset is a more challenging corpus than its fullband counterpart.

2.2.1.2 Eval set – Noise-added CSTR

The CSTR corpus contains about 5 minutes of speech from an adult male and an adult female (50 sentences each). The phonemes in the sentences are biased towards voiced fricatives, nasals, liquids and glides, for which accurate pitch estimation is difficult. Since the pitch contours in the reference files are not computed at a regular rate of 100 Hz, linear interpolation and extrapolation are applied to obtain a reference pitch value at every 10 ms for performance evaluation.

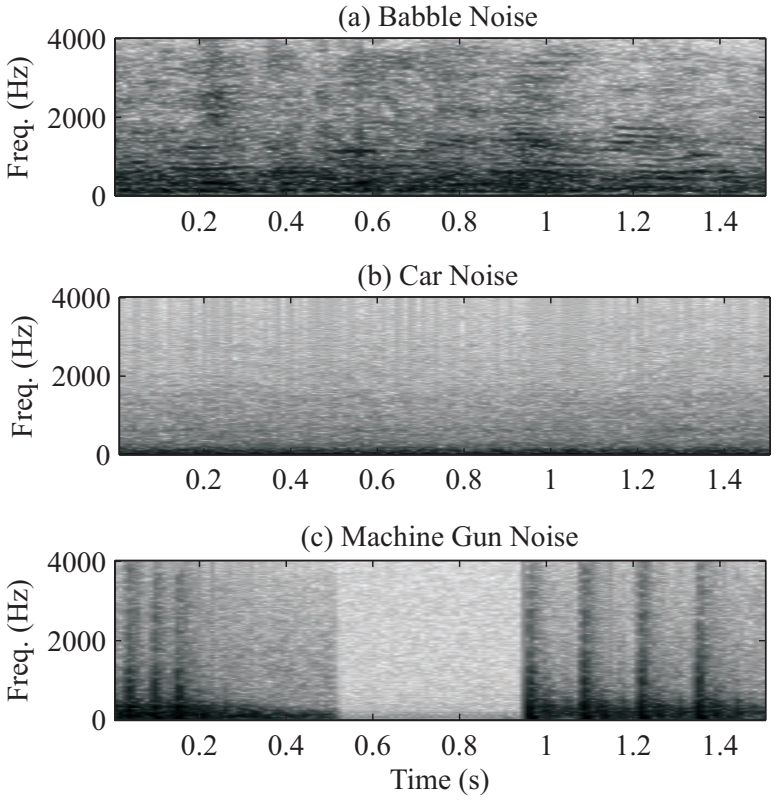

Figure 2.12: Spectrograms of the three noise types: (a) babble, (b) car noise, and (c) machine gun.

2.2.2 Algorithms for comparison

The following pitch estimation/detection algorithms are evaluated on the same pitch corpora for comparison purposes. These algorithms are selected because they are popular benchmarking algorithms – RAPT [3] used in Wavesurfer [86], YIN [4]) – or they employ signal processing techniques that have similarities to our proposed pitch detector – SHR [6], SWIPE, SWIPE' [8], WWB [13]. The default parameter values proposed in these algorithms are used, unless specified otherwise in this chapter. The following parameters are set common in all algorithms – sampling rate = 8 kHz, pitch range from 50 to 400 Hz, and frame rate = 100 Hz.

1. RAPT [3]. This is a time-domain pitch detection algorithm that is implemented in Wavesurfer [86]. This algorithm uses cross-correlation to obtain F0 candidates, followed by a dynamic programming algorithm to yield the final pitch contour. The Wavesurfer software is available at `www.speech.kth.se/wavesurfer/download.html`.

2. YIN [4]. This is a time-domain pitch estimation algorithm in which a series of improvements on the classic AC algorithm are introduced to reduce F0 estimation errors. This improved AC-based technique reported high F0 estimation accuracies for clean speech and music. YIN's MATLAB code is available at `audition.ens.fr/adc/sw/yin.zip`.

3. SHR [6]. This pitch detection algorithm has a frequency domain PEA based on a subharmonic-to-harmonic ratio (SHR), and a rule-based V/UV detection algorithm based on time-domain parameters. SHR is the ratio of the spectrum's subharmonic amplitude summation to its harmonic counterpart. F0 estimation is performed on frames with energies higher than the estimated noise floor, with a subsequent V/UV detection based on a correlation value and zero-crossing rate of the extracted frame. To download SHR's MATLAB code, search for "pitch determination algorithm" at `www.mathworks.com/matlabcentral/fileexchange`.

4. SWIPE and SWIPE' [8]. SWIPE and SWIPE' (a variant of SWIPE) are recently proposed comb-based, frequency domain F0 estimation algorithms. In contrast to SWIPE, SWIPE' uses only the first and prime harmonics in its decaying comb-function for F0 estimation, which significantly reduces subharmonic errors. It was shown in [8] that SWIPE' outperformed 12 other PEAs when evaluated on Keele, CSTR, a pathological voice database, and a music database. The MATLAB code is available at `www.cise.ufl.edu/`

~acamacho/publications/swipep.m

5. WWB [13]. This multi-pitch-tracker is a time-frequency-domain, correlogram-cum-statistical-model-based F0 tracker proposed by Wu et. al (2003). It performs frequency decomposition using a Gammatone filterbank, and computes AC on each channel. For the mid- and high-frequency channels, signal envelopes are extracted prior to AC computation. This is followed by channel and peak selections to enhance the peak corresponding to the true fundamental period in the SCgram. Statistical information on the peaks in the SCgram are post-processed using a hidden Markov model (HMM)-framework to perform pitch tracking. It is shown in [13] that this algorithm yields reliable single and double F0 contours for noisy speech. With its original code, less than 6 s of speech can be processed even after the "MAX_SAMPLES" value is maximized to a value that avoids insufficient memory problem at runtime. This is not a problem for the CSTR dataset since the maximum utterance length is 5 s. To work around this issue for the Keele dataset, the input data is broken into 5-seconds long segments with 100 ms overlap. The multi-segment F0 outputs are then concatenated into a single file by removing the last 4 values in the former segment, and first 5 values in the latter segment for each pair of overlapping segments. This concatenation procedure results in the correct number of F0 outputs in the final concatenated file, and the minimum F0 tracking error for the 8 kHz fullband clean Keele corpus. Codes for this pitch tracker is available at www.cse.ohio-state.edu/~dwang/pnl/shareware/wu-tsap03.

2.2.3 Performance metrics

Two experiments are conducted: Experiment 1 focuses only on pitch estimation accuracy, and all voiced frames in the reference pitch contours are evaluated, i.e. assuming V/UV detection is perfect. Experiment 2 evaluates pitch detection accuracy by taking into account both pitch estimation and V/UV detection errors. The performance metric used for performance evaluation in Experiment 1 is the gross pitch error for all reference voiced frames (GPE_{refV}), while the key performance metric in Experiment 2 is the pitch detection error (PDE). These performance metrics are explained in the following subsections.

2.2.3.1 Experiment 1: Pitch estimation

Since perfect V/UV detection is assumed in Experiment 1, only pitch estimation errors can occur under this pitch detection scenario. Hence, the gross pitch error computed over all voiced frames noted in the reference files is the performance metric used in this experiment. This popular performance metric, denoted by GPE_{refV} in Eq. (2.18), has also been used by developers of comparative algorithms in [4,6,8]. N_V and $N_{P,refV}$ are the number of reference voiced frames, and the number of frames with gross pitch errors (estimated and reference F0 differ by more than 20%), respectively.

$$GPE_{refV} = \frac{N_{P,refV}}{N_V} \times 100\% \tag{2.18}$$

To ensure there is an F0 estimate for almost every reference voiced frame, voicing-related parameters in RAPT, SHR and WWB are altered – RAPT's "VO_BIAS" to 1; SHR's "CHECK_VOICING" to 0; and state transitions probabilities ($\Omega_i \rightarrow \Omega_j$) in WWB are set to 0, except for $\Omega_0 \rightarrow \Omega_1$, $\Omega_1 \rightarrow \Omega_1$, and $\Omega_2 \rightarrow \Omega_1$, which are set to 1. For the proposed MBSC algorithm, the estimate given by the best

pitch candidate of each frame is used for computing GPE_{refV}.

2.2.3.2 Experiment 2: Pitch detection

Both pitch estimation and V/UV detection accuracies are of interest in Experiment 2. Three performance metrics are computed – Gross Pitch Error (GPE), Voicing Decision Error (VDE) [17], and Pitch Detection Error (PDE) [87]. In contrast to Experiment 1, GPE in Eq. (2.19) is only computed over the reference voiced frames that is detected by the respective algorithm, whose count is represented by N_{VV}, and N_P is the number of frames with gross pitch errors. VDE in Eq. (2.20) is the percentage of V/UV detection errors. $N_{V\rightarrow UV}$ is the number of voiced frames misclassified as unvoiced, and vice versa for $N_{UV\rightarrow V}$, while N is the number of frames in the utterance. A pitch detector can have a low GPE, but a high VDE because many challenging voiced frames are misclassified as unvoiced, and vice versa, which makes pitch detection performance comparison based on these two separate metrics difficult. Thus, PDE in Eq. (2.21), which represents the percentage of pitch detection errors is the key performance metric used for comparative evaluation in this experiment. The PDE takes into account all possible mutually exclusive pitch detection errors that can occur in any given frame, i.e., voiced-to-unvoiced, unvoiced-to-voiced, or gross pitch error. This performance metric is formally defined in [87], and it has also been used for pitch detection performance evaluation in [88] and [12]. Fig. 2.13 [87] gives an illustration of the pitch error frames, $N_{V\rightarrow UV}$, $N_{UV\rightarrow V}$, and N_P that are found, when a detected F0 contour is compared against the reference F0 contour that contains N frames.

$$GPE = \frac{N_P}{N_{VV}} \times 100\% \tag{2.19}$$

$$VDE = \frac{N_{V\rightarrow UV} + N_{UV\rightarrow V}}{N} \times 100\% \tag{2.20}$$

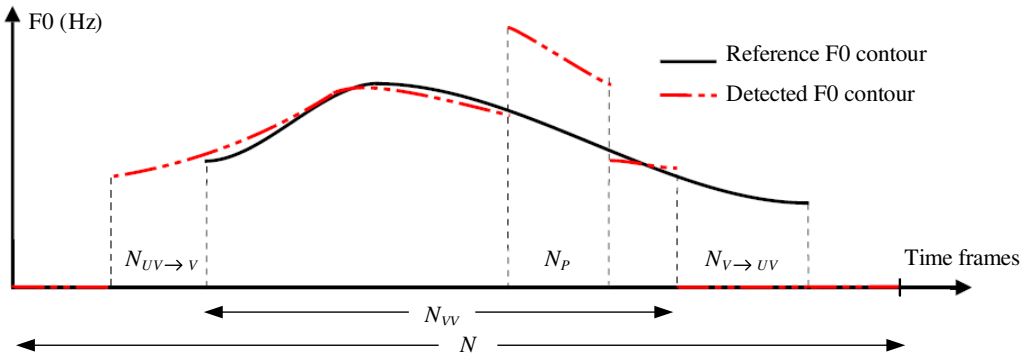

Figure 2.13: An illustration of frames with pitch detection errors in an utterance. N_P is the number of frames with gross pitch errors, $N_{V\rightarrow UV}$ is the number of voiced frames misclassified as unvoiced, and vice versa for $N_{UV\rightarrow V}$. N_{VV} is the number of reference voiced frames that is detected as voiced, while N is the total number of frames in the utterance.

$$PDE = \frac{N_{V\rightarrow UV} + N_{UV\rightarrow V} + N_P}{N} \times 100\% \tag{2.21}$$

For RAPT, SHR, and WWB, pitch detection performance evaluation is conducted using the algorithms' default settings. As for pitch estimation algorithms – YIN, SWIPE, and SWIPE', V/UV detection is performed using the same constant thresholding and median filtering scheme as that used in our proposed algorithm. The threshold level (varied in steps of 0.025) that gives the lowest averaged VDE for noisy speech in the Keele (dev) corpus is used. Table 2.2 shows the threshold levels tuned for these algorithms and the proposed MBSC PDA.

Table 2.2: Degree-of-voicing feature and the constant threshold level applied in Experiment 2

Algorithm	**Degree-of-voicing feature**	**Value**
YIN	Aperiodicity measure	0.425
SWIPE	Max. normalized inner product	0.175
SWIPE'	Max. normalized inner product	0.175
MBSC	Max. peak amplitude in $\tilde{r}_t^{mb}(\tau)$	0.375

2.3 Experiment 1: Pitch estimation performance evaluation

2.3.1 Results

The GPE_{refV} for the Keele (dev set) and CSTR corpora (eval set) are shown in this section. To reduce the amount of data presented, the GPE_{refV} results are averaged on a per noise-type or a per SNR basis. Table 2.3 contains the results for clean CSTR speech, while Tables 2.4 and 2.5 show the mean GPE_{refV} obtained by averaging on a per noise-type and a per SNR basis, respectively. The lowest value in each column is boldfaced to indicate the best-performing algorithm in each case. Note that the GPE_{refV} obtained for the eval and dev sets have similar trends. For the Keele (dev) corpus results, please refer to Tables 2.6 to 2.8.

SWIPE' has the lowest GPE_{refV} for fullband clean speech. For G.712 clean speech, WWB has the lowest GPE_{refV}. In these cases, the absolute difference between MBSC and the best performing algorithm is small, less than 0.4%.

For noisy speech, MBSC achieves the lowest mean GPE_{refV}, whether it is averaged over SNRs for each noise-type or averaged over noise-types at each

Table 2.3: Average GPE_{refV} (%) for clean CSTR (eval) corpus, assuming perfect V/UV detection. Boldfaced numbers indicate the best performance.

Algorithm	**Clean, fullband**	**Clean, G.712**
RAPT	9.63	10.4
YIN	4.02	7.76
SHR	3.78	11.0
SWIPE	3.69	10.2
SWIPE'	**3.49**	13.7
WWB	4.11	**4.19**
MBSC	3.83	4.50

SNR. A minimum overall GPE_{refV} of 7.85% and 9.66% are achieved for fullband and G.712-filtered noisy speech in the CSTR (eval) corpora, respectively.

2.3.2 Discussion

In general, GPE_{refV} is higher for the G.712-filtered data compared to its fullband version because of an increase in over-estimation (doubling/tripling) error, especially for low-pitched utterances whose fundamental harmonic is severely attenuated after G.712-filtering. The reverse trend is observed in some cases for the RAPT and WWB algorithms when the reductions in under-estimation errors for G.712-filtered, high-pitched utterances exceed increases in over-estimation errors for the low-pitched utterances.

The low GPE_{refV} achieved by SWIPE' and SWIPE for clean fullband speech shows that their comb-filter-based pitch estimation algorithm is effective in reducing harmonic and subharmonic errors, which are usually the main error contributors for clean speech. However, for the G.712 version, SWIPE' and SWIPE

Table 2.4: GPE_{refV} (%) averaged across SNRs from 20 to 0 dB per noise-type for noise-corrupted, fullband / G.712 CSTR (eval) corpus, assuming perfect V/UV detection.

Algorithm	**Babble**	**Car**	**Mach. Gun**	**Avg. All**
RAPT	34.2 / 31.1	11.3 / 12.7	25.5 / 19.2	23.7 / 21.0
YIN	18.9 / 22.6	9.68 / 16.4	12.0 / 15.2	13.5 / 18.1
SHR	16.7 / 24.9	6.35 / 17.5	11.0 / 17.5	11.4 / 19.9
SWIPE	19.3 / 24.4	7.41 / 27.0	11.3 / 18.6	12.7 / 23.3
SWIPE'	17.6 / 28.6	7.23 / 33.2	10.7 / 23.8	11.9 / 28.5
WWB	20.9 / 18.8	6.46 / 9.71	7.39 / 8.54	11.6 / 12.3
MBSC	**12.5** / **13.7**	**5.06** / **8.47**	**5.95** / **6.76**	**7.85** / **9.66**

Table 2.5: GPE_{refV} (%) averaged across noise-types per SNR for noise-corrupted, fullband / G.712 CSTR (eval) corpus, assuming perfect V/UV detection.

Algorithm	**20 dB**	**10 dB**	**0 dB**	**Avg. All**
RAPT	13.3 / 11.2	21.9 / 18.7	37.4 / 35.6	23.7 / 21.0
YIN	4.63 / 8.88	9.39 / 15.4	30.4 / 32.8	13.5 / 18.1
SHR	4.64 / 12.7	8.98 / 18.0	22.8 / 31.0	11.4 / 19.9
SWIPE	5.09 / 14.5	10.1 / 21.8	25.2 / 35.2	12.7 / 23.3
SWIPE'	4.73 / 19.7	8.99 / 27.7	24.5 / 39.1	11.9 / 28.5
WWB	5.63 / 6.18	9.71 / 9.70	21.1 / 23.7	11.6 / 12.3
MBSC	**4.16** / **5.34**	**6.00** / **8.24**	**15.1** / **17.1**	**7.85** / **9.66**

Table 2.6: Average GPE_{refV} (%) for clean Keele (dev) corpus, assuming perfect V/UV detection.

Algorithm	**Clean, fullband**	**Clean, G.712**
RAPT	5.08	5.32
YIN	3.04	6.83
SHR	2.18	8.18
SWIPE	2.09	8.46
SWIPE'	2.07	11.46
WWB	6.56	6.49
MBSC	**2.01**	**3.33**

Table 2.7: GPE_{refV} (%) averaged across SNRs from 20 to 0 dB per noise-type for noise-corrupted, fullband / G.712 Keele (dev) corpus, assuming perfect V/UV detection.

Algorithm	**Babble**	**Car**	**Mach. Gun**	**Avg. All**
RAPT	22.6 / 25.4	9.56 / 11.8	20.8 / 17.3	17.7 / 18.2
YIN	15.0 / 23.0	10.3 / 18.9	11.5 / 16.5	12.2 / 19.4
SHR	12.8 / 22.7	5.75 / 17.5	9.28 / 16.0	9.28 / 18.7
SWIPE	11.7 / 21.4	5.97 / 26.5	10.1 / 17.3	9.28 / 21.7
SWIPE'	11.1 / 25.9	6.15 / 32.9	9.52 / 21.8	8.92 / 26.9
WWB	15.6 / 14.1	7.95 / 10.4	9.38 / 10.0	11.0 / 11.5
MBSC	**8.29 / 11.9**	**3.54 / 7.44**	**3.85 / 5.83**	**5.23 / 8.39**

are not among the top three best-performing algorithms, and the performance of SWIPE' is worse than SWIPE. This is because SWIPE and SWIPE' use a decreasing amplitude comb-filter, so its performance degrades significantly when

Table 2.8: GPE_{refV} (%) averaged across noise-types per SNR for noise-corrupted, fullband / G.712 Keele (dev) corpus, assuming perfect V/UV detection.

Algorithm	**20 dB**	**10 dB**	**0 dB**	**Avg. All**
RAPT	7.80 / 7.76	15.3 / 15.6	32.0 / 33.6	17.7 / 18.2
YIN	3.70 / 8.44	8.46 / 16.1	28.5 / 36.9	12.2 / 19.4
SHR	2.97 / 10.2	6.83 / 16.5	20.5 / 31.8	9.28 / 18.7
SWIPE	2.96 / 11.9	6.97 / 19.6	20.4 / 35.6	9.28 / 21.7
SWIPE'	2.74 / 16.3	6.52 / 25.7	20.0 / 39.9	8.92 / 26.9
WWB	7.21 / 7.44	9.09 / 9.61	18.5 / 19.2	11.0 / 11.5
MBSC	**2.22** / **4.00**	**3.64** / **6.59**	**11.4** / **16.5**	**5.23** / **8.39**

the lower harmonics in the signal are missing or heavily attenuated in the G.712-filtered data. The degradation is more severe for SWIPE' that heavily relies on the first and prime harmonics (many non-prime harmonics of higher frequencies are ignored) in its comb-filter.

Although MBSC is not the best performing algorithm for clean speech in the eval set, it still has the lowest average GPE_{refV} at a high SNR of 20 dB, as shown in Table 2.5. MBSC's low GPE_{refV} under different noise-types and SNR levels shows that the proposed algorithm gives robust pitch estimation accuracy under a wide range of noise conditions. From Table 2.4, it is also observed that accurate pitch estimation is challenging for babble noise-corrupted speech for all the algorithms investigated. This is because babble noise also has a harmonic structure – it is difficult to estimate a correct pitch from the speech harmonics when an interfering set of harmonics is present.

Pitch estimation performance of the algorithms are also evaluated on frequency-shifted speech, i.e., harmonics are present at $n\,f_0+\delta$, where n is a positive integer,

Table 2.9: Average GPE_{refV} (%) for frequency-shifted, clean CSTR (eval) corpus, assuming perfect V/UV detection.

Algorithm	**Freq-shift, fullband**	**Freq-shift, G.712**
RAPT	58.29	54.95
YIN	59.65	56.98
SHR	56.23	58.69
SWIPE	58.89	61.35
SWIPE'	58.63	63.17
WWB	54.48	54.33
MBSC	**23.88**	**17.43**

f_0 is the pitch value, and δ is the amount of frequency offset. This frequency-shift phenomenon can occur due to a carrier frequency offset between the transmitter and receiver. It is commonly found in received speech of communication systems that use the single sideband suppressed carrier (SSB-SC) modulation scheme, which are popular for voice transmission in the high frequency (HF) radio spectrum by amateur radio, commercial, and military operators [89]. Frequency-shifted speech are generated from clean CSTR corpus, with and without G.712-filtering. The GPE_{refV} obtained by each algorithm is shown in Table 2.9. A frequency shift of δ = -120 Hz is chosen because this amount of frequency shift is found in some of the preliminary speech data recorded for the DARPA Robust Automatic Transcription of Speech (RATS) program [81, 90].

The comparative algorithms generally output a pitch estimate that is either the first positive frequency of the shifted harmonics, or the frequency of the strongest shifted harmonic. For WWB, its mid- and high-frequency subbands' ACRs are also computed from the envelopes of subband signals containing mul-

tiple harmonics. Thus, these ACRs contain information for accurate pitch estimation. However, ACRs computed from the individual shifted, low-frequency harmonics, (captured by the narrowband Gammatone filters at the low frequencies) provide pitch information that is inconsistent with the envelope ACRs. As a result, the pitch estimation accuracy of WWB is only slightly better than the other time-, or frequency-domain algorithms evaluated in this case.

The usage of signal envelopes in all subbands of the proposed MBSC-based algorithm leads to a significantly ($> 30\%$) lower GPE_{refV} in comparison to other algorithms, since the inter-harmonic frequency separation is invariant for frequency-shifted speech. Thus, the subband signals' amplitude modulation frequency will still be equal to true F0. However, the GPE_{refV} of clean, frequency-shifted data is not as good as that achieved for the normal clean speech. This is because for frames with F0 that is close to 240 Hz (twice the amount of frequency shift), shifting the spectrum down by 120 Hz results in short-time spectra with harmonics near odd multiples of 120 Hz. As such, the HSR of the low-frequency, non-envelope stream (i.e. s=0) will be high for the comb-channel with F_k near 120, and low for the comb-channel with F_k near the true F0 of 240 Hz. This produces a $r_{s=0,t}(\tau)$ with a strong peak near lag value, $\tau = \frac{f_s}{120}$, and a deep valley near the true pitch period of $\tau = \frac{f_s}{240}$. Although $r_{s,t}(\tau)$ for the envelope streams (s = 1, 2, 3, and 4), have their maximum peaks at $\tau \approx \frac{f_s}{240}$, they also have their second-strongest peak at twice this lag value, by nature of the AC function. Thus, the resulting MBSC would have a stronger peak at $\tau \approx \frac{f_s}{120}$, especially if $\alpha_t(s = 0)$ is larger than those of the other streams.

2.4 Experiment 2: Pitch detection performance evaluation

2.4.1 Results

Tables 2.10 to 2.18 present the GPE, VDE, and PDE obtained for the CSTR (eval) corpus, arranged according to the data characteristics and noise-types – Table 2.10 for clean speech, Tables 2.11 to 2.14 for 4-kHz fullband noisy speech, and Tables 2.15 to 2.18 for G.712-filtered noisy speech, respectively. The lowest value in each column is boldfaced. Note that similar trends in pitch detection performance are observed for the eval and dev sets. For pitch detection results of the Keele (dev) corpus, please refer to Tables 2.10 to 2.27.

For clean fullband and G.712-filtered speech in the eval set, it can be observed in Table 2.10 that RAPT has the lowest VDE and PDE.

For noisy speech in the eval set, although MBSC does not always have the lowest GPE and VDE among the algorithms at each individual noise-condition, it still has the lowest PDE because both its GPE and VDE are low. MBSC also has the lowest GPE, VDE and PDE when averaged over all noise-types investigated at each SNR (tabulated in Tables 2.14 and 2.18), thus it also has the best overall pitch detection performance (shown in the right column of these tables), for both fullband and G.712 noisy speech.

2.4.2 Discussion

For clean speech, MBSC has a slightly higher PDE than RAPT because the longer frame length defined in MBSC can cause an early onset and late offset detection of voiced segments. RAPT's good pitch detection performance for

clean speech is mainly attributed to its low VDE. The usage of a short 7.5 ms frame in RAPT produces sharper transitions at voicing boundaries. In addition, RAPT derives its voicing transition costs in its dynamic programming algorithm based on inter-frame energy ratio and spectral difference, which provide good indications of voicing transition boundaries for clean and some cases of high SNR noise conditions.

For noisy speech, MBSC's low overall PDE is attributed to its good pitch detection performance over various SNRs and noise-types. MBSC has the lowest average PDE at both high (20 dB) and low (0 dB) SNRs, which can be observed in Tables 2.14 and 2.18. Similarly, MBSC also has the lowest average PDE for each noise-type, as seen in the last column of these two tables. These results affirm that the pitch detection performance of the proposed MBSC is robust against a variety of noise conditions. This also shows that the proposed signal processing schemes are generally effective in enhancing MBSC's peak amplitude at the true pitch period, since low VDE is achievable with the simple constant V/UV threshold detection scheme.

For babble noise-corrupted speech, MBSC has a slightly higher VDE than RAPT because some of the harmonic-containing babble noise frames are erroneously enhanced and misclassified as voiced speech. However, MBSC still has the lowest PDE because of its higher pitch estimation accuracy for babble noise-corrupted speech compared to other algorithms. In the case of car noise-corrupted speech, some comparative algorithms have lower GPEs than MBSC because there is a higher proportion of frames with a pitch estimation error for the additional weakly voiced frames detected by MBSC, but missed by the other algorithms (thus the VDEs of these algorithms are higher than MBSC's). Machine gun noise is highly non-stationary, such that there is a large variation in SNR within

each machine gun noise-corrupted speech utterance. Since the MBSC peak enhancement schemes improve the algorithm's noise-robustness under a wide range of SNRs, MBSC gives superior performance in GPE, VDE and PDE compared to other algorithms for machine gun noise-corrupted speech at all SNR levels.

Table 2.10: GPE, VDE, and PDE (%) for clean CSTR (eval) corpus.

Algorithm	**Clean, Fullband**			**Clean, G.712**		
	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.55	**4.99**	**5.93**	2.86	**5.70**	**6.75**
YIN	1.98	7.38	8.10	4.86	7.97	9.79
SHR	2.27	7.95	8.81	9.43	9.32	13.1
SWIPE	2.00	6.18	6.93	7.46	9.05	11.8
SWIPE'	1.95	7.01	7.75	10.7	9.85	13.9
WWB	1.91	8.13	8.82	1.93	8.25	8.95
MBSC	**1.85**	5.82	6.52	**1.90**	6.67	7.37

Table 2.11: GPE, VDE, and PDE (%) for babble noise-corrupted, fullband CSTR (eval) corpus.

Babble noise-corrupted CSTR, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.62	8.12	9.05	7.95	**12.3**	14.9	35.9	**25.7**	33.8	13.7	**14.6**	18.2
YIN	1.84	9.60	10.3	5.90	14.1	16.0	28.7	28.9	34.6	10.8	16.7	19.4
SHR	2.82	**7.76**	8.78	6.09	12.8	14.7	16.0	30.1	33.0	7.70	16.0	8.78
SWIPE	2.75	12.6	13.6	8.18	18.0	20.6	30.2	33.3	39.3	12.3	20.6	23.6
SWIPE'	2.41	13.5	14.4	5.90	18.8	20.7	26.4	35.6	40.3	10.2	21.7	24.1
WWB	4.78	14.3	15.9	9.18	17.9	20.7	21.2	28.7	33.7	11.0	19.7	22.8
MBSC	**1.71**	7.98	**8.61**	**2.48**	12.4	**13.1**	**9.30**	26.8	**28.5**	**3.98**	14.9	**15.8**

Table 2.12: GPE, VDE, and PDE (%) for car noise-corrupted, fullband CSTR (eval) corpus.

Car noise-corrupted CSTR, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.17	4.68	5.46	1.59	6.36	6.90	1.73	15.5	16.0	3.35	14.3	15.0
YIN	1.57	4.97	5.53	1.18	8.04	8.43	3.40	22.2	22.9	1.75	10.9	11.4
SHR	2.19	7.92	8.76	2.25	12.3	13.0	3.61	26.2	27.2	2.51	14.8	15.6
SWIPE	1.33	4.79	5.27	0.73	6.86	7.09	0.91	16.7	16.9	0.92	8.83	9.13
SWIPE'	**1.24**	4.82	5.26	**0.65**	7.31	7.52	**0.58**	18.3	18.4	**0.75**	9.53	9.77
WWB	1.95	7.38	8.07	1.82	7.26	7.88	2.86	**8.36**	9.27	2.10	7.53	8.24
MBSC	1.70	**4.57**	**5.21**	1.54	**4.98**	**5.54**	1.35	8.55	**8.99**	1.54	**5.76**	**6.31**

Table 2.13: GPE, VDE, and PDE (%) for machine gun noise-corrupted, fullband CSTR (eval) corpus.

Machine gun noise-corrupted CSTR, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	3.80	8.86	10.3	10.2	11.0	14.7	19.1	16.3	22.4	10.9	11.7	15.5
YIN	2.04	7.48	8.21	3.76	10.6	11.9	7.03	18.5	20.4	4.04	11.9	13.1
SHR	3.29	12.1	13.3	7.50	21.8	24.5	18.8	31.6	36.6	17.0	26.0	30.1
SWIPE	2.43	6.04	6.94	3.81	9.28	10.6	3.95	16.9	17.9	3.43	10.4	11.5
SWIPE'	2.12	6.69	7.47	2.61	10.7	11.6	3.00	18.7	19.5	2.57	11.7	12.6
WWB	1.97	8.34	9.02	2.07	9.75	10.5	3.47	12.5	13.6	2.35	10.1	10.9
MBSC	**1.74**	**5.19**	**5.84**	**1.69**	**6.39**	**7.00**	**1.75**	**10.4**	**10.9**	**1.72**	**7.12**	**7.72**

Table 2.14: Averaged GPE, VDE, and PDE (%) obtained for noisy fullband CSTR (eval) corpus.

CSTR, Fullband - Avg. across noise-types												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.86	7.22	8.25	6.58	9.88	12.2	18.9	19.2	24.1	8.79	11.5	14.1
YIN	1.82	7.35	8.00	3.62	10.9	12.1	13.1	23.2	26.0	5.53	13.2	14.6
SHR	2.77	9.26	10.3	5.28	15.6	17.4	12.8	29.3	32.3	9.06	18.9	21.2
SWIPE	2.17	7.82	8.61	4.24	11.4	12.7	11.7	22.3	24.7	5.55	13.3	14.8
SWIPE'	1.92	8.34	9.03	3.05	12.3	13.2	9.98	24.2	26.1	4.49	14.3	15.5
WWB	2.90	10.0	11.0	4.36	11.7	13.0	9.17	16.5	18.9	5.16	12.5	14.0
MBSC	**1.71**	**5.91**	**6.55**	**1.90**	**7.91**	**8.55**	**4.13**	**15.2**	**16.1**	**2.41**	**9.24**	**9.96**

Table 2.15: GPE, VDE, and PDE (%) for babble noise-corrupted, G.712 CSTR (eval) corpus.

Babble noise-corrupted CSTR, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.59	10.6	11.5	6.73	15.2	17.4	28.4	28.3	35.0	11.4	17.4	20.5
YIN	4.58	11.9	13.6	9.36	17.1	20.2	26.9	30.3	36.0	12.5	19.0	22.5
SHR	11.0	11.2	15.5	13.0	15.5	19.8	20.2	30.5	34.1	14.2	18.3	22.4
SWIPE	9.33	13.7	17.1	13.9	18.7	23.2	24.9	31.2	36.3	15.3	20.7	25.0
SWIPE'	13.2	15.5	20.3	19.5	20.7	26.8	31.5	32.5	38.6	20.8	22.4	28.1
WWB	3.62	14.0	15.1	7.88	18.7	21.1	20.5	30.5	35.1	10.1	20.3	22.9
MBSC	**2.06**	**7.75**	**8.49**	**2.55**	**13.2**	**14.0**	**4.75**	**26.4**	**27.3**	**2.96**	**15.1**	**15.9**

Table 2.16: GPE, VDE, and PDE (%) for car noise-corrupted, G.712 CSTR (eval) corpus.

Car noise-corrupted CSTR, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.76	**5.03**	6.03	2.18	7.45	8.13	**1.32**	18.3	18.5	**2.03**	9.58	10.2
YIN	4.68	5.64	7.33	5.24	9.60	11.3	3.24	22.7	23.2	4.34	11.9	13.2
SHR	9.93	11.0	14.9	9.84	15.2	18.7	9.78	28.5	30.8	9.86	17.6	20.9
SWIPE	11.4	7.22	11.0	12.9	12.7	15.8	9.43	24.2	25.0	11.5	14.2	16.9
SWIPE'	18.7	8.00	13.9	21.0	14.0	18.3	16.9	25.3	26.4	19.3	15.3	19.2
WWB	**1.61**	6.92	7.48	**2.06**	8.54	9.21	3.45	15.1	16.0	2.35	9.80	10.5
MBSC	1.97	5.06	**5.77**	3.19	**6.67**	**7.78**	2.10	**13.4**	**13.9**	2.55	**7.92**	**8.75**

Table 2.17: GPE, VDE, and PDE (%) for machine gun noise-corrupted, G.712 CSTR (eval) corpus.

Machine gun noise-corrupted CSTR, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	3.78	18.6	20.0	9.02	22.1	25.5	20.2	27.2	34.7	10.4	22.5	26.4
YIN	5.17	10.6	12.5	7.21	13.7	16.2	9.79	18.8	21.7	7.37	14.3	16.7
SHR	10.3	18.3	22.4	12.7	24.1	28.8	22.5	31.3	37.4	14.5	24.4	29.4
SWIPE	8.42	9.80	12.8	9.96	12.4	15.7	11.2	17.1	20.1	9.81	13.0	16.1
SWIPE'	13.0	11.3	15.9	14.9	14.5	19.1	16.0	19.4	23.4	14.7	14.9	19.4
WWB	1.93	8.78	9.47	2.22	10.6	11.3	3.51	13.6	14.7	2.46	10.9	11.7
MBSC	**1.92**	**6.20**	**6.89**	**1.71**	**8.32**	**8.92**	**1.60**	**11.5**	**12.0**	**1.73**	**8.58**	**9.18**

Table 2.18: Average GPE, VDE, and PDE (%) for noise-corrupted, G.712 CSTR (eval) corpus.

CSTR, G.712 - Avg. across noise-types												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	3.04	11.4	12.5	5.98	14.9	17.0	16.6	24.6	29.4	7.95	16.5	19.1
YIN	4.81	9.37	11.1	7.27	13.4	15.9	3.24	22.7	23.2	8.07	15.1	17.5
SHR	10.4	13.5	17.6	11.8	18.3	22.4	17.5	30.1	34.1	12.9	20.1	24.2
SWIPE	9.71	10.2	13.6	12.2	14.6	18.2	15.2	24.2	27.2	12.2	16.0	19.3
SWIPE'	15.0	11.6	16.7	18.5	16.4	21.4	21.5	25.8	29.5	18.3	17.5	22.2
WWB	2.38	9.89	10.7	4.05	12.6	13.9	9.15	19.7	21.9	4.96	13.7	15.1
MBSC	**1.99**	**6.34**	**7.05**	**2.48**	**9.39**	**10.2**	**2.81**	**17.1**	**17.7**	**2.41**	**10.5**	**11.3**

Table 2.19: Average GPE, VDE, and PDE (%) for clean Keele (dev) corpus.

Algorithm	**Clean, Fullband**			**Clean, G.712**		
	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.24	**3.95**	**5.07**	2.74	**4.94**	**6.28**
YIN	1.67	4.54	5.34	3.98	6.15	7.94
SHR	1.97	13.2	14.2	7.59	13.4	17.1
SWIPE	1.05	5.19	5.70	4.61	8.41	10.4
SWIPE'	**1.02**	5.38	5.89	7.13	9.55	12.5
WWB	2.91	8.92	10.2	3.03	9.06	10.4
MBSC	1.19	4.97	5.55	**1.59**	5.56	**6.28**

Table 2.20: GPE, VDE, and PDE (%) for babble noise-corrupted, fullband Keele (dev) corpus.

Babble noise-corrupted Keele, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	1.72	**5.30**	**6.13**	5.15	**10.5**	12.8	22.2	**29.1**	36.0	8.68	**13.8**	17.0
YIN	1.62	6.71	7.46	4.90	12.8	15.0	21.0	32.5	38.1	8.00	16.3	19.0
SHR	2.55	14.6	15.9	5.88	16.2	19.0	19.8	30.3	37.1	8.52	19.3	22.8
SWIPE	1.08	7.64	8.16	3.30	13.7	15.1	16.4	32.8	37.3	5.96	16.9	19.0
SWIPE'	**0.99**	7.98	8.46	2.62	14.4	15.5	15.0	34.1	37.8	5.26	17.8	19.4
WWB	2.82	12.3	13.5	4.59	17.4	19.3	15.7	31.2	36.0	6.83	19.6	22.1
MBSC	1.16	6.60	7.15	**1.42**	11.2	**11.8**	**5.29**	30.3	**31.7**	**2.27**	14.9	**15.7**

Table 2.21: GPE, VDE, and PDE (%) for car noise-corrupted, fullband Keele (dev) corpus.

Car noise-corrupted Keele, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	1.75	**4.25**	**5.11**	1.28	7.35	7.92	2.21	21.0	21.6	1.65	10.0	10.7
YIN	1.61	5.00	5.74	1.70	10.7	11.4	4.45	30.3	31.3	2.31	14.2	15.0
SHR	2.06	11.4	12.4	2.07	18.9	19.8	3.17	39.1	40.1	2.37	22.3	23.3
SWIPE	0.79	5.49	5.87	0.49	9.05	9.26	0.52	21.5	21.7	0.57	11.3	11.6
SWIPE'	**0.78**	5.65	6.03	**0.36**	10.0	10.2	**0.25**	23.9	23.9	**0.47**	12.4	12.6
WWB	2.68	8.48	9.67	2.44	8.49	9.54	2.15	10.4	11.3	2.44	8.98	10.0
MBSC	1.15	4.70	5.26	1.18	**5.70**	**6.24**	1.26	**10.2**	**10.7**	1.21	**6.58**	**7.12**

Table 2.22: GPE, VDE, and PDE (%) for machine gun noise-corrupted, fullband Keele (dev) corpus.

Machine gun noise-corrupted Keele, Fullband												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	3.15	**5.07**	6.64	9.67	8.38	13.1	18.2	16.0	23.9	10.3	9.43	14.2
YIN	1.91	5.53	6.41	2.79	10.2	11.4	6.69	19.7	22.1	3.68	11.4	12.9
SHR	3.05	17.5	19.0	6.78	24.1	27.4	17.7	33.4	40.4	8.56	24.7	28.5
SWIPE	1.46	6.05	6.76	2.87	10.1	11.4	3.58	18.8	20.1	2.64	11.3	12.4
SWIPE'	**1.11**	6.34	6.88	1.64	11.1	11.8	1.88	20.7	21.3	1.52	12.3	13.0
WWB	2.76	9.20	10.4	2.82	11.0	12.2	3.46	14.9	16.2	2.97	11.6	12.8
MBSC	1.18	5.16	**5.71**	**1.20**	**6.88**	**7.42**	**1.25**	**11.2**	**11.7**	**1.20**	**7.51**	**8.04**

Table 2.23: Averaged GPE, VDE, and PDE (%) for noise-corrupted, fullband Keele (dev) corpus.

Keele, Fullband - Avg. across all four noise-types												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.20	**4.87**	**5.96**	5.37	8.74	11.3	14.2	22.0	27.1	6.87	11.1	14.0
YIN	1.71	5.75	6.54	3.13	11.2	12.6	10.7	27.5	30.5	4.66	14.0	15.6
SHR	2.55	14.5	15.8	4.91	19.7	22.1	13.6	34.3	39.2	6.48	22.1	24.9
SWIPE	1.11	6.39	6.93	2.22	10.9	11.9	6.83	24.4	26.4	3.05	13.2	14.3
SWIPE'	**0.96**	6.66	7.12	1.54	11.8	12.5	5.72	26.2	27.7	2.42	14.2	15.0
WWB	2.75	9.98	11.2	3.29	12.3	13.7	7.10	18.9	21.2	4.08	13.4	15.0
MBSC	1.17	5.49	6.04	**1.27**	**7.92**	**8.48**	**2.60**	**17.2**	**18.0**	**1.56**	**9.67**	**10.3**

Table 2.24: GPE, VDE, and PDE % obtained for babble noise-corrupted, G.712-filtered Keele (dev) corpus.

Babble noise-corrupted Keele, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.55	**7.69**	8.89	6.81	**13.7**	16.7	24.8	**30.7**	38.5	10.3	**16.7**	20.5
YIN	4.04	10.5	12.2	9.41	17.2	21.0	27.4	34.2	41.2	12.5	19.9	24.0
SHR	9.25	17.2	21.6	14.8	19.8	26.4	29.8	32.4	41.7	17.1	22.3	29.0
SWIPE	5.04	12.2	14.3	8.33	19.4	22.3	24.0	35.1	39.8	11.35	21.6	24.7
SWIPE'	8.40	14.3	17.6	14.2	22.3	26.5	31.4	36.6	42.4	17.1	23.9	28.3
WWB	2.66	12.3	13.5	3.70	17.2	18.7	13.8	31.5	35.6	5.95	19.7	21.8
MBSC	**1.49**	7.70	**8.34**	**2.00**	13.8	**14.5**	**5.23**	31.9	**33.1**	**2.56**	16.9	**17.7**

Table 2.25: GPE, VDE, and PDE % obtained for car noise-corrupted, G.712-filtered Keele (dev) corpus.

Car noise-corrupted Keele, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.05	**5.46**	**6.42**	1.69	9.89	10.6	**1.54**	25.5	25.8	**1.64**	12.7	13.3
YIN	3.71	7.02	8.60	4.20	13.7	15.1	5.42	31.4	32.3	4.36	16.5	17.8
SHR	7.96	13.7	17.3	9.02	22.7	26.0	10.9	41.4	43.8	9.10	25.3	28.4
SWIPE	5.99	10.4	12.5	7.35	18.7	20.4	10.6	33.8	34.4	7.76	20.4	21.8
SWIPE'	10.5	12.7	15.8	15.8	21.0	23.4	18.2	35.0	35.7	15.1	22.4	24.6
WWB	2.32	8.50	9.47	2.05	10.8	11.6	1.92	18.0	18.7	2.09	12.0	12.8
MBSC	**1.60**	5.75	6.45	**1.63**	**8.51**	**9.16**	1.92	**16.8**	**17.4**	1.74	**9.84**	**10.5**

Table 2.26: GPE, VDE, and PDE % obtained for machine gun noise-corrupted, G.712-filtered Keele (dev) corpus.

Machine gun noise-corrupted Keele, G.712												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	3.87	10.8	12.7	10.3	15.7	20.9	22.2	22.9	33.7	11.6	16.3	22.1
YIN	4.70	8.80	10.9	6.75	13.3	16.1	11.6	20.9	25.0	7.55	14.0	17.0
SHR	8.91	19.0	23.2	12.7	24.9	30.7	24.6	34.2	42.8	14.6	25.8	31.9
SWIPE	4.95	10.0	12.0	6.47	13.8	16.2	7.87	20.9	23.2	6.46	14.7	16.9
SWIPE'	8.29	11.9	15.0	10.8	16.3	19.8	12.6	23.2	26.6	10.6	16.9	20.3
WWB	2.92	9.68	10.9	2.76	12.2	13.3	3.80	16.4	17.7	3.05	12.7	13.9
MBSC	**1.44**	**6.16**	**6.79**	**1.38**	**8.39**	**8.97**	**1.45**	**12.8**	**13.3**	**1.41**	**8.92**	**9.51**

Table 2.27: Average GPE, VDE, and PDE % obtained for noise-corrupted, G.712-filtered Keele (dev) corpus.

Keele, G.712 - Avg. across noise-types												
SNR (dB)	**20**			**10**			**0**			**Avg. 20 to 0**		
Algo.	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE	GPE	VDE	PDE
RAPT	2.82	7.97	9.33	6.28	13.1	16.1	16.2	26.4	32.7	7.84	15.2	18.6
YIN	4.15	8.76	10.6	6.79	14.7	17.4	14.8	28.8	32.8	8.14	16.8	19.6
SHR	8.71	16.6	20.7	12.2	22.5	27.7	21.8	36.0	42.8	13.6	24.5	29.8
SWIPE	5.33	10.9	12.9	7.39	17.3	19.6	14.2	29.9	32.5	8.52	18.9	21.2
SWIPE'	9.07	13.0	16.1	13.6	19.8	23.2	20.7	31.6	34.9	14.3	21.1	24.4
WWB	2.63	10.2	11.3	2.84	13.4	14.5	6.50	22.0	24.0	3.70	14.8	16.2
MBSC	**1.51**	**6.54**	**7.20**	**1.67**	**10.2**	**10.9**	**2.87**	**20.5**	**21.3**	**1.90**	**11.9**	**12.6**

Besides being a robust feature for V/UV detection in a pitch detection application, the maximum MBSC peak amplitude is also a reliable frame-level degree-of-voicing feature that can be used to enhance the noise-robustness of other speech applications. One such application is speech activity detection (SAD). A speed-optimized version of the proposed MBSC PDA is one of the algorithms used in SRI's SAD system for the DARPA Robust Automatic Transcription of Speech (RATS) project [91]. The RATS data was collected by the Linguistic Data Consortium (LDC) by retransmitting conversational telephone speech through eight different communication channels using multiple signal transmitters/transceivers, listening station receivers, and signal collection and digitization apparatus [81]. The RATS rebroadcasted data contain a wide array of real transmission distortions, including band limitation, strong channel noises, nonlinear speech distortions (e.g., clipping), frequency shifts, high energy burst in non-transmission time intervals. One-dimensional discrete cosine transform (DCT) is applied to a long-term window containing 30 MBSC's maximum peak amplitudes from consecutive time frames, and the first 4 DCT coefficients are used to build the speech and non-speech Gaussian mixture models (GMMs). Table 2.28 shows the equal error rate (EER) (probability of missed speech detection = probability of false speech detection) for different input features on the RATS SAD Dev-1 and Dev-2 sets, which is published in [91]. MBSC has a higher EER for the Dev-2 set because there is a significant amount of non-speech tonal interferences in Dev-2. Tonal interference causes false speech detection because the proposed algorithm does not differentiate between signal periodicity in speech and non-speech. However, when MBSC features are combined with MFCCs, which capture information regarding the spectral shapes of speech signals, false alarms caused by tonal interferences are suppressed, since their spectral shapes do not resemble speech. This MFCC+MBSC feature combination also has the lowest EERs for Dev-1

Table 2.28: EER (%) for RATS SAD Dev-1 and Dev-2 sets obtained by SRI's SAD system.

Features	Dev-1	Dev-2
MBSC	4.10	6.15
MFCC	2.05	2.70
MFCC+MBSC	1.65	2.45

and Dev-2.

2.5 Conclusion

The proposed MBSC PDA involves several algorithmic novelties to improve the robustness of its pitch detection performance. For frequency decomposition, four wideband FIR filters equally-spaced in the linear frequency scale are used. The low-frequency wideband FIR filter in the MBSC PDA facilitates the use of signal envelope in this band, which contributes to a higher pitch estimation accuracy for bandpass-filtered speech. The FIR filters have equal separation in the linear frequency range so that there is no bias towards harmonics in particular frequency ranges. This helps avoid a significant performance degradation when several harmonics in a particular frequency band are masked by noise. Besides the Hilbert signal envelope of each subband, the non-envelope signal stream from the low-frequency subband is also used. Multi-channel comb-filtering is performed separately for each subband stream. This is in contrast to other comb-filter-based algorithms that apply comb-functions spanning the full spectrum, making them highly dependent on prominent low-frequency harmonics to perform well, especially if the comb-functions have decreasing amplitude with frequency (e.g. [8,80]).

To derive the peak-enhanced MBSC, an SC is first computed for each stream using an HSR-weighted-average of the ACRs computed from comb-filtered outputs of selected channels. These SC streams are further fused to form the MBSC based on their within-stream and between-stream reliability factors. Together, the proposed signal processing schemes (subband multi-channel comb-filtering, HSR-based channel-selection-and-weighting, stream-reliability-weighting) helps to enhance the maximum MBSC peak at the most likely pitch period, which improves the accuracy of pitch estimation, as well as V/UV detection. The variability of the maximum MBSC peak amplitude with SNRs is reduced, such that robust V/UV detection is achieved by simply applying a constant threshold on this single feature, followed by median filtering – without requiring additional features [6], a separate V/UV detection module, or a pitch continuity tracking algorithm involving dynamic programming [3], or statistically-trained, data-driven modeling techniques [13]. To ascertain that the threshold is not highly data-dependent, our evaluations involve separate development and evaluation data sets.

When the pitch estimation accuracy is evaluated on every voiced frame (assuming perfect V/UV detection) in Experiment 1, noise-robustness of the proposed algorithm's pitch estimation accuracy is shown through the low average gross pitch error (GPE_{refV}) achieved for noisy speech at various SNRs and noise-types.

When pitch detection performance is assessed based on both pitch estimation and voicing detection accuracies in Experiment 2, the MBSC-based pitch detector has the lowest average pitch detection error (PDE) under various SNRs and noise types. This shows that the novel peak-enhancement schemes are generally effective in boosting the MBSC's peak at the true pitch period, since the sim-

ple constant threshold V/UV detection scheme still yields good pitch detection performances for many of the noise conditions.

CHAPTER 3

Feature enhancement using jointly-sparse reference and estimated soft-mask representations for noise-robust speech recognition

In this chapter, a feature enhancement technique for noise-robust speech recognition using jointly-sparse reference soft-mask (SM_{ref}) and estimated soft-mask (SM_{est}) representations is proposed. Dictionary learning is used to derive the jointly-sparse SM_{ref} and SM_{est} dictionaries from soft-mask exemplar-pairs extracted from the training set. By solving an l_1-minimization problem, the sparse linear combination of SM_{est} representations that best resembles the target utterance's SM_{est} is found. The same sparse linear combination is applied to the SM_{ref} dictionary representations to obtain an enhanced soft-mask for denoising the target utterance's Mel-spectrogram before MFCC extraction. The proposed technique is evaluated against other sparse representation-based feature enhancement techniques on the Aurora-2 database – a noisy digit speech corpus, and the Aurora-4 database – a 5000-word vocabulary noisy speech corpus. For the Aurora-4, additional noisy test sets produced by mixing other noise types to Aurora-4's clean test data to evaluate the algorithms' performance on unseen noise types. This chapter is based on L. N. Tan and A. Alwan, "Feature en-

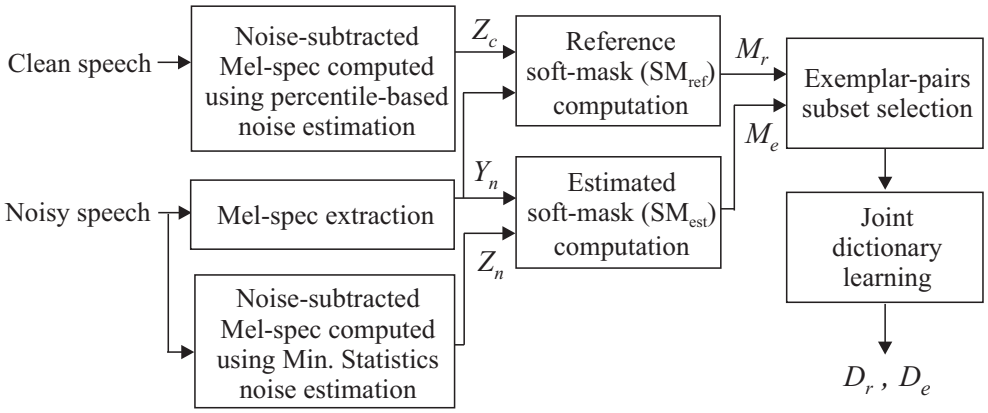

Figure 3.1: Joint soft-mask dictionary learning (training phase)

hancement using sparse reference and estimated soft-mask exemplar-pairs for noisy speech recognition," in *Proc. ICASSP*, 2014, pp. 1729-1733.

3.1 Proposed JDictMask feature enhancement algorithm

3.1.1 Joint soft-mask dictionary learning

Fig. 3.1 summarizes the joint soft-mask dictionary learning scheme. First, $Y_c[f,t]$ and $Y_n[f,t]$, the Mel-frequency magnitude spectrogram (Mel-spectrogram) of the clean and noisy versions of each training utterance-pair, are computed by applying Mel-filter weighting on the respective pre-emphasized short-time FFT magnitude spectrum in each frame. Each time-frequency (T-F) unit in the spectrogram is denoted by two indices, f and t, which indicate the Mel-frequency bin and time frame, respectively. The SM_{ref} at each T-F unit, $M_r[f,t]$, is computed by taking the ratio of $Z_c[f,t]$ – a noise-subtracted version of $Y_c[f,t]$ – to $Y_n[f,t]$, as shown in Eq. (3.1). Z_c is calculated by subtracting the ambient noise Mel-spectrogram, N_c, from Y_c, with negative values set to 0, as shown in Eq. (3.3). N_c has a constant noise spectral value in time at each frequency bin, and these constant values are

the 25th percentile of individual frequency bin's Mel-spectrogram magnitudes found in the first and last 20 time frames of Y_c. Similarly, the $\mathrm{SM}_{\mathrm{est}}$, denoted by $M_e[f,t]$, is computed by taking the ratio of Z_n to Y_n (Eq. (3.2)), where Z_n is the noise-subtracted version of Y_n. The noise Mel-spectrogram, N_n in Eq. (3.4), is estimated using the minimum statistics (MS) noise estimation technique [92] implemented in the "estnoisem" function of Voicebox [93]. The MS technique is not used to estimate N_c because some of the speech energy tends to leak into the estimated noise spectra, resulting in over-estimation of the ambient noise.

$$M_r[f,t] = \min(Z_c[f,t]/Y_n[f,t], 1) \tag{3.1}$$

$$M_e[f,t] = Z_n[f,t]/Y_n[f,t] \tag{3.2}$$

$$Z_c[f,t] = \max(Y_c[f,t] - N_c[f,t], 0) \tag{3.3}$$

$$Z_n[f,t] = \max(Y_n[f,t] - N_n[f,t], 0) \tag{3.4}$$

Fig. 3.2 shows the $\mathrm{SM}_{\mathrm{ref}}$ computed with and without ambient noise subtraction on a clean training utterance. In the presence of a low-frequency noise, the high-frequency energy in the noisy spectra is very similar to that in the clean spectra. Thus, the $\mathrm{SM}_{\mathrm{ref}}$ computed without ambient noise subtraction has values close to 1 at the high Mel-frequency bins (Fig. 3.2a), resulting in poor high-frequency noise suppression at non-speech regions. With ambient noise subtraction (Eq. 3.4), the resulting $\mathrm{SM}_{\mathrm{ref}}$ have lower values at non-speech regions across all frequencies (Fig. 3.2b), and hence better noise suppression across all frequencies. Ambient noise subtraction is also helpful in generating a $\mathrm{SM}_{\mathrm{ref}}$ with good noise suppression property when the clean training utterances contain a consistent low-energy background noise (e.g. hum or tones) from the recording equipment.

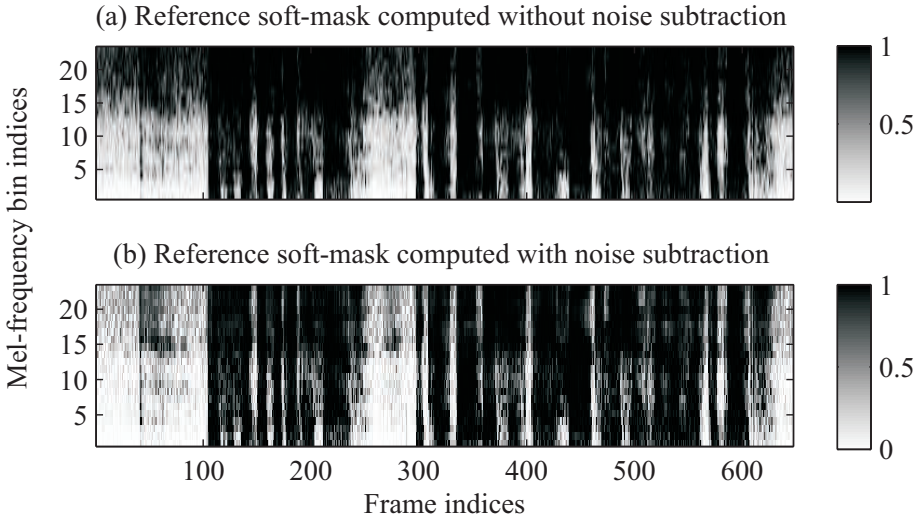

Figure 3.2: Reference soft-mask, SM_{ref}, computed with/without ambient noise subtraction in the clean training utterance

To avoid memory overload, a subset of randomly selected soft-mask exemplar-pairs from the training utterances is used to learn jointly-sparse SM_{ref} and SM_{est} dictionaries. This soft-mask exemplar-pair subset is obtained as follows. Soft-mask exemplar-pairs (m_r^i, m_e^i) are extracted from the same T-F region of each clean and noisy training utterance-pair. Note that clean and noisy training utterance-pairs required for these soft-mask exemplar-pairs extraction can be easily generated by adding noise to clean training data. Each i-th exemplar, $m_*^i \in \mathbb{R}^{FT\times 1}$ is formed by concatenating the columns in $M_*[1{:}F, \tilde{t}{:}(\tilde{t}+\text{T-1})] \in \mathbb{R}^{F\times T}$ that contains F=23 Mel-frequency bins, and T=11 consecutive frames (covers about one phoneme interval). The notation $a{:}b$ represents the range of integers $\{a, a+1, \ldots, b\}$. T=11 is also used in [48], and the feature enhancement scheme in [47] reported good performance with T=10. Eight such exemplar-pairs are obtained by randomly selecting $\tilde{t}$ – the starting frame of the region. To enable *joint* learning, each (m_r^i, m_e^i) exemplar-pair is concatenated to form $m^i \in \mathbb{R}^{2FT\times 1}$ (see Eq. (3.5)), which serves as the input to the dictionary learning algorithm.

Table 3.1: MexTrainDL parameters for dictionary learning

param.K = 4000	param.posAlpha = 1
param.lambda = 0.15	param.posD = 1
param.batchsize = 10000	param.whiten = 0
param.mode = 2	param.iter = 100

The joint dictionary, $D \in \mathbb{R}^{2FT \times L}$, is learned using the *mexTrainDL* function in the Dictionary Learning and Matrix Factorization toolbox of the SPArse Modeling Software (SPAMS) [94, 95], which solves the optimization problem in Eq. (3.6). The *mexTrainDL* parameters listed in Table 3.1 are used, which resulted in a joint D containing $L = 4000$ representations. A dictionary of this size was used in our preliminary experiment on the Aurora-2 noisy digit recognition task [96], and it also gives reasonable performance for the Aurora-4 database. The SM_{ref} and SM_{est} dictionaries, $\tilde{D}_r$ and $\tilde{D}_e$, (each$\in \mathbb{R}^{FT \times L}$) are subsequently obtained by extracting the relevant rows in D, as shown in Eq. (3.7). The final SM_{ref} and SM_{est} dictionaries that are used during feature enhancement, D_r and D_e, are computed in Eqs. (3.8) and (3.9). They are obtained by normalizing the columns in both $\tilde{D}_e$ and $\tilde{D}_r$ by the L2-norm of the corresponding $\tilde{d}_e$ column in $\tilde{D}_e$. This ensures that all columns in D_e (on which the l_1-minimization optimization is performed during feature enhancement) have unit norm.

$$m^i = \begin{bmatrix} m_r^i \\ m_e^i \end{bmatrix} \tag{3.5}$$

$$\begin{gathered} \min_D \frac{1}{n} \sum_{i=1}^{n} 0.5 ||m^i - D\alpha^i||_2^2 + \lambda ||\alpha^i||_1, \\ \text{such that } \alpha^i \geq 0,\ d^j \geq 0,\ ||d^j||_2^2 \leq 1 \end{gathered} \tag{3.6}$$

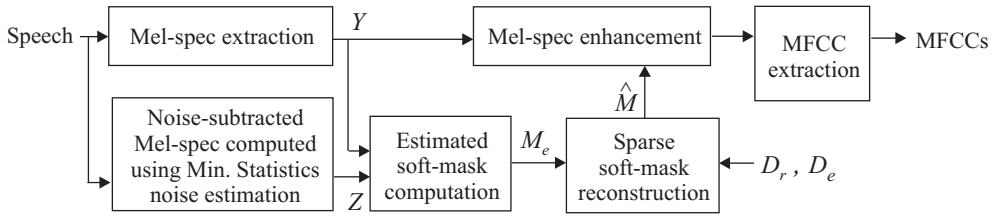

Figure 3.3: The proposed feature enhancement algorithm using jointly-sparse soft-mask dictionaries (during MFCC extraction in both training and testing phases)

$$D = [d^1, d^2, ..., d^{4000}] = \begin{bmatrix} \tilde{D}_r \\ \tilde{D}_e \end{bmatrix} \tag{3.7}$$

$$D_e = [d_e^1, d_e^2, ..., d_e^{4000}], \text{ where } d_e^i = \tilde{d}_e^i / ||\tilde{d}_e^i||_2 \tag{3.8}$$

$$D_r = [d_r^1, d_r^2, ..., d_r^{4000}], \text{ where } d_r^i = \tilde{d}_r^i / ||\tilde{d}_e^i||_2 \tag{3.9}$$

3.1.2 Feature enhancement using learned soft-mask dictionaries

The proposed feature enhancement technique is illustrated in Fig. 3.3. The SM_{est} of the input utterance, M_e, is computed as described in Section 3.1.1, and a SM_{est} vector, m_e^j, is derived by concatenating the columns in each $M_e[1{:}F,\ j{:}(j{+}\text{T-}1)]$ region, for $j = 1, 2, \ldots$. By solving the Lasso l_1-minimization problem [97] in Eq. (3.10), x^j, the sparse linear combination of D_e entries that best approximates each m_e^j is found. The *SolveLasso* function in the SparseLab toolbox [98] is used to perform "nnlasso" (non-negative Lasso) with a maximum of 50 iterations, and with $\lambda_{\text{stop}} = 0.15$. The value of λ in Eq. (3.10) is iteratively updated in SparseLab's Lasso implementation, and the algorithm terminates when $\lambda \leq \lambda_{\text{stop}}$ or the maximum iteration is reached. The enhanced soft-mask vector, $\hat{m}^j$ is then reconstructed by multiplying the same sparse vector, x^j to D_r as shown

in Eq. (3.11). Each $\hat{m}^j$ is shaped back into rectangular T-F regions at their respective frame positions, and overlapping T-F units are averaged [47] to obtain the enhanced soft-mask of the entire utterance, $\hat{M}$. Amplitudes in $\hat{M}$ are upper-bounded to 1 before they are multiplied to the original Mel-spectrum to yield the enhanced Mel-spectrum, $X[f,t]$ (see Eq. (3.12)). A 39-dimension MFCC feature vector (includes C0–C12, and their deltas and double-deltas) is computed from the enhanced Mel-spectrum at each frame. Feature mean and variance normalization (MVN) is applied on a per utterance basis to produce the final feature vector for ASR.

$$\min_{x^j} \lambda||x^j||_1 + 0.5||m_e^j - D_e x^j||_2^2 \quad \text{such that } x^j \geq 0 \tag{3.10}$$

$$\hat{m}^j = D_r\, x^j \tag{3.11}$$

$$X[f,t] = \min(\hat{M}[f,t], 1)\, Y[f,t] \tag{3.12}$$

3.2 Sparsity-based algorithms for comparison

The following sparsity-based feature enhancement algorithms are implemented for comparative purposes:

3.2.1 JDictLgMel

This JDictLgMel algorithm [48] uses jointly-sparse clean and noisy speech log-Mel-spectral dictionaries. It is implemented with the same joint dictionary learning procedure (in Section 3.1.1) on log-Mel-spectral exemplar-pairs extracted from clean and noisy speech training utterance-pairs. The sparse linear combination of noisy log-Mel-spectral dictionary entries that best approximates the target's log-Mel-spectra is found by solving the same Lasso expression, with the

same parameter settings. The denoised log-Mel-spectrum, from which MFCCs are computed, is obtained by applying the sparse solution to the clean log-Mel-spectral dictionary.

3.2.2 DictMelMask

In this DictMelMask algorithm [47], the clean speech and pure noise dictionaries, D_s and D_n, each containing 4000 Mel-spectral representations, are separately learned using the same dictionary learning parameters in Table 3.1. They are combined to form $D = [D_s, D_n]$ that has 8000 entries. The rows and columns of D are normalized to yield the final dictionary. The sparse linear combination of the dictionary entries that best represents the target Mel-spectrum is found by solving a similar l_1-minimization problem (using 200 iterations) with the Euclidean distance in Eq. (3.10) replaced by the generalized Kullback-Leibler (KL) divergence, and different λ values to penalize the activation weights of speech and noise representations. To reduce the run-time, I implemented the iterative update to end when $||x_k$ - $x_{k-1}||_2/||x_k||_2 < 0.01$, where x_k and x_{k-1} are the sparse solutions obtained in the current and previous iterations, respectively. Clean speech (S) and pure noise (N) Mel-spectra are separately reconstructed using the corresponding sparse weights and dictionary representations. A soft-mask is then computed by taking the ratio $S/(S + N)$, which is multiplied to the target Mel spectrum before MFCC extraction.

3.3 Experiment on noisy digit speech recognition

3.3.1 The Aurora-2 database and experimental setup

The Aurora-2 noisy digit speech recognition task is used as a preliminary study on the effectiveness of the proposed soft-mask-based feature enhancement technique. The Aurora-2 database contains recordings of American adults uttering strings of (1 to 7) digits. There are two training sets – (1) clean, (2) multi-conditional. The multi-conditional training set contains ITU G.712-filtered [76] clean speech and noisy speech (SNRs between 20 and 5 dB). Suburban train, babble, car, or exhibition hall noise is artificially added to clean speech to generated each noisy utterances in the multi-conditional training set. Since the exemplar dictionaries are already computed with knowledge of the noisy training utterances, the HMMs are trained using the multi-conditional set. There are three testing sets – (1) Test A, (2) Test B, and (3) Test C. Test A and Test B contains G.712-filtered speech (SNRs between 20 to -5 dB). The noise-types in Test A are the same as those found in the multi-conditional training set, while a different set of noise-types (restaurant, street, airport, train-station) is found in Test B. Test C contains ITU MIRS-filtered [76] utterances, and the noise-types involved are suburban-train and street noises. The major difference between the frequency characteristics of G.712 and MIRS filters is that the former has a flat response in the range between 300 and 3400 Hz, while the latter has a rising response with a greater attenuation at lower frequencies (illustrated in Fig. 1 of [49]). MIRS simulates the telecommunication terminal input frequency response in the technical specification GSM 03.50 [99].

The standard hidden Markov model (HMM) architecture [49] is used for the ASR system, which is built using the HTK software package [100]. A 16-state

HMM, with 3 Gaussian mixture models (GMMs) per state is used to model each digit, assuming a diagonal covariance feature matrix. Two pause models – "sil" and "sp" are also defined. The sil HMM consists of 3 states, with 6 GMMs per state. It is used to model silence intervals at the start and end of each speech file. The sp HMM consists of a single state that is tied to state 2 of the sil HMM. This sp HMM is used to model short pauses in-between words. The language model enables the test utterance to be modeled by any sequence of digits, with the flexibility of inserting a sil at the start and end of the utterance, and inserting a sp between digits.

In this preliminary study, noise subtraction was not performed on clean Mel-spectrogram in computing $\mathrm{SM}_{\mathrm{ref}}$, i.e. treat $Z_c = Y_c$ in Eq. (3.3). The soft-mask exemplar-pairs in the joint dictionary, D are randomly selected and no dictionary learning algorithm is involved. Dictionary learning is less essential in a digit recognition task, since Aurora-2 is a 11-word (0–9, and "oh") small vocabulary corpus. Four exemplar-pairs are randomly selected from each clean and noisy training utterance-pair. From this initial subset, 4000 (m_r^i, m_e^i) exemplar-pairs are randomly selected, and each exemplar-pair is concatenated to form each d^i column in D, as shown in Eq. (3.7). For the comparative sparsity-based feature enhancement algorithms, their dictionary exemplars are extracted from the same 4000 $(K \times T)$ T-F regions as the exemplars used in the proposed algorithm. The parameter λ_{stop} to the *SolveLasso* function is set to its default value of 0. Feature enhancement and utterance level MVN are applied during both training and testing.

3.3.2 Results

ASR performance is measured in terms of word accuracy ($Wacc$), which is defined in Eq. (3.13).

$$Wacc = \frac{\text{No. of words correctly recognized} - \text{No. of words inserted}}{\text{No. of words spoken}} \tag{3.13}$$

For comparison, the $Wacc$ results on Aurora-2 using plain MFCCs (i.e. no enhancement), and ETSI-AFE's [37] MFCC features, which is a state-of-the-art algorithm for the Aurora-2 ASR task, are also included. From Table 3.2, it can be observed that the DictMelMask method has the best performance, on average, for Test A. However, for Tests B and C, the performance of sparsity-based methods (JDictLgMel, DictMelMask) that utilize spectral-based dictionaries decreases sharply, such that their $Wacc$s are lower than those obtained by MFCCs. The ETSI-AFE algorithm has the best performance for Test B. The proposed JDictMask technique that uses joint SM_{ref} and SM_{est} exemplars has the best performance in Test C, and comparable performance with ETSI-AFE for Test sets A and B. The proposed JDictMask also has significant gains in $Wacc$ over the other sparsity-based methods for Tests B and C at low SNRs.

3.3.3 Discussion

Comparative methods using Mel-spectral exemplars, whether with dual (or joint) dictionaries (in the case of JDictLgMel), or with a combined dictionary (in the case of DictMelMask), perform well when the test noise and channel frequency characteristics match those present in the dictionary, as observed for Test A. However, the performance of these methods for Tests B and C suffers a large degradation when spectral shape mismatches are present. On the other hand, the proposed technique using the dual soft-mask exemplar dictionaries, is less

Table 3.2: *Wacc* (%) obtained on Aurora-2 using the multi-condition training set, with mean and variance normalization (MVN) applied. The highest *Wacc* among all algorithms are in bold.

Algorithm	20 dB	10 dB	0 dB	Avg.
Test A				
MFCC	98.53	95.99	72.73	91.03
ETSI-AFE [37]	98.68	96.23	76.45	92.21
JDictLgMel [48]	97.83	95.72	74.09	91.06
DictMelMask [47]	98.53	96.38	**80.19**	**93.03**
JDictMask	**98.70**	**96.57**	75.25	91.95
Test B				
MFCC	**98.57**	96.21	73.24	91.29
ETSI-AFE [37]	98.52	96.36	**75.09**	**91.81**
JDictLgMel [48]	97.80	95.03	66.21	88.71
DictMelMask [47]	98.50	95.65	69.72	90.05
JDictMask	98.56	**96.54**	74.34	91.74
Test C				
MFCC	**98.48**	95.20	72.50	90.59
ETSI-AFE [37]	98.37	95.31	72.84	90.87
JDictLgMel [48]	97.32	91.69	56.99	84.61
DictMelMask [47]	98.33	94.90	63.60	88.21
JDictMask	98.28	**95.84**	**75.37**	**91.50**

sensitive to such spectral shape mismatches, with less performance degradation observed across the three test sets. Fig. 3.4 plots the denoised log-Mel spectrograms obtained by the sparse exemplar-based techniques evaluated in this study for a test utterance corrupted by airport noise (in Test set B, which is not present in the multi-conditional training set) at 0 dB SNR. In this example, the proposed technique does a better job in noise suppression at the beginning of the utterance. One possible reason is that the estimated mask is computed using a noise estimation algorithm that does not make any assumption regarding the noise spectral shape present in the utterance.

An ASR experiment using enhanced MFCCs obtained by directly applying the estimated soft-mask, M_e, to the noisy Mel-spectrum, was also conducted. The average $Wacc$ obtained with MVN on test sets A, B, and C are 88.66%, 89.65% and 86.74%, respectively, which are 3–5% worse than those achieved with the proposed algorithm. This shows that the sparse mask reconstruction step is essential in enhancing ASR performance, since noise estimation is not perfect.

The advantage of using a soft-mask as part of feature enhancement can also be observed by comparing the performance of JDictLgMel with the soft-mask-based feature enhancement methods (DictMelMask and JDictMask). Generating denoised spectra by applying a soft-mask on the original noisy spectra tends to be more error-forgiving compared to reconstructing it from clean spectral exemplars as done in the JDictLgMel method. A decrease of 2–3 % in absolute Avg. $Wacc$ is observed for all test sets when the denoised Mel-spectra (S) is reconstructed directly from the sparse linear combination of clean exemplars ($Wacc$ results of this variant implementation are not shown), instead of reconstructing it indirectly via the soft-mask in the DictMelMask method.

Supplementing the dictionary with artificial noise exemplars or noise exem-

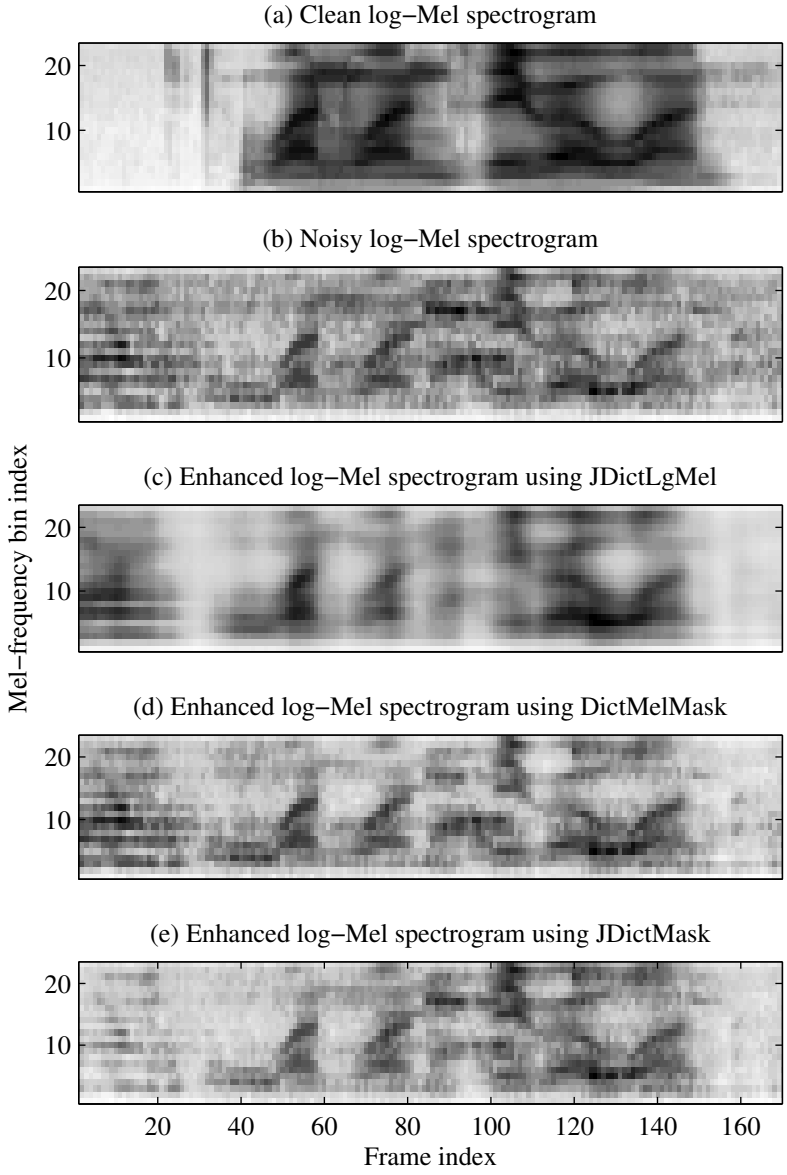

Figure 3.4: Log-Mel spectrograms of a test utterance corrupted by airport noise at 0 dB SNR. (a) Clean log-Mel spectrogram, (b) Noisy log-Mel spectrogram, (c)–(e) Enhanced log-Mel spectrograms obtained using the JDictlgMel, DictMelMask, and JDictMask, respectively.

plars extracted from the initial frames of test utterance [101] can improve the performance of DictMelMask on mismatched noise conditions. However, the focus of this study is to investigate the algorithms' performance with exemplars extracted solely from the training data. Noise exemplars extracted from the initial frames of test utterance will be incorporated in the DictMelMask algorithm in the comparative performance evaluation on the large vocabulary speech recognition task in the next section.

3.4 Experiment on large vocabulary speech recognition

3.4.1 The Aurora-4 database and experimental setup

The Aurora-4 database [102] is a standard speech corpus for noisy continuous speech recognition evaluation. It is formed by artificially adding six noise types (airport, babble, car, restaurant, street, and train) at varying signal-to-noise ratios (SNRs) to the original 5000-word vocabulary Wall Street Journal (WSJ0) corpus [103], which contains clean "read" speech utterances corresponding to sentences read from Wall Street Journal newspapers. For a training utterance, the noise (one of six types) and SNR conditions (between 10 and 20 dB in steps of 1 dB) were randomly chosen. For each of the six test conditions, the SNR was randomly chosen between 5 and 15 dB in steps of 1 dB. The 8 kHz Sehnheiser microphone data set with G.712 filtering in Aurora-4 is used for training and performance evaluation. Since the exemplars for building the dictionaries are already computed with knowledge of the noisy training utterances, the 7138 files in the Aurora-4's multi-noise training list is used for training the Hidden Markov Models (HMMs). The exemplars for dictionary learning are extracted from the noisy speech files in the same list and their corresponding clean speech versions.

The same noise types exist in the training and testing sets of Aurora-4. Additional noisy test sets are generated by adding NOISEX-92's [84] F16, factory, pink, and white noises to Aurora-4's clean test data, to evaluate the algorithms' performance under unseen noise conditions. The HTK software package [100] is used to build a HMM-based ASR system using the CMU dictionary [104] and a bigram language model (LM) with an insertion penalty of -4, and an LM scale factor of 15. Word-internal triphone models with 8 Gaussian mixture components (16 mixtures for the silence model) are trained.

In addition to the algorithms that are used for performance comparison in Aurora-2, the DictMelMask-onlineNoise algorithm – a variant implementation of the DictMelMask feature enhancement scheme – is also evaluated on Aurora-4. It appends 20 additional noise Mel-spectral exemplars to the combined dictionary, D. These 20 noise exemplars are obtained from the first and last ten 23×11 T-F Mel-spectrogram regions of the target utterance, assuming speech absence in these regions.

3.4.2 Results

Tables 3.3 and 3.4 show the word recognition accuracy ($Wacc$) obtained using various soft-mask dictionaries with the proposed JDictMask method, for test sets involving the seen and unseen noise types, respectively. "Rand-noCleanDenoise" refers to the case when 4000 soft-mask exemplar-pairs are randomly selected from the initial exemplar subset to derive the dictionaries as done in [96] (i.e. *mexTrainDL* function is not used). "DL-noCleanDenoise" means joint dictionary learning is used instead of random selection, while "DL-cleanDenoise" refers to the case where the dictionaries are learned using $\mathrm{SM}_{\mathrm{ref}}$ exemplars computed with an additional ambient noise subtraction step, as shown in Eqs. (3.1) and (3.3).

Table 3.3: *Wacc* (%) obtained by the proposed method using different soft-mask dictionaries for the seen noise types. Multi-noise training is used and the highest *Wacc* for each noise type is bold-faced. Abbreviations: Rstrnt. – Restaurant, Avg. – Average.

Dictionary	**Seen Noise Types**							
	Clean	Airport	Babble	Car	Rstrnt.	Street	Train	Avg
Rand-noCleanDenoise	87.13	78.39	77.66	85.22	74.13	75.34	75.64	79.07
DL-noCleanDenoise	86.90	78.26	78.12	85.62	**74.95**	76.67	76.14	79.52
DL-cleanDenoise	**87.26**	**78.48**	**78.44**	**86.06**	74.11	**76.95**	**77.28**	**79.80**

Table 3.4: *Wacc* (%) obtained by the proposed method using different soft-mask dictionaries for the unseen noise types. Multi-noise training is used.

Dictionary	**Unseen Noise Types**				
	F16	Factory	Pink	White	Avg.
Rand-noCleanDenoise	75.92	71.96	68.22	63.27	69.84
DL-noCleanDenoise	76.61	71.40	69.27	63.31	70.15
DL-cleanDenoise	**77.27**	**72.15**	**69.94**	**65.20**	**71.14**

Table 3.5: *Wacc* (%) obtained on the Aurora-4 test sets involving seen noise types by the various algorithms using multi-noise training. The highest *Wacc* for each noise type is bold-faced.

Algorithm	**Seen Noise Types**							
	Clean	Airport	Babble	Car	Rstrnt.	Street	Train	Avg.
MFCC	86.64	77.73	74.71	83.90	73.30	74.24	73.55	77.72
ETSI-AFE [37]	85.52	76.86	75.34	83.37	71.23	74.86	74.84	77.43
JDictLgMel [48]	78.87	65.94	67.27	76.09	62.9	63.89	62.97	68.28
DictMelMask [47]	86.44	80.93	**78.98**	85.43	**76.98**	77.53	76.44	**80.39**
DictMelMask-onlineNoise [47]	86.76	**81.00**	78.07	84.70	76.35	**78.03**	76.05	80.14
JDictMask	**87.26**	78.48	78.44	**86.06**	74.11	76.95	**77.28**	79.80

Table 3.6: *Wacc* (%) obtained on the Aurora-4 test sets involving unseen noise types by the various algorithms using multi-noise training.

Algorithm	**Unseen Noise Types**				
	F16	Factory	Pink	White	Avg.
MFCC	73.01	68.58	65.37	58.73	66.42
ETSI-AFE [37]	73.55	69.74	**70.07**	**66.69**	70.01
JDictLgMel [48]	61.46	57.22	49.88	29.76	49.58
DictMelMask [47]	73.21	72.45	66.37	50.46	65.62
DictMelMask-onlineNoise [47]	**77.96**	**72.52**	66.49	60.04	69.25
JDictMask	77.27	72.15	69.94	65.20	**71.14**

Dictionary learning improves the averaged $Wacc$ for seen noise types, as well as unseen noise types, with $Wacc$ increments observed in the majority of the noise types evaluated. This is an evidence that dictionary learning helps in generating a more complete (or evenly distributed) representations, compared to using random selection. When the additional ambient noise subtraction step is used to generate the $\mathrm{SM}_{\mathrm{ref}}$ exemplars, a more significant improvement in the averaged $Wacc$ is observed on unseen noise types than the seen noise types.

Tables 3.5 and 3.6 show the *Acc*'s achieved by the various sparsity-based feature enhancement techniques with multi-noise training, for the seen and unseen noise types, respectively.

The JDictLgMel technique has the worst performance. ETSI-AFE has a similar averaged *Acc* as plain MFCCs on seen noise types, and the best performance on the unseen pink and white noise types. The DictMelMask and DictMelMask-onlineNoise techniques have slightly higher averaged *Acc*'s than the proposed JDictMask technique on the seen noise types. On the unseen noise types, DictMelMask's averaged *Acc* is worse than JDictMask by more than 5%. With additional target noise exemplars obtained with the assumption that the first and last 200 ms of the test utterance contains pure noise, DictMelMask-onlineNoise improves DictMelMask's performance on unseen noise types. However, JDictMask still has the highest averaged *Acc* for the unseen noise types – it performs significantly better than ETSI-AFE on F16 and factory noises, and significantly better than DictMelMask-onlineNoise on pink and white noises.

3.4.3 Discussion

The proposed JDictMask technique performs better than DictMelMask and DictMelMask-onlineNoise on speech corrupted with car, train, pink and white noises

(in both seen and unseen noise types) that are relatively stationary, in which noise estimation tends to be more accurate. On the other hand, ETSI-AFE performs better than JDictMask on unseen, stationary pink and white noises because JDictMask has more insertion errors, even though it has a higher percentage of correctly recognized words. On seen, stationary car and train noises, the insertion errors of JDictMask are better controlled. This is likely because the silence model has been trained with similar seen noise remnants in the non-speech portions of the training data. These results suggest that the performance of JDictMask can potentially improve with more accurate noise estimation on both stationary and non-stationary noise types.

JDictLgMel performs significantly worse than plain MFCCs on Aurora-4, on both seen and unseen noise types. This shows the weakness of direct spectral reconstruction from dictionary representations, even when the representations are learned via dictionary learning. The learned dictionary might still be insufficient to cover all possible spectral variations, especially for a large vocabulary task. Performing spectra enhancement via a soft-mask, as done in DictMelMask, DictMelMask-onlineNoise, and JDictMask, is more error-forgiving, and yields better performance.

With our MATLAB implementation, the feature enhancement step in JDictMask takes ≈3×real-time (RT) on an Intel Xeon 2.6 GHz processor (non-parallel computing). DictMelMask and DictMelMask-onlineNoise takes ≈32×RT due to the larger number of dictionary representations and iterations used in the l_1-minimization. When the maximum number of iterations is reduced to 50, DictMelMask-onlineNoise takes ≈20×RT, and yields lower averaged *Acc* of 79.56% and 67.05% for seen and unseen noise types, respectively.

3.5 Conclusion

A feature enhancement scheme using jointly-sparse reference (SM_{ref}) and estimated (SM_{est}) soft-mask representations is proposed. SM_{est} is the ratio of a noise-subtracted Mel-spectrogram to its original noisy Mel-spectrogram (the noise spectrogram is estimated from the same noisy speech utterance). SM_{ref} is the ratio of the clean Mel-spectrogram to the noisy Mel-spectrogram. Ambient noise is subtracted from the clean Mel-spectrogram to obtain a SM_{ref} with better noise suppression properties at non-speech regions. The jointly-sparse SM_{ref} and SM_{est} dictionaries are trained via a sparsity-based dictionary learning algorithm. The sparse linear combination of SM_{est} dictionary representations that best approximates the target SM_{est} is found by solving an l_1-minimization problem. This sparse vector is applied to the SM_{ref} dictionary to produce an enhanced soft-mask for Mel-spectrogram denoising before MFCC extraction. On the Aurora-2 noisy digit speech recognition task, the proposed JDictMask algorithm has the highest averaged word accuracy on noisy speech whose channel frequency characteristics are different from the training set. On the Aurora-4 large vocabulary speech recognition task, the proposed JDictMask algorithm has the highest averaged word accuracy on speech corrupted with noise types not found in the training set. It also performs well on speech corrupted with relatively stationary noise types found in the training set. This shows that the proposed feature enhancement scheme is robust to channel and noise mismatches, and its ASR performance can potentially improve further with a more accurate noise estimation scheme. Compared to the DictMelMask algorithm that has a higher word recognition accuracy of about 2% (on average) under matched noise conditions, the proposed algorithm is about 10 times more computationally efficient, which is an advantage when runtime is of importance.

CHAPTER 4

An Exemplar-based Sparse Representation Classifier for Birdsong Phrase Classification using Limited Training Data

In this chapter, an exemplar-based sparse representation (SR) classification technique is described. This technique can perform well with limited data, as shown by a birdsong phrase classification task.

This research is a collaboration with UCLA Ecology and Evolutionary Biology (EEB) department. Prof. Charles Taylor and Prof Martin Cody of the EEB department, together with George Kossan and Kantapon Kaewtip provided assistance in data annotation and analysis. This chapter is based on the following publications:

L. N. Tan, K. Kaewtip, M. L. Cody, C. E. Taylor, and A. Alwan, "Evaluation of a sparse representation-based classifier for bird phrase classification under limited data conditions," in *Proc. Interspeech*, 2012, pp. 2522-2525.

L. N. Tan, G. Kossan, M. L. Cody, C. E. Taylor, A. Alwan, "A sparse representation-based classifier for in-set bird phrase verification and classification with limited training data," in *Proc. ICASSP*, 2013, pp. 763-767.

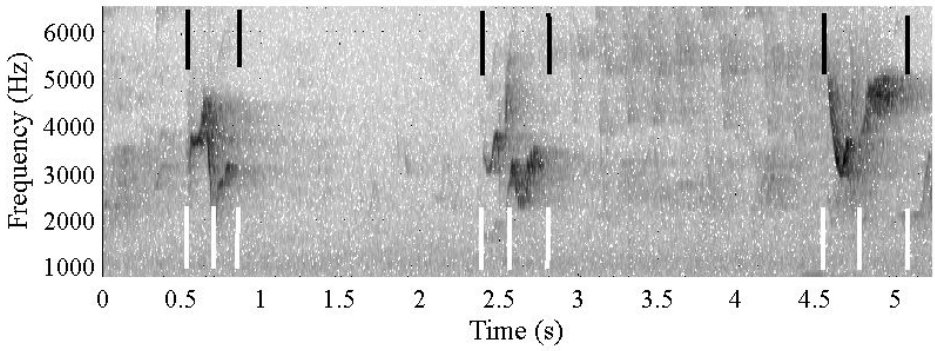

Figure 4.1: A spectrogram of a CAVI song segment. The phrase boundaries are marked by black lines, while the syllable boundaries are marked by white lines.

4.1 The CAVI Database

The birdsong phrases were obtained from song recordings of male Cassin's Vireos (abbreviated as "CAVI" in singular, or "CAVIs" in plural in this paper). This species is found commonly in many coniferous and mixed-forest bird communities in far western North America. Only the males of this species give full songs, and their songs have been described as "... a jerky series of burry phrases, separated by pauses of ≥ 1 second. Each phrase is made up of 2 to 4 notes [syllables], with song often alternating between ascending and descending phrases ..." The "song [is] repeated tirelessly, particularly when [the singing male is] unpaired ..." [105]. Fig. 4.1 shows the Mel-spectrogram of a CAVI song segment containing three different phrases separated by about 1.5 s, each consisting of two syllables.

The song recordings were obtained from two separate data collections conducted in a mixed conifer-oak forest at approximately 800 m elevation (38°29′04″N, 120°38′04″W), near the city of Volcano in California, USA. The first data collection was done between April and June 2010, when two different males, denoted by CAVI1 and CAVI2, on adjacent territories were recorded. The recordings

and annotations for this 2010 collection are available online at `http://taylor0.biology.ucla.edu/al/bioacoustics/`. The second data collection was done between April and August 2012, and song recordings from four territorial males, denoted by CAVI3–CAVI6, were used in this study. Songs were recorded using a Marantz PMD 670 with a Telinga parabolic reflector and a Sennheiser omni-directional microphone. They were saved in WAV-format (16-bit, mono) with a sampling rate of 44.1 kHz. Each file contains songs from a single CAVI, with occasionally other species songs/calls in the background. Manual annotation was performed using the Praat software [106] to note the phrase identity, and the start and end times of each phrase in the song, based on both visual spectrogram inspection and auditory recognition. In our recordings, the phrase durations are between 0.12 to 1.25 s, with a mean duration of 0.36 s. Each individual CAVI has 10 - 55 unique phrase types (which is dependent on the number of songs recorded from the individual CAVI), and 101 unique phrase classes are observed in the combined CAVI dataset. Fig. 4.2(a)–(f) show the distribution of the phrases sang by each CAVI.

Figs. 4.3 and 4.4 show the linear frequency spectrograms and Mel-spectrograms of 12 of these phrase classes, respectively. It can be observed that the acoustic signatures of some classes are very similar to each other. For example, the first two classes in the first row of Figs. 4.3 and 4.4 resemble each other except at the starting and ending portions of the phrase. The phrase classes in the last row also have very similar ascending and descending frequency signatures. Some phrases have a long pause between syllables, which can be observed in the spectrograms plotted in the second row of Figs. 4.3 and 4.4. Linear frequency spectrograms are used in the SR classification algorithm described in Section 4.2, while Mel-spectrograms are used in the DTW-SR-2pass classification algorithm described in Section 4.3.

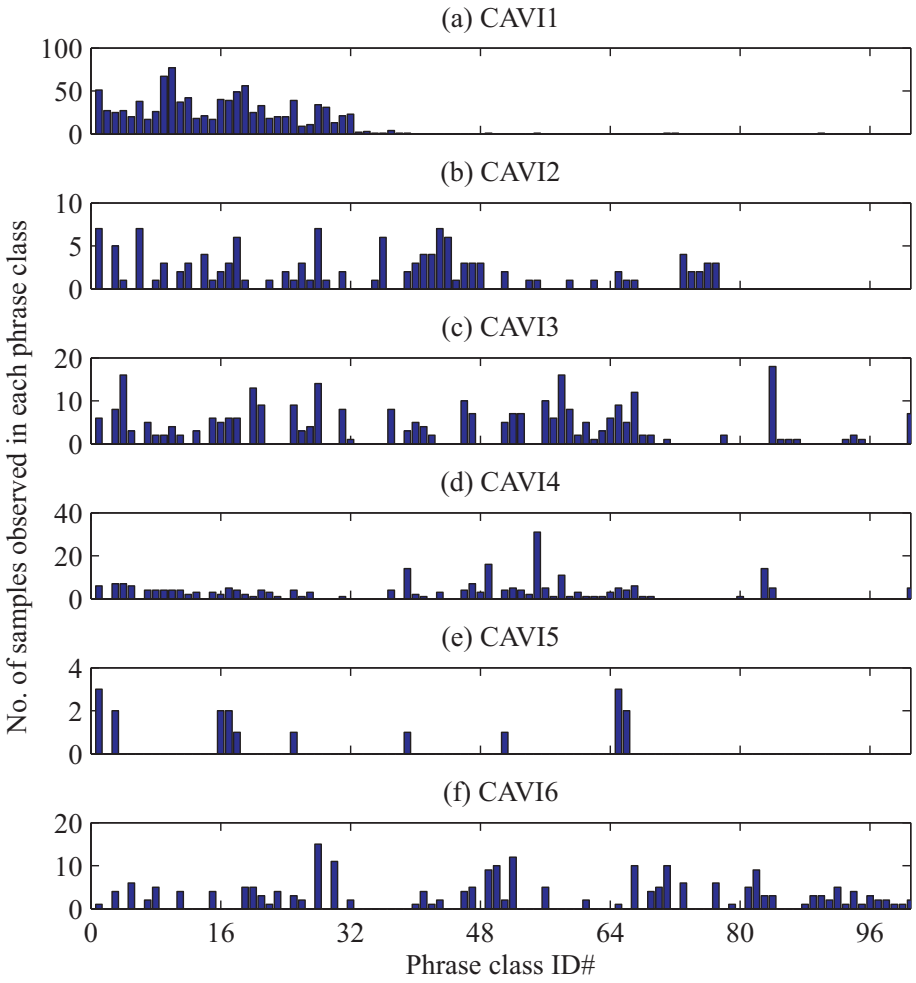

Figure 4.2: Number of samples observed in each phrase class from each of the six CAVI individuals

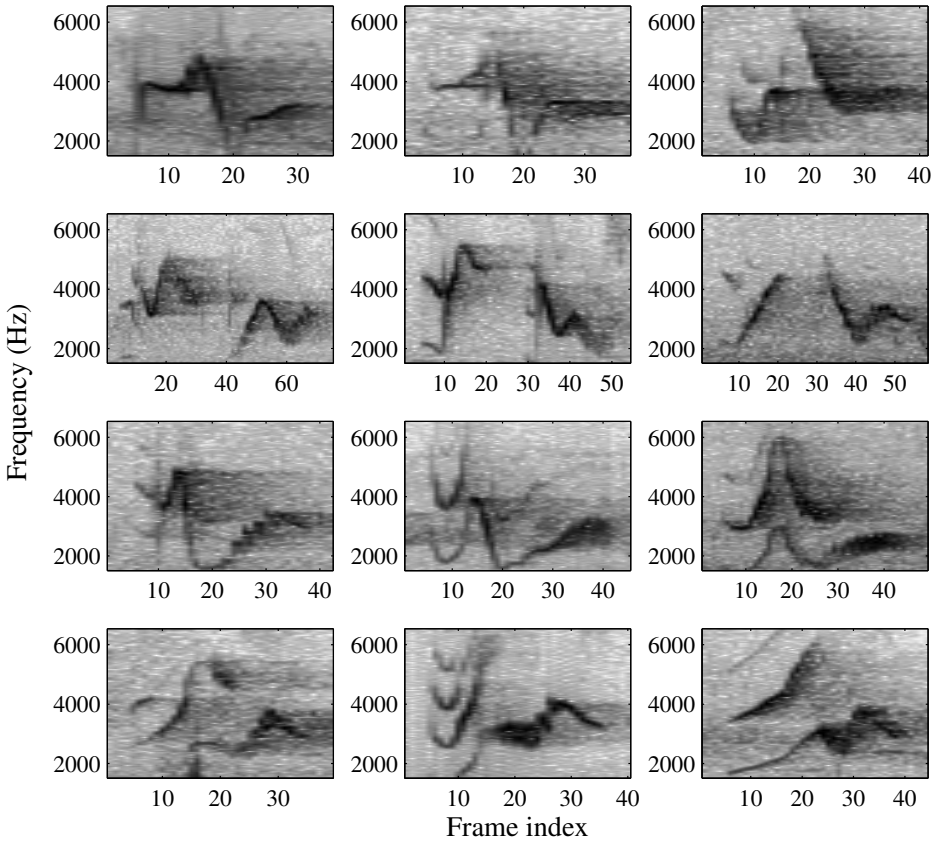

Figure 4.3: Linear frequency spectrograms of 12 phrase classes

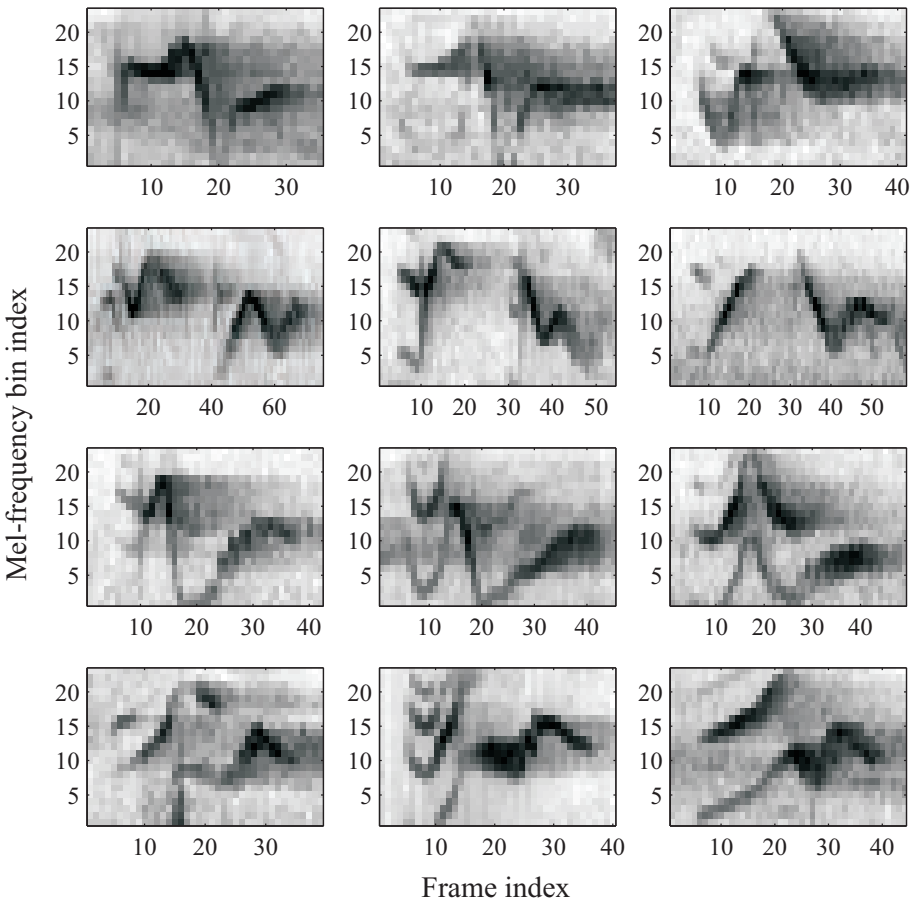

Figure 4.4: Mel-spectrograms of the 12 phrase classes

4.2 A study on the efficacy of an exemplar-based sparse representation classifier for birdsong phrase classification and verification

In this study, an exemplar-based SR classification technique that is first proposed for face recognition [71] is applied to classification and verification of CAVI phrases. The spectrographic feature extraction framework and the SR classification algorithm are described. Comparison with the NS and SVM classifiers is conducted using the same features and performance evaluation framework.

4.2.1 Spectrographic feature extraction

From Fig. 4.3, it can be observed that phrases from different classes may have very similar dominant frequency trajectories in the majority of the phrase segment. Since manual phrase identification is performed via spectrogram inspection, to avoid hand-crafting features that will be sufficiently discriminative, features are explicitly derived from spectrograms. Another reason for using features directly extracted from spectrograms is related to the sparse linear combination of feature vectors used in the SR classification technique. For example, feature vectors containing frequency values would not work well with this SR classification framework because two different phrase classes whose frequency trajectories are a factor (or multiple) of each other would likely be misclassified to each other. Sparsity-based techniques that rely image pixel intensities or time-frequency energies in spectrograms as features, have reported good performance in several image and speech applications [43, 44, 46, 47]. Although spectrographic features are generally not noise-robust without additional signal processing, they yield reasonable classification accuracies for our task because the majority of segmented

phrase tokens in our database has a high signal-to-noise ratio (SNR).

The spectrographic feature extraction algorithm is summarized in Fig. 4.5. The sampling rate was first reduced to 20 kHz because little energy is found above 10 kHz. Since every phrase token has a variable file duration, a file-duration-dependent frame shift is used to compute its spectrogram, so as to generate a spectrographic feature vector of the same dimension for each token. This file-duration-dependent frame shift is calculated as shown in Eq. (4.1), to ensure that the spectrogram of each file always contains 64 frames in time. The round(.) operator rounds off the input argument to the nearest integer. The starting sample index for frame t is denoted by S_t , while D and W denote file duration and frame length in number of samples, respectively. This translates to a frame shift of between 3 to 16 ms for the phrase durations present in the database. A frame length of 20 ms is used, thus $W = 400$ samples.

$$S_t = \text{round}\left(1 + \frac{D - W}{63}\right), \quad t = 0, 1, ..., 63. \tag{4.1}$$

A 512-point FFT is computed at each frame, and their magnitudes are converted to decibels (dB) units. It was observed that most of the bird phrase acoustic energy falls within 1.5 and 6.5 kHz, hence only the 128 FFT bins corresponding to this 5-kHz bandwidth in the spectrogram are retained. Next, this 128-by-64 spectrographic image is normalized so that it takes values between 0 and 255 (inclusive), as shown in Eq. (4.2):

$$X_{norm}(f, t) = \frac{255\,(X(f, t) - X_{\min})}{X_{\max} - X_{\min}} \tag{4.2}$$

where $X_{norm}(f, t)$ and $X(f, t)$ are the post- and pre-normalized spectrographic pixels, respectively, with frequency bin index f, and frame index t. $X_{\min}$ and $X_{\max}$ are the corresponding minimum and maximum values of the pre-normalized

spectrogram. Finally, the normalized spectrographic image is reshaped into a 8192-by-1 feature vector by concatenating the columns in $X_{norm}(f, t)$.

The dimension of the feature vector is then reduced by doing a principal component analysis (PCA) on the training set. Mean subtraction is not performed before computing the eigenvectors because omitting it yields higher phrase classification accuracies for all classifiers evaluated. For our performance evaluation, the feature dimension is reduced to three different values, $d = 32$, 50 and 128, which corresponds to image resizing factors of 1/16, 1/12 and 1/8, respectively. The dimensionally reduced feature vectors are subsequently normalized to unit length before they are passed into the classifier for training and testing. These same features are used in all the classification techniques for a fair comparison. The feature dimension, d, is varied to investigate the performance dependency of the proposed and comparative classification algorithms on the dimension of the feature vectors.

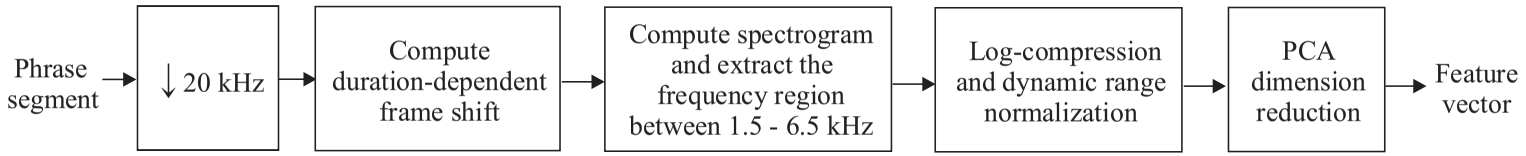

Figure 4.5: Block diagram for spectrographic feature extraction.

4.2.2 Sparse representation (SR) classifier

The SR classification algorithm summarized in Eqs. (4.3) – (4.6), follows "Algorithm 1" described in [71]. The SR classifier finds a sparse linear combination of training feature vectors that best represents the test feature vector, b. This linear combination is found by solving for a sparse vector $x = [x[1], x[2], ..., x[m]]^T \in \mathbf{R}^{m\times 1}$ via the l_1-minimization convex optimization problem defined in Eq. (4.3). In this case, each column, $a_i \in \mathbf{R}^{p\times 1}$, in dictionary matrix $\mathbf{A} = [a_1, a_2, ..., a_m] \in \mathbf{R}^{p\times m}$ contains one exemplar or feature vector from the training set, and $m = Kn$, where K is the total number of classes in the training set, and n is the number of training samples per class. Generally, the exemplars in the dictionary matrix are normalized to unit length [43, 47] so as not to bias towards the selection of exemplars with larger feature values when the l_1 solver tries to minimize $||x||_1$. The difference tolerance, ε is usually set to a small value, to allow a small degree of differences between the test feature vector and the one reconstructed using a sparse linear combination of the training feature vectors. Normalizing the test feature vector to unit length enables ε to be fixed to a constant value, instead of varying it proportional to the Euclidean norm of each test vector. The l_1 solver used to solve Eq. (4.3) is the l_1-MAGIC MATLAB toolbox [107], with ε set to 0.05.

$$\min \|x\|_1 \text{ subject to } \|\mathbf{A}x - b\|_2 \leq \varepsilon \tag{4.3}$$

$$r_k = b - \mathbf{A}\delta_k(x) \text{ , for } k = 1, 2, ..., K. \tag{4.4}$$

$$\delta_k(x) = y, \text{ where } y[i] = \begin{cases} x[i], \text{ if } a_i \in \text{class } k \\ 0 \text{ , otherwise} \end{cases} \tag{4.5}$$

$$O_{\text{SR}} = \arg\min_k \|r_k\|_2 \tag{4.6}$$

$$q_{\text{SR}} = \text{SCI}(x) = \frac{K \cdot \max_i \|\delta_i(x)\|_1 / \|x\|_1 - 1}{K - 1} \in [0, 1] \tag{4.7}$$

After the sparse representation is found, the residual vector, r_k between b and $\mathbf{A}\delta_k(x)$ is computed in Eq. (4.4) for each of the K classes in the training set. The $\delta_k(x)$ function outputs a vector, y, whose coefficients, $y[i] = x[i]$ if a_i is a training feature vector from class k. All other coefficients in y are set to zero, as shown in Eq. (4.5). The class that yields the minimum residual norm, $\|r_k\|_2$, is the output decision, O_{SR} of the SR classifier, as shown in Eq. (4.6).

The confidence measure of the SR classifier, q_{SR}, that is used for phrase verification is the sparsity concentration index (SCI(x)). The SCI(x) was used successfully for outlier face rejection in [43, 71]. This confidence index measures the maximum concentration of the computed sparse x coefficients on a single in-set phrase class, and is computed using Eq. (4.7). At one extreme, SCI(x)=1 when the coefficients of only one phrase class are activated in x. At the other extreme, SCI(x)=0 when the coefficients of all classes are equally activated. In general, the sparse solution, x, computed from an in-set phrase would have large non-zero coefficients corresponding to training feature vectors from the same phrase class. This is illustrated in Figs. 4.6a(i)–b(ii). Fig. 4.6b(i) shows the spectrogram of a test sample from an in-set class. Fig. 4.6b(ii) shows the sparse x solution computed with the test feature vector extracted from the spectrogram in Fig. 4.6b(i). The sum of the first three x coefficients in Fig. 4.6b(ii) is large because the first three training feature vectors in $\mathbf{A}$, whose spectrograms are plotted in Fig. 4.6a(i) - a(iii), are of the same phrase class and have very similar phrase spectrographic profile as the test phrase in Fig. 4.6b(i). On the other hand, an out-of-set bird phrase in Fig. 4.6c(i), would have coefficients distributed across multiple in-set classes, as observed in Fig. 4.6c(ii), since it is usually not well represented by any single class.

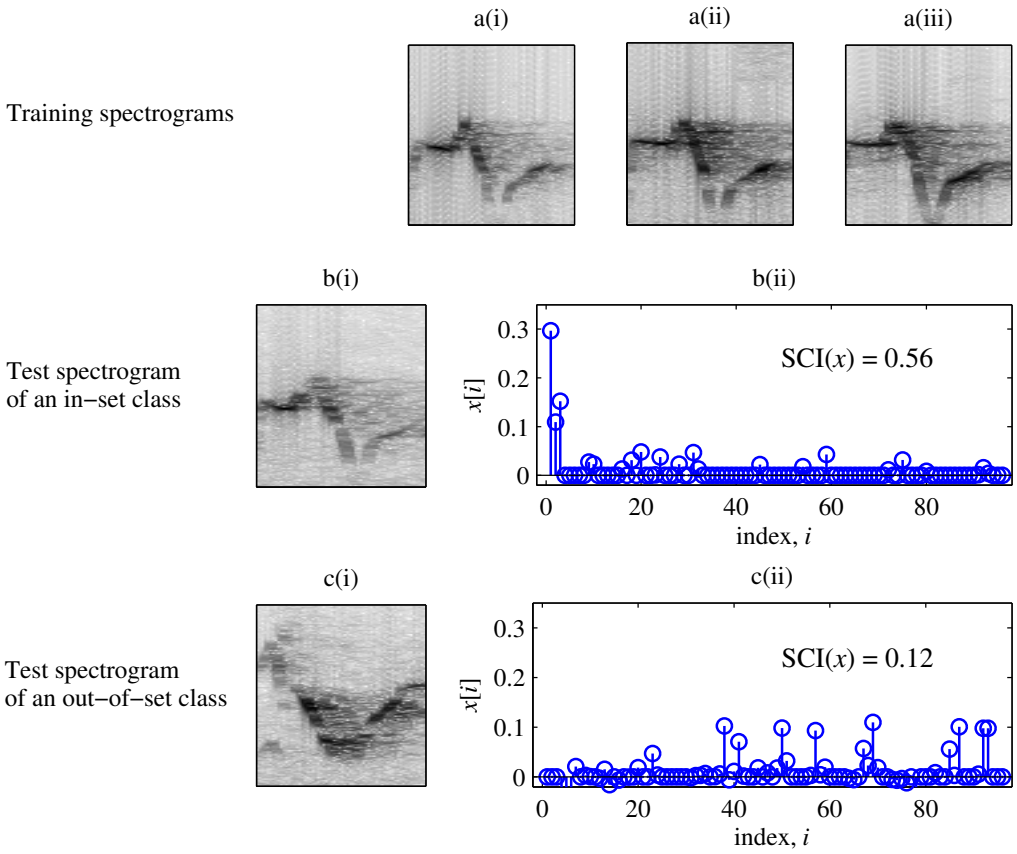

Figure 4.6: Examples of sparse vectors x computed from in-set and out-of-set bird phrases with manually detected time boundaries. There are 3 training samples per in-set class. a(i)-a(iii) plot the spectrograms of the training phrases whose corresponding feature vectors are stored in the first three columns of matrix $\mathbf{A}$. b(i) plots the spectrogram of a test phrase of the same class as a(i)-a(iii). c(i) plots the spectrogram of a test phrase from an out-of-set class. b(ii), and c(ii) plot the sparse vector x computed using the feature vector extracted from the spectrogram on its respective left.

4.2.3 Comparative classification algorithms

4.2.3.1 Support vector machine (SVM) classifier

The multi-class SVM classifier is implemented using the software tool known as LIBSVM [108], which uses a one-against-one decomposition strategy. The SVM classifier's confidence measure, q_{SVM}, for phrase verification is the probability estimate of the most-likely class [109] computed by LIBSVM. The Gaussian radial basis function (RBF), $K(y, z) = \exp(-\gamma\|y - z\|^2)$ is the selected kernel, and a cross-validation (CV) training strategy is used to select an optimal pair of *i.* regularization factor (a.k.a soft margin parameter), $C \in \{2^{-1}, 2^0, ..., 2^7\}$, and *ii.* the RBF parameter, $\gamma \in \{2^{-4}, 2^{-3}, ..., 2^5\}$. This parameter tuning is done for all pairs of n and d, except when $n = 3$ and $d = 128$. For this pair, only C is tuned using CV, with a fixed $\gamma = 4$, because a significant decrease ($> 20\%$ in absolute difference) in the classification accuracy is observed with a CV-tuned γ value, due to over-fitting to the small set of training set when d is large. When a linear kernel function is used, it gives better classification accuracies than the RBF only at $d = 128$ when $n = 3$, 4, and 5. Even in these cases, the linear SVM still performs significantly worse than the comparative algorithms. Hence, only the RBF SVM results will be presented.

4.2.3.2 Nearest subspace (NS) classifier

The classical NS classifier [110] finds the class subspace that best represents the test vector, b. First, the orthonormal basis of each phrase class is found by performing singular value decomposition (SVD) on a matrix $\mathbf{A}_k$, where each column in $\mathbf{A}_k$ contains one feature vector from class k in the training set. The first n left singular vectors, u_1, ..., u_n in the orthogonal matrix $\mathbf{U}_k$ form the orthonormal

basis, $\mathbf{P}_k$, of class k's subspace. The output of the NS classifier, O_{NS}, is the class that yields the minimum residual norm between b and its class-subspace projection, as shown in Eqs. (4.8) and (4.9). The NS classifier's confidence measure, q_{NS}, that is used for phrase verification is computed using the minimum residual norm, as shown in Eq. (4.10)

$$r_k = b - \mathbf{P}_k \mathbf{P}_k^T b \tag{4.8}$$

$$O_{\text{NS}} = \arg\min_k \|r_k\|_2 \tag{4.9}$$

$$q_{\text{NS}} = 1 - \|r_{O_{\text{NS}}}\|_2 \tag{4.10}$$

4.2.4 Performance evaluation framework

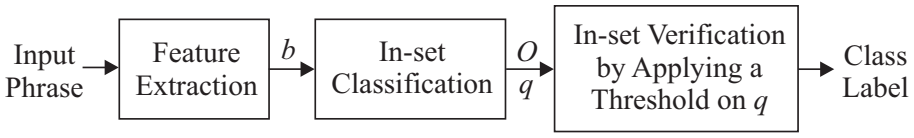

Figure 4.7: The classification and verification evaluation framework

The block diagram in Fig. 4.7 illustrates the common framework used with each classification algorithm to perform birdsong phrase classification and verification. The dimension-reduced spectrographic feature vector, b is extracted from the test segment, followed by the in-set classification. Note that no token from the *out-of-set* classes is used for training. For each test sample, the classifier outputs an in-set class label – O, and a confidence measure – q, regarding the correctness of O. In-set/out-of-set verification is performed by applying a fixed threshold on q. If q is larger than the threshold, the test sample would be detected as an in-set class, and the decision O will be the final class label attributed to the input bird phrase. Otherwise, the test sample is detected as an out-of-set class, and given the class label named "Others".

The 2010 CAVI database is used in this study. The training and testing sets contains phrases from the same pair of CAVI individuals. There are 63 phrase classes, each with 1 to 69 tokens. The more frequently observed $K = 32$ phrase classes that have at least 10 tokens are used for the closed-set classification task. There are 1034 tokens in this closed-set, and they form the *in-set* training and test data. The 82 tokens of the remaining 31 classes form the *out-of-set* test data, which are used together with the *in-set* tokens for the in-set/out-of-set verification task.

Phrase classification accuracy (Acc) is evaluated on the test samples that belong to the in-set classes. It is calculated as shown in Eq. (4.11), where C_{in} is the number of in-set test samples that are classified correctly, and N_{in} is the total number of in-set test samples.

$$\text{Acc} = \frac{C_{\text{in}}}{N_{\text{in}}} \times 100\% \tag{4.11}$$

The performance measures of the verification task are p_{Det} – the proportion of in-set phrases that is correctly identified as in-set (true positives), and p_{FA} – the proportion of out-of-set phrases that is incorrectly identified as in-set (false positives). They are calculated as shown in Eqs. (4.12) and (4.13), where D_{in} is the number of in-set test samples that are correctly detected as in-set, D_{out} is the number of out-of-set test samples that are correctly detected as out-of-set, and N_{out} is the total number of out-of-set test samples.

$$p_{Det} = \frac{D_{\text{in}}}{N_{\text{in}}} \times 100\% \tag{4.12}$$

$$p_{FA} = 1 - \frac{D_{\text{out}}}{N_{\text{out}}} \times 100\% \tag{4.13}$$

Besides evaluating the classification and verification performances separately, the classifiers' joint classification and verification performances are evaluated as well. This joint task can also be considered as a 33-phrase class (32 in-set classes + 1 out-of-set class) bird phrase classification task. The joint classification accuracy, JAcc, is calculated by taking an average of the percentage of in-set bird phrases that is correctly classified ($\frac{C_{\text{in}}}{N_{\text{in}}}$), and the percentage of out-of-set bird phrases that is correctly labeled as "Others" ($\frac{D_{\text{out}}}{N_{\text{out}}}$), as shown in Eq. (4.14). Taking the average of these two components ensures that an equal importance is placed on the classification accuracies of both in-set and out-of-set test samples, which is necessary since the number of in-set test tokens ($N_{\text{in}} = 1034 - Kn > 800$) is much greater than the number of out-of-set test tokens ($N_{\text{out}} = 82$). Otherwise, the results would be biased towards the classification accuracy of in-set phrase tokens, and the selected threshold would yield an undesirably high p_{FA} for the verification task.

$$\text{JAcc} = 0.5 \left(\frac{C_{\text{in}}}{N_{\text{in}}} + \frac{D_{\text{out}}}{N_{\text{out}}} \right) \times 100\% \tag{4.14}$$

4.2.5 Results

4.2.5.1 Classification accuracy of in-set test samples

To evaluate classification accuracy of in-set phrases, the verification threshold is set to -1, so that none of the in-set test samples is detected as out-of-set. Table 4.1 tabulates Acc, the phrase classification accuracies of the in-set test samples, with different pairs of n (number of training samples per class) and d (feature dimension), for the classifications techniques evaluated. For each case, the results were obtained by averaging the Acc's of five independent experiments, whereby training samples are randomly selected in each experiment. In general, the Acc

Table 4.1: Average Acc (%) for different values of n and d. The highest value for each case is boldfaced.

n	Classifier	$d = 32$	$d = 50$	$d = 128$
3	SR	**89.3**	**90.2**	N.A
	SVM	79.0	79.5	76.7
	NS	86.0	86.7	**87.6**
5	SR	**91.9**	**92.6**	**92.9**
	SVM	86.6	87.6	89.4
	NS	89.6	90.6	91.5
7	SR	**92.8**	**93.6**	**93.8**
	SVM	89.2	90.0	91.2
	NS	91.6	91.9	92.5

improves with increasing n and d for all three classifiers. No results are shown for the SR classifier at $n = 3$ and $d = 128$ because there are more columns than rows in matrix $\mathbf{A}$ such that Eq. (4.3) becomes an over-determined linear system, and the l_1 solver used is unable to find a feasible x solution.

The Acc's achieved by the SR classifier are consistently the highest compared to the SVM and NS classifiers in all cases. The McNemar test [111] was performed to evaluate the statistical significance of SR classifiers performance against the SVM and NS classifiers. A p-value of less than 0.03 is obtained for all the cases tabulated above, indicating that the improvement in bird phrase classification accuracies of the SR classifier over the other two classifiers is of statistical significance. The performance gain of the SR classifier over the comparative algorithms increases as n decreases. For example, the SR classifier's Acc is 1.2% higher (absolute difference) than the NS classifier's at $n = 7$ and $d = 32$,

and the Acc difference increases to 3.3% at $n = 3$, with the same d. This implies that the performance of the SR classifier is more robust in limited training data condition, compared to the NS and SVM classifiers. One reason for the good performance of the SR classifier is that spectrograms of phrases from the same class are generally very similar to one another for this test set, except for small time differences in the silence intervals in the beginning and end of a phrase segment, due to human inconsistencies in specifying phrase boundaries during the manual annotation process.

4.2.5.2 Performance of in-set/out-of-set verification

The performance of the classifiers for in-set bird phrase verification are evaluated based on the receiver operating characteristic (ROC) curve, which plots p_{Det} versus p_{FA}. To show the performance trends of the classifiers' with different values of n and d, Fig. 4.8 plots the ROC curves of each classifier when n is varied at a fixed $d = 50$, while Fig. 4.9 plots the ROC curves when d is varied at a fixed $n = 5$.

In Fig. 4.8, it is observed that the performance of all classifiers generally improves as n increases. It is also observed in Fig. 4.8b that when $d = 50$, the SR classifier has the highest p_{Det} for $p_{FA} < 0.2$ among the other classifiers, and the NS classifier has the second-best performance, with an ROC curve that is very similar to the SR classifier's at higher p_{FA}. Fig. 4.9 shows that the SR classifier's verification performance has a greater improvement over the NS classifier when $d = 128$ (Fig. 4.9b) compared to $d = 50$ (Fig. 4.8b). At $d = 32$, the SR classifier's performance is slightly worse than the NS's for $p_{FA} > 0.2$ % (see Fig. 4.9a). The SVM classifier performs significantly worse than the SR and NS classifiers in this verification task.

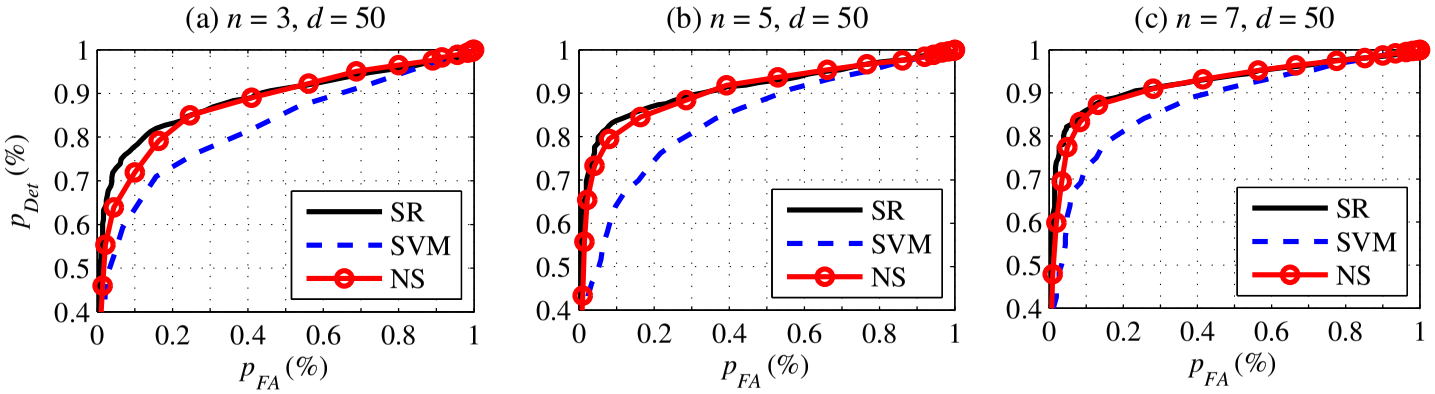

Figure 4.8: ROC curves for in-set bird phrase verification with $d = 50$, for (a) $n = 3$, (b) $n = 5$, and (c) $n = 7$, where d is the dimension of the feature vector retained after PCA, and n is the number of training tokens per class.

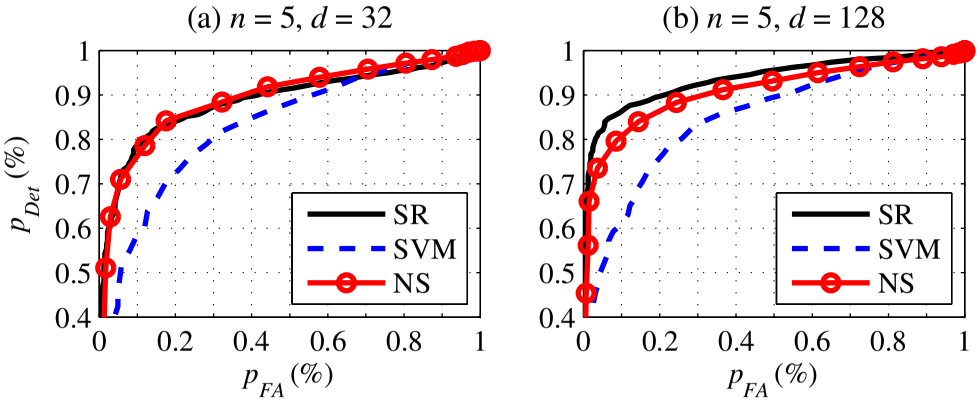

Figure 4.9: ROC curves for bird phrase verification with $n = 5$, for (a) $d = 32$ and (b) $d = 128$.

Table 4.2: Average JAcc (%) for different values of n and d. The highest value for each case is boldfaced.

n	Classifier	$d = 32$	$d = 50$	$d = 128$
3	SR	**81.8**	**83.6**	N.A.
	SVM	73.5	74.8	72.4
	NS	79.8	79.6	**82.3**
5	SR	**83.9**	**87.0**	**88.6**
	SVM	75.3	76.8	77.3
	NS	82.0	84.7	84.7
7	SR	**85.5**	**88.2**	**89.6**
	SVM	78.7	80.8	81.6
	NS	84.5	86.4	86.4

4.2.5.3 Joint classification and verification performance

Table 4.2 shows the averaged JAcc obtained by the classifiers for this joint classification and verification task, with different values of n and d. For each pair of n and d, the verification threshold is varied between 0 and 1, in steps of 0.005, and the value that yields the highest average *Acc* (over the five experiments) is the threshold used for each classifier. The SR classifier achieves the highest JAcc in all cases, except when $n = 3$ and $d = 128$ (for which the SR classifier is not able to generate a classification output) because Eq. (4.3) becomes an over-determined linear system in this case, and the l_1 solver used is unable to find a feasible x solution.

Note that the JAcc values are generally lower than the Acc's reported in Section 4.2.5.1 for all classifiers, due to additional verification errors. This is especially true for the SVM classifier whose verification performance is poor as

observed in the ROC curves in Section 4.2.5.2. Thus, SVM's JAcc is much lower than the SR and NS classifiers in all cases. In general, the improvement in JAcc of the SR classifier over the second-best performing algorithm (NS classifier) increases with d at a fixed n, due to the SR classifier's increasing verification performance gain over the other algorithms. This is also true in the case of $n = 7$, when the SR classifier's in-set classification accuracy, Acc (prior to in-set verification), is just slightly higher than the SVM and NS classifiers. On the other hand, at $d = 32$, even though SR's verification performance is slightly worse than NS's, as observed in the ROC curves in Section 4.2.5.2, the SR classifier still has a higher JAcc due to its superior in-set phrase classification performance.

4.2.6 Discussion

In general, the SR classifier has a better in-set classification performance over a wider range of conditions compared to the nearest neighbor (NN) and NS classifiers, because the L1 minimization algorithm selects the smallest subset of training exemplars that best represents the test vector, i.e. it is not dependent solely on the nearest exemplar (as in an NN classifier), nor on the subspace spanned by all the training exemplars of a class (as in an NS classifier). However, this also means that the SR classifier would be more sensitive to classification errors due to outliers or mislabeled exemplars in the training set, since the SR classifier might misclassify a test vector that is in close proximity to a single or a neighborhood of outlier(s) or mislabeled exemplar(s). For the SVM classifier, if the outliers are not one of the support vectors, its decision boundary would not be affected by these outliers. On the other hand, when the classes in training data are non-separable by the SVM, the SR classifier might have an advantage over the SVM classifier, depending on similarity between the test vector and the

training exemplars from the same class. This is the most likely scenario that led to the SR classifiers superior performance over the SVMs. An evidence of over-fitting with the RBF-based SVM classifier is also observed when n is small and d is high, since its Acc at $n = 3$ and $d = 128$ are lower than those obtained at $d = 32$ or 50 with the same n.

The SCI of the SR classifier is generally a robust measure for the verification task because the sparse x vector is computed with full (global) information from all phrase classes in the training set [71]. In contrast, each class residual of the NS classifier is separately computed using information only from its respective phrase class; while LibSVM's probability estimates [109] are obtained by combining pair-wise class probabilities derived from each one-against-one binary classifier in the multi-class SVM. The SCI yields the best ROC curve at $d = 128$, and a comparable performance to the NS classifier's residual measure at $d = 32$. The additional between-class differences that are retained in the training exemplars when a larger d is used, result in a higher sparse weight concentration on the correct in-set class, which in turn improve the reliability of the SCI measure for this verification task. However, for the SR algorithm, the maximum d allowed is also upper-bounded by the total number of training samples.

The joint classification and verification accuracy, JAcc of the SR classifer is also the least sensitive to verification threshold perturbations compared to the NS and SVM. This is shown in Fig. 4.10, which plots the variation of JAcc with the threshold used in each classifier, for a particular experiment with $n = 5$ and $d = 50$ (similar trends in JAcc are observed at other values of n and d). The JAcc of the SR classifier has the broadest peak lobe, while the NS classifier has the narrowest.

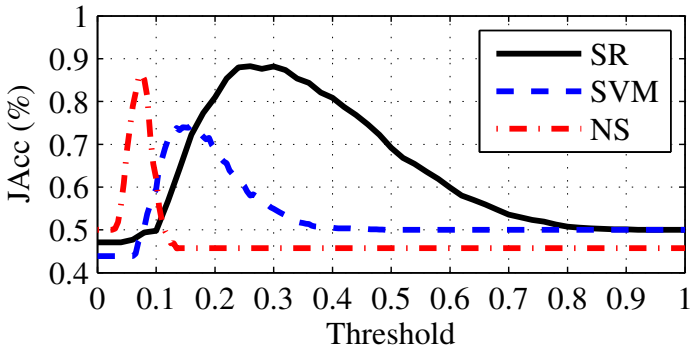

Figure 4.10: JAcc variation with verification threshold for an experiment with $n = 5$ and $d = 50$.

4.2.7 Conclusion

The SR classifier that had reported good face recognition and outlier rejection performance [43, 71] was applied to limited data, bird phrase classification and verification. The data set consists of manually-segmented birdsong phrases from two male Cassin's Vireos. The training set contains only bird phrases from the 32 in-set phrase classes. The SR classifier obtains the highest in-set classification accuracies compared to NS and SVM classifiers. An increasing performance gain over the latter two classifiers was observed with a decreasing number of tokens per phrase used for training. This suggests that the SR classifier is a promising technique for bird phrase classification when the amount of training data is limited. When the feature dimension is large enough, the SR's sparsity concentration index (SCI) is also a more reliable measure for distinguishing between in-set and out-of-set bird phrases in the verification task, compared to the NS classifier's residual, and the SVM classifier's probability estimate. Besides evaluating the classification and verification performances separately, a joint classification and verification performance evaluation is conducted, in which the classifier has to classify the bird phrases into one of the 33 phrase classes, consisting of 32 in-set

phrase classes, and 1 collective out-of-set class for the remaining phrase classes. The SR classifier also outperforms the NS and the SVM classifiers in this joint task, due to its good performances for both in-set classification and verification.

4.3 Dynamic time warping and a two-pass sparse representation classifier for birdsong phrase classification

When the feature extraction scheme used in the preliminary study (which varies the amount of inter-frame overlap to obtain spectrograms of the same dimensions for different phrase segments) is applied to automatically segmented phrases or phrases from different bird individuals, spectrographic features from the same phrase class tend to be time-misaligned. This results in large within-class feature variabilities, and low classification accuracies.

In this section, an improved SR-based classification algorithm is developed [112] to reduce within-class feature variabilities. This DTW-SR-2pass classification algorithm consists of dynamic time warping (DTW), feature normalization, and a two-pass SR classification. The algorithm improves classification accuracy when acoustic variations due to individual bird differences and phrase segmentation inconsistencies exist. Two separate test sets are formed – one containing phrases from the bird individuals found in the training set, and the other containing phrases from a different group of bird individuals. Besides manually-segmented phrases, performance evaluation is also conducted on automatically detected and segmented phrases obtained using an existing energy-based detection and segmentation algorithm. [54].

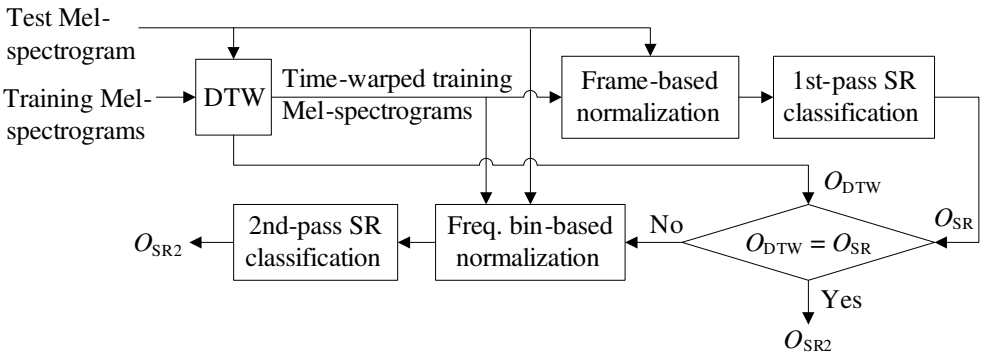

Figure 4.11: Flow chart of the proposed two-pass sparse representation classification framework. O_X is the class decision of classifier X, where X is either DTW, SR (from 1st-pass SR), or SR2 (from 2nd-pass SR).

4.3.1 Proposed DTW-SR-2pass classification algorithm

The flow chart in Fig. 4.11 summarizes the proposed bird phrase classification algorithm, abbreviated DTW-SR-2pass. The algorithm applies dynamic time warping on the training Mel-spectrograms, followed by a two-pass SR classification. Mel-spectrogram generation and the signal processing performed in each stage are described in the following subsections.

4.3.1.1 Mel-spectrogram generation

A Mel-spectrogram is computed from each bird phrase segment in both the training and testing sets. Fig. 4.12 shows the various steps involved in generating each Mel-spectrogram. The segment is first downsampled from 44.1 kHz to 20 kHz because little phrase energy is found above 10 kHz. It is then split into multiple 20 ms frames, with 50% overlap (10 ms shift). A Hamming window is applied to each frame, followed by a 512-point FFT to obtain the power spectrum of each frame. The Mel-spectrogram is generated by applying a 23-channel Mel-

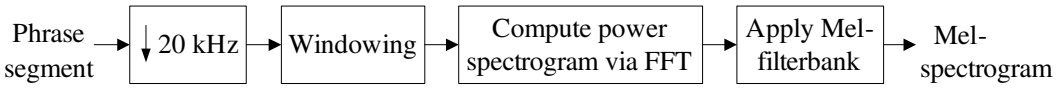

Figure 4.12: Block diagram on Mel-spectrogram generation

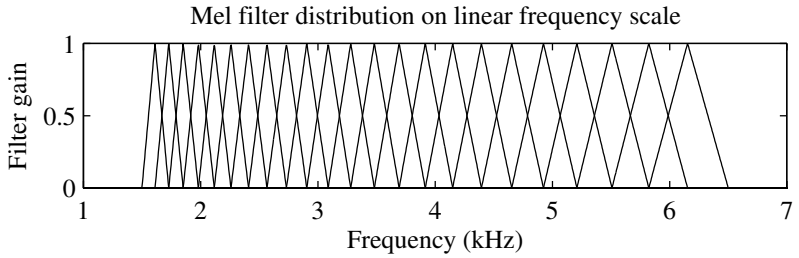

Figure 4.13: Distribution of Mel filters in the frequency range of interest

filterbank (shown in the Fig. 4.13) that covers the 1.5 – 6.5 kHz frequency range of the power spectrum, in which most of the CAVI phrase energy falls within. When energies in multiple linear-frequency bins are summed to produce each frequency bin component in the Mel-spectrogram, it results in smoother energy transitions across frequency bins. This can be observed by comparing the linear frequency spectrograms in Fig. 4.3 to the Mel-spectrograms in Fig. 4.4. Using Mel-spectrograms instead of linear-frequency power spectrograms yields a higher classification accuracy when tested on DTW, SVM and SR classifiers in a preliminary experiment. This is likely because the energy in the same frequency bin and frame of Mel-spectrograms from the same phrase class are more similar to one another, compared to the case when linear frequency spectrograms are used.

4.3.1.2 Dynamic time warping of the training set using Mel-spectrograms

Next, dynamic time warping is applied on each training Mel-spectrogram to yield a time-warped version that is more similar to and has the same number of frames

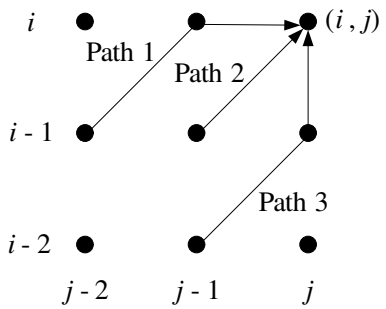

Figure 4.14: The valid DTW paths permitted to reach point (i,j)

as that in the test Mel-spectrogram (whose phrase class is to be determined). This is done by computing the optimal path through a similarity matrix, Θ, where each $\Theta(i,j)$ is a cosine similarity measure (see Eq. (4.15)) that emphasizes the similarity in spectral shape, instead of energy values, between vectors s_i and r_j, which are the i-th frame of the test Mel-spectrogram, and the j-th frame of the training Mel-spectrogram, respectively. Cosine similarity has also been found to be a good spectral similarity measure for performing DTW in [73, 113]. Computing the cosine similarity using a linear power scale rather than a log power scale Mel-spectrogram yields a higher classification accuracy with the DTW classifier (the same trend is also reported in [113]). Hence, the linear power scale Mel-spectrogram is used.

$$\Theta(i,j) = \frac{{s_i}^T r_j}{||s_i||_2 \, ||r_j||_2} \tag{4.15}$$

$$D(i,j) = \max \begin{cases} D(i\text{-}1,j\text{-}2) + \frac{1}{2}\Theta(i,j\text{-}1) + \frac{1}{2}\Theta(i,j), \text{ Path 1} \\ D(i\text{-}1,j\text{-}1) + \Theta(i,j), \text{ Path 2} \\ D(i\text{-}2,j\text{-}1) + \Theta(i\text{-}1,j) + \Theta(i,j), \text{ Path 3} \end{cases} \tag{4.16}$$

$$d = \max_{j \in [N\text{-}T\text{-}1,\ldots,N+T\text{-}1]} D(i = M\text{-}1, j) \tag{4.17}$$

$$\begin{aligned} &\text{If Path 1,} \quad \tilde{r}_i = \tfrac{1}{2}(r_j + r_{j\text{-}1}) \\ &\text{If Path 2,} \quad \tilde{r}_i = r_j \\ &\text{If Path 3,} \quad \tilde{r}_i = \tilde{r}_{i\text{-}1} = r_j \end{aligned} \tag{4.18}$$

Fig. 4.14 illustrated the Type I local path constraints [114] imposed, which limit the minimum and maximum scale of time-warping to 0.5 and 2, respectively. The intermediate accumulative score, $D(i, j)$, is computed recursively for i = 0, 1, ... , M-1, and for j = -T, -T+1, ... , -1 , 0 , 1, ... , N+T-1, where M and N denotes the number of frames in the test and training segments within the detected time boundaries, respectively. T is the additional number of frames extended beyond the training segment boundaries that are used to generate each training Mel-spectrogram. In Eq. (4.16), the Θ values are weighted by 0.5 for Path 1, so as not to double count the similarity score with the same frame of the test spectrogram, i.e. s_i. This makes d, the final accumulative score of the optimal path found between the test spectrogram and the current training spectrogram comparable across all training spectrograms or samples. For each training sample, d is computed using Eq. (4.17), which allows the optimal path to stop at any frame within the last $2T$+1 frames of the extended training Mel-spectrogram, while covering all the test frames. The algorithm also permits paths starting within the first $2T$+1 frames of the extended training Mel-spectrogram. This flexibility (that resulted from setting $T > 0$) improves classification accuracies when there is phrase position variation within the segment due to inconsistently determined time boundaries. Large phrase position variations can exist when an automatic segmentation algorithm is used. To obtain the time-warped training spectrogram, $\tilde{R} = [\tilde{r}_0, ..., \tilde{r}_{M-1}]$, that has the same number of frames as the test spectrogram, the optimal path is back-tracked as shown in Eq. (4.18). Fig. 4.15 shows an example of a training Mel-spectrogram before and after DTW to match

the test Mel-spectrogram.

The phrase class of the training sample that yields the highest d, over all training samples, is the classification decision of this DTW classifier, O_{DTW}. The DTW decision is used in a subsequent step in deciding whether a second-pass of SR classification is necessary.

4.3.1.3 Feature normalization prior 1st-pass SR classification

To emphasize spectral shape similarity over spectral amplitude similarity, all time-warped training Mel-spectrograms and the test Mel-spectrogram are normalized to have the same Euclidean norm for every frame, prior to amplitude log-compression. For simplicity, the vector of values in each frame is normalized such that it has a Euclidean norm of 1. Fig. 4.16 shows the log Mel-spectrograms of two samples from the same phrase class without frame normalization (Figs. 4.16a and 4.16c), and with frame normalization (Figs. 4.16b and 4.16d). It can be observed that the features of these two samples become more similar with the additional frame normalization.

4.3.1.4 Dimension reduction and SR classification

After frame normalization and log-compression, the frames of each Mel-spectrogram are concatenated to form a single feature vector per sample. The dimension of the feature vector is then reduced to p, by performing a principal component analysis (PCA). The principal components are the leading $p = \min(m, P)$ eigenvectors computed from the matrix containing $m = Kn$ feature vectors for training, where K is the total number of phrase classes in the training set, n is the number of training samples per class, and P is a user-specified dimension. The final p-by-1 feature vector is obtained by normalizing the dimensionally reduced

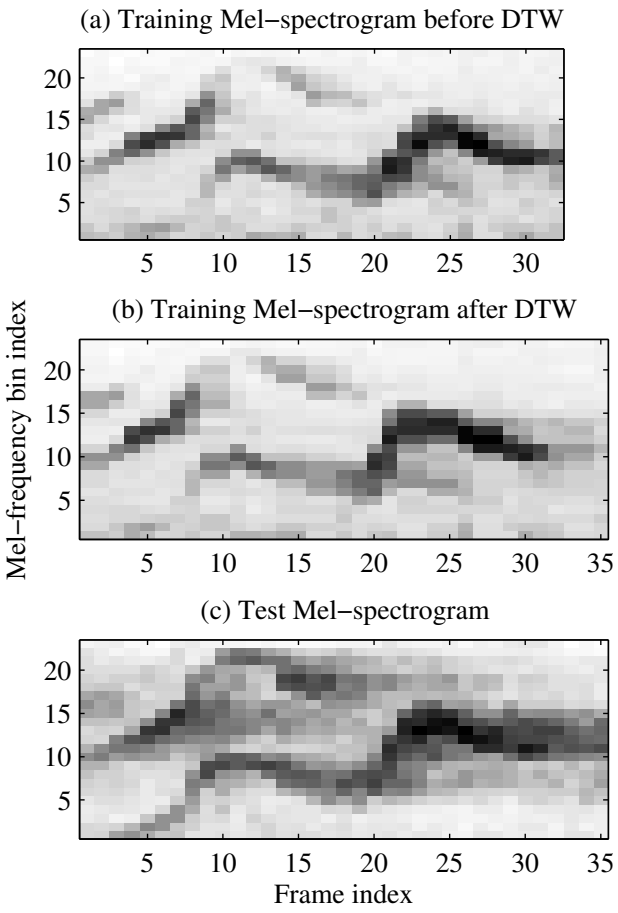

Figure 4.15: An example to illustrate the effect of DTW on the training Mel-spectrogram. (a) and (b) show the training Mel-spectrograms before and after DTW, respectively, to match the test Mel-spectrogram shown in (c). Amplitude log-compression has been applied to give a clearer visual display of the spectrograms in this figure.

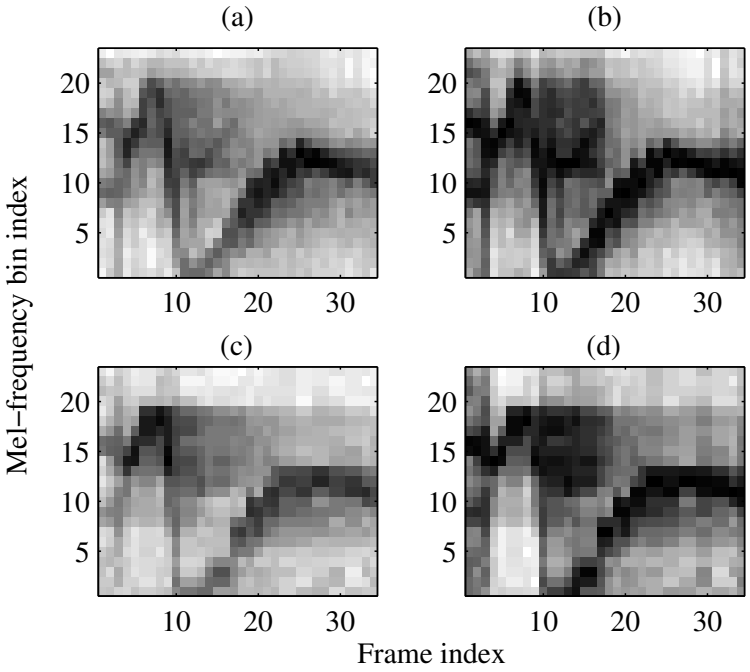

Figure 4.16: An example to show that frame normalization improves within-class spectral shape similarity. (a) and (c) show the log Mel-spectrograms of two phrases from the same class without frame normalization, while (b) and (d) show the frame-normalized log Mel-spectrograms of the same phrase segments on their respective left.

vector to unit length (this is done in both training and test sets).

The SR classification algorithm is the same as that described in Section 4.2.2. Dimension reduction ensures that Eq. (4.3) is not an over-determined linear system of equations, such that a solution x, can always be found. The output decision of the SR classifier, O_{SR} in Eq. (4.6), is the class decision of the 1st-pass SR classification. O_{SR} is compared with O_{DTW} that is computed during DTW. If they are the same, it is the final class decision of the proposed algorithm, i.e. $O_{\mathrm{SR2}} = O_{\mathrm{SR}} = O_{\mathrm{DTW}}$. The spg_bpdn function in the SPGL1 MATLAB toolbox [115,116] is used to solve Eq. (4.3). The SPGL1 toolbox's l_1 optimization

is more speed-efficient than l_1-MAGIC's (the toolbox that was used in the earlier study), while achieving similar classification performance.

4.3.1.5 2nd-pass of SR classification

If $O_{\text{DTW}} \neq O_{\text{SR}}$, a second (2nd)-pass of SR classification is activated. SR classification in the 2nd-pass is performed using only the training samples from the two conflicting phrase classes, and with a modification to the feature normalization step. Instead of frame normalization, frequency bin normalization is performed on the time-warped training Mel-spectrograms and the original test Mel-spectrogram, such that the Euclidean norm of the values in each frequency bin is equal to one. This provides a different perspective of the features to the SR classifier in the 2nd-pass. Experimental results show that normalizing by frequency bin in the 2nd-pass improves the classification accuracy of the 1st-pass by more than 2% on average, while using frame normalization in the 2nd-pass gives an improvement of less than 1%. Subsequent steps after feature normalization follow the same procedure as that used in the 1st-pass of SR classification, which include log-compression, dimension reduction, and sparse vector computation. The class with the smaller residual norm of the two classes in the 2nd-pass, is the final class decision of the proposed algorithm. Note that the computation time required to perform the l_1-minimization optimization in the 2nd-pass is much less than that used in the 1st-pass, since the dictionary matrix involved is significantly smaller.

Fig. 4.17 shows an example of a test sample that is misclassified in the 1st-pass of SR classification, but correctly classified in the 2nd-pass. In the 1st-pass, the SR classifier misclassified the test sample to the phrase class of a training sample whose frame-normalized, log Mel-spectrogram resembles its own, except for the rising frequency trend between frames 5 and 10. This can be observed by com-

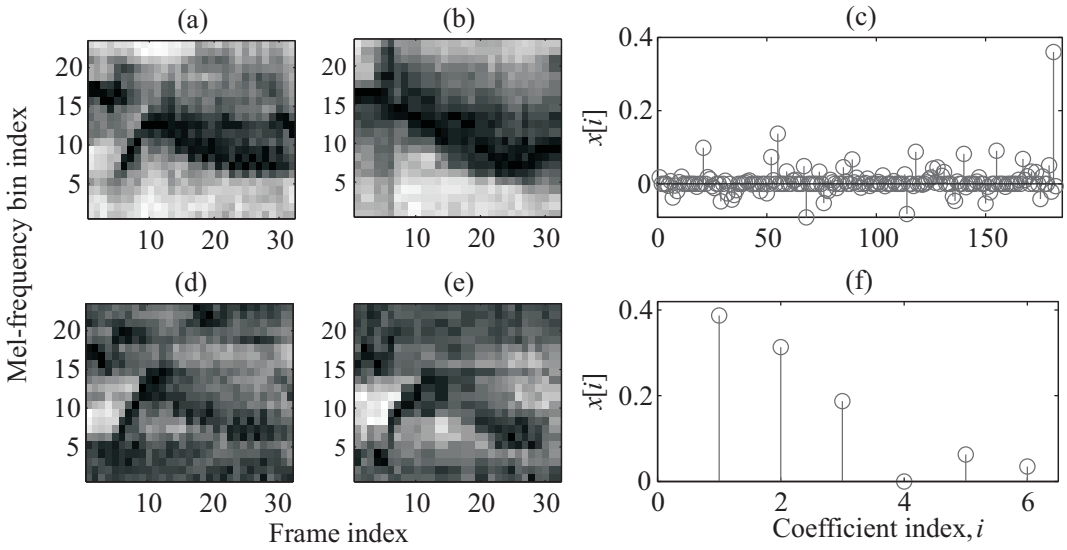

Figure 4.17: An example to illustrate the advantage of performing a 2nd-pass of SR classification with features derived from frequency bin-normalized log Mel-spectrograms. The Mel-spectrograms of a test sample with frame normalization, and frequency bin normalization, are shown in (a) and (d), respectively. (b) Frame-normalized Mel-spectrogram of the training sample that corresponds to the largest coefficient of the sparse solution vector plotted in (c), which is obtained in the 1st-pass of SR classification. (e) Frequency bin-normalized, Mel-spectrogram of the training sample that corresponds to the largest coefficient of the sparse solution vector plotted in (f), which is obtained in the 2nd-pass of SR classification.

paring the test Mel-spectrogram in Fig. 4.17a to the training Mel-spectrogram in Fig. 4.17b, which corresponds to the largest x coefficient (in Fig 4.17c) computed in the 1st-pass. On the other hand, the DTW classifier gives the correct class decision in this case. With frequency bin normalization performed in the 2nd-pass, the SR classifier is able to correctly classify the test sample. The frequency bin-normalized, log Mel-spectrogram of the same test sample is shown in Fig. 4.17d. Fig. 4.17e shows the frequency bin-normalized, log Mel-spectrogram of the training sample that corresponds to the largest x coefficient (in Fig 4.17f) computed in the 2nd-pass. Note that three training samples per phrase class is used in this example, and the three largest x coefficients in Fig 4.17f correspond to training samples of the same phrase class as the test sample.

The parameters of the proposed DTW-SR-2pass algorithm are tuned using the set of values listed in Table 4.3. The classification results of the best configuration is presented in Section 4.3.4, while performance variations with different parameter values are presented in Section 4.3.5.3.

Table 4.3: The various parameter values used to find the best configuration of the proposed algorithm

Parameter	Values
PCA dimension, P	32, 64, 128
Difference tolerance, ε	0.025, 0.05, 0.1, 0.2
DTW frame extension, T	0, 5, 10, 20, 30, 40

4.3.2 Training and test data

4.3.2.1 Manually-segmented CAVI phrases for training and testing

CAVI3, CAVI4, and CAVI5 in the 2012 collection are selected as the "training CAVIs" for their variety of phrase classes (refer to Fig. 4.2 for the distribution of the phrases sang by each of the six bird individual in the CAVI database), so that there is a larger number of phrase classes available to train the classifier. The remaining CAVIs are the "test CAVIs". Phrase classes with at least n samples or tokens from the training CAVIs are used for training. Let K be the total number of classes that satisfies this condition. In each experiment, n training samples from the training CAVIs in each of these K classes are randomly selected, and the average results of five such experiments are reported for each test condition. Two test sets containing phrases from these K classes are formed for performance evaluation – 1) Test A contains phrases produced by the training CAVIs that are not used for training, and 2) Test B contains phrases produced by the test CAVIs. Table 4.4 shows the value of K as n varies from one to five, and the number of *manually-segmented* test phrases in each of the two test sets.

4.3.2.2 Test segments obtained with an automatic detection and segmentation algorithm

Besides evaluating the classification accuracy on segments whose time boundaries are derived from human annotations, performance evaluation is also conducted on test segments whose time boundaries are derived from an automatic detection and segmentation algorithm. The bird syllable detection and segmentation algorithm selected is the time-frequency energy-based algorithm proposed in [54], which is also used in [117, 118]. The energy threshold values in a MATLAB code [119] of

Table 4.4: The number of classes found in the training set, K, that has at least n training samples per class, for $n = 1, 2, ..., 5$. The total number of manually-segmented test phrases in Test A and Test B at each value of n are also tabulated.

n	K	No. of manually-segmented test phrases in	
		Test A	Test B
1	81	419	1269
2	67	352	1208
3	61	291	1138
4	50	241	1061
5	43	198	821

this algorithm is modified such that more than 90% of the phrase segments are detected, with almost all time boundaries staying within 100 ms of the human detected time boundaries. The details of this syllable detection and segmentation algorithm are described below:

1. Compute the linear frequency power spectrogram, as described in Section 4.3.1.1, for each 10-second file segment. Extract the frequency bins between 1.5 and 6.5 kHz, and convert the spectrogram amplitudes to dB, i.e. $Q(f,t) = 10\log_{10}|S(f,t)|^2$, where $|S(f,t)|^2$ is the power spectrogram, with f and t denoting the frequency and frame indices, respectively.

2. Find the maximum value in each frame, $q(t) = \max_f Q(f,t)$.

3. Initialize the syllable detection frame vector, $r(t) = 0, \forall t$

4. Set $n = 0$.

5. Find the maximum value in $q(t)$ and its corresponding frame index, such

that $a_n = \max_t q(t)$, and $t_n = \operatorname{argmax}_t q(t)$

6. Find the n-th syllable boundary by increasing t_1 and t_2 separately until $q(t_n - t_1) < a_n - 15$ and $q(t_n + t_2) < a_n - 15$. The n-th syllable segment detected will be located at frames, $\tau_n = [t_n - t_1 + 1, ..., t_n + t_2 - 1]$.

7. Set $r(\tau_n) = 1$ and $q(\tau_n) = -\infty$. Increase n by 1.

8. Repeat steps 5 – 7, till $a_n < a_0 - 15$. This will yield n syllable segments.

This syllable detection and segmentation algorithm is performed separately for every 10-second segment of each sound file, assuming that there are phrases present in every 10-second long segment. Since the algorithm's detection threshold is dependent on a_0 – the global maximum energy of the input spectrogram, dividing each sound file into smaller chunks (some of which are more than 5 minutes in length) improves the detection rate, when there is energy variation across different parts of the sound file. After the above detection and segmentation algorithm is performed on all 10-s segments (the last file segment can be less than 10 s long) in the sound file, the syllable detection frame vector, $r(t)$, of adjacent segments are concatenated, and a 21-point median smoothing is performed on this concatenated vector. Two successive detected syllables are merged into a single detected segment if the in-between pause is less than 0.5 s and the sum of these two syllables' durations. Only segments of duration between 0.1 and 1.5 s are retained. This is slightly wider than the range of phrase duration noted in human annotations, to allow for some errors in machine-segmented time boundaries. Finally, these automatically detected segments are each considered a valid phrase detection, if it has some time overlap with a manually-segmented phrase. These automatically detected and segmented phrases are referred to as the "machine-segmented" phrase segments in this dissertation. Histograms of the time dif-

ference between the machine-segmented and human-annotated time boundaries (machine-segmented time boundary minus human-annotated time boundary) for all machine-segmented phrase segments of the training CAVIs and test CAVIs are plotted in Fig. 4.18. Figs. 4.18a and 4.18b show the histograms of time differences observed in training CAVIs' phrases at the left/starting and right/ending time boundaries, respectively; while Figs. 4.18c and 4.18d show the histograms of time differences observed in test CAVIs' phrases. All histograms are roughly Gaussian in shape, with 35–60% of the time differences falling within -0.025 to 0.025 s, and about 5% of the machine-segmented phrase segments has a time boundary difference exceeding 0.2 s.

When machine-segmented phrase segments are used for training, it yields about 2% lower classification accuracy on the machine-segmented test phrases (for all classification algorithms evaluated), compared to using manually-segmented training phrases. Hence, manually-segmented phrase segments are still used to train the classifiers in the performance evaluation of machine-segmented test phrase segments. Table 4.5 shows the number of *machine-segmented* test phrase segments/samples in Test A (from the training CAVIs) and Test B (from the test CAVIs), as n varies from one to five. Care is taken to ensure that the machine-segmented training CAVIs' test phrase segments are distinct from the manually-segmented phrases in the training set.

4.3.3 Comparative algorithms

4.3.3.1 Dynamic time warping (DTW) classifier

The dynamic time warping-based classifier used for performance comparison is the same as the one described in Section 4.3.1.2, in which the cosine similarity measure is used to compute the similarity between the Mel-spectrogram frames of

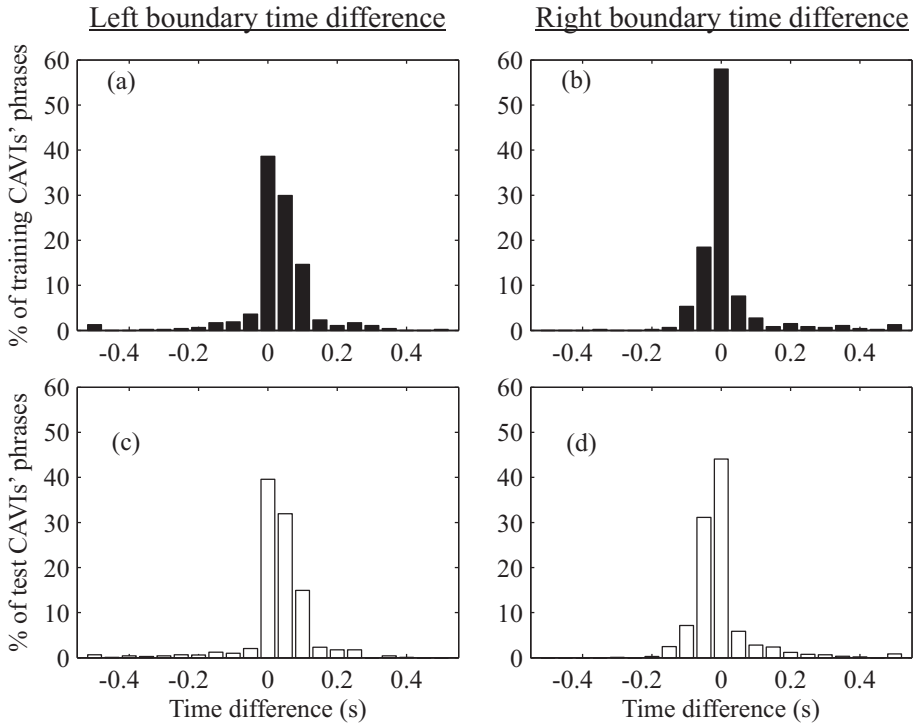

Figure 4.18: Histograms of time boundary differences between machine-segmented and manually-segmented phrases from the training and test CAVIs. (a) and (b) show the histograms of left and right boundary time differences observed in phrases of the training CAVIs, respectively. (c) and (d) show the histograms of left and right boundary time differences observed in phrases of the test CAVIs, respectively.

the training sample and the test sample. The classification accuracy is calculated based on the decision of the DTW classifier, O_{DTW}.

4.3.3.2 Sparse representation (SR) classifier

This SR classifier is similar to our previous implementation in [72], in which no DTW is involved. Instead, a phrase-duration-dependent frame shift is calculated, so that the spectrogram of each phrase segment (of variable duration)

Table 4.5: The total number of machine-segmented test phrases from the K training classes in Test A and Test B, as n varies between 1 to 5. K is the number of classes found in the training set that has at least n training samples per class.

n	K	No. of machine-segmented test phrases in	
		Test A	Test B
1	81	396	1218
2	67	336	1159
3	61	277	1049
4	50	228	1021
5	43	190	788

always contains 64 frames in time. A 64-frame Mel-spectrogram is computed, followed by the frame normalization, log-compression, dimension reduction, and l_1-minimization schemes described in Sections 4.3.1.3 and 4.3.1.4. The class that yields the minimum residual norm, is the decision of this SR classifier.

4.3.3.3 DTW-SR-1pass classifier

This DTW-SR-1pass classifier is similar to our proposed DTW-SR-2pass algorithm, but without the 2nd-pass of SR classification. The decision, O_{SR} (see Section 4.3.1.4), is used to compute the classification accuracy of this DTW-SR-1pass classifier. It enables assessment of the performance gain obtained with the additional 2nd-pass of SR classification.

4.3.3.4 Support vector machine (SVM) classifier

The multi-class support vector machine (SVM) classifier is implemented using a popular SVM software library known as LIBSVM [108]. Like the SR classifier described in Section 4.3.3.2, this SVM classifier does not incorporate DTW, and a frame-normalized, log Mel-spectrogram with 64 frames is computed. The input feature vector to the SVM is the frame-concatenated vector of this Mel-spectrogram. No dimension reduction is performed and the linear kernel is used because experiments show that this configuration yields better classification performance, compared to using dimension reduction or the Gaussian radial basis function (RBF) kernel, under the limited training data condition. The innate multi-class SVM classifier in LIBSVM uses a one-against-one decomposition scheme. However, this one-against-one decomposition scheme breaks down when there are less than 3 samples per class to train the SVM. Hence, the one-against-all decomposition strategy is implemented based on a code modification found in the list of LIBSVM's frequency asked questions (FAQ). The optimum regularization factor (a.k.a soft margin parameter), C, is determined from $\{2^{-1}, 2^{0}, ..., 2^{3}\}$, by evaluating which of these values gives the highest classification accuracy on the training samples.

4.3.3.5 DTW-SVM-2pass classifier

The DTW-SVM-2pass classifier is similar to the proposed DTW-SR-2pass classifier. The SR classification stage (including dimension reduction) in each pass is replaced by the linear SVM classifier described in Section 4.3.3.4. It enables assessment of the performance difference between the two classification techniques (SR and SVM) with the proposed DTW framework.

4.3.4 Results

4.3.4.1 Classification accuracy on manually-segmented test phrases

In this section, the classification accuracies (Acc) of the classifiers on manually-segmented test phrases are presented. Table 4.6 shows the Acc of the comparative algorithms and the proposed DTW-SR-2pass classifier. The values of P, ε and T for the DTW-SR-based classifiers that give the best performance on this dataset are 128, 0.05, and 5, respectively. The results of the DTW and DTW-SVM-2pass classifiers shown are obtained with T set to 5 (which is also the best value for each classifier). For the SR classifier, the best results shown is obtained with P = 128 and $\varepsilon = 0.1$.

The results in Table 4.6 show that the proposed DTW-SR-2pass classifier achieves the highest classification accuracies on manually-segmented test phrases from Tests A and B. Both SVM and DTW-SVM-2pass classifiers perform worse than their SR-based counterparts by about the same amount of degradation, which are $\approx 4\%$ lower in averaged (across n) Acc for Test A, and $\approx 3\%$ lower in averaged Acc for Test B.

It is also observed that the SR and SVM classifiers that do not have DTW incorporated, perform worse. When DTW is applied on the training Mel-spectrograms, the averaged Acc of the DTW-SR-1pass classifier is higher than the SR classifier's by about 3% for Test A, and more than 13% for Test B. The averaged Acc of the DTW-SR-1pass and DTW-SR-2pass are also better than the DTW classifier's. With the additional 2nd-pass SR classification, the averaged Acc of DTW-SR-2pass increases by more than 2% over those achieved by DTW-SR-1pass, on both test sets.

Table 4.6: Acc (%) obtained by the different classifiers on manually-segmented test phrases in test sets A and B, as the number of training samples per class, n, varies between 1 to 5.

Manually-segmented test phrases in Test A						
Algorithm	n=1	n=2	n=3	n=4	n=5	Average
DTW	76.0	85.5	91.2	94.4	96.5	88.7
SR	73.7	84.4	88.6	92.0	95.7	86.9
SVM	68.9	80.4	85.0	88.5	91.7	82.9
DTW-SVM-2pass	75.5	86.4	91.8	94.4	95.4	88.7
DTW-SR-1pass	74.1	88.4	92.9	96.4	98.3	90.0
DTW-SR-2pass	**80.7**	**90.9**	**94.8**	**96.8**	**98.6**	**92.4**
Manually-segmented test phrases in Test B						
Algorithm	n=1	n=2	n=3	n=4	n=5	Average
DTW	73.7	83.3	89.9	91.3	92.2	86.1
SR	60.0	71.5	77.8	80.9	81.1	74.2
SVM	58.5	69.4	74.5	77.1	78.1	71.5
DTW-SVM-2pass	75.3	85.8	91.1	92.4	92.5	87.4
DTW-SR-1pass	74.6	86.3	91.2	93.4	94.1	87.9
DTW-SR-2pass	**80.3**	**90.4**	**93.6**	**94.3**	**94.3**	**90.6**

Table 4.7: Acc (%) obtained by the different classifiers on machine-segmented test phrases in test sets A and B, as n varies between 1 to 5.

Machine-segmented test phrases in Test A						
Algorithm	n=1	n=2	n=3	n=4	n=5	Average
DTW	60.9	72.7	77.9	85.1	86.9	76.7
SR	36.2	44.4	45.9	49.8	52.3	45.7
SVM	34.9	43.5	43.9	46.8	50.3	43.9
DTW-SVM-2pass	62.7	76.3	84.4	87.8	90.0	80.2
DTW-SR-1pass	62.5	80.1	85.4	91.4	93.5	82.6
DTW-SR-2pass	**68.0**	**81.3**	**86.4**	**91.7**	**93.7**	**84.2**
Machine-segmented test phrases in Test B						
Algorithm	n=1	n=2	n=3	n=4	n=5	Average
DTW	58.2	69.6	76.9	79.5	81.8	73.3
SR	34.9	42.2	44.6	49.4	51.6	44.6
SVM	32.5	38.6	40.4	43.2	46.4	40.2
DTW-SVM-2pass	62.7	74.4	83.8	86.0	86.4	78.7
DTW-SR-1pass	62.6	76.0	85.2	86.5	88.0	79.7
DTW-SR-2pass	**67.5**	**80.2**	**86.0**	**88.8**	**89.4**	**82.4**

4.3.4.2 Classification accuracy on machine-segmented test phrases

Classification accuracies (Acc) of the comparative algorithms and the proposed DTW-SR-2pass on machine-segmented test phrases are tabulated in Table 4.7. The optimum values of P and ε for the SR and DTW-SR-based classifiers are the same as those for the manually-segmented test phrases (in Section 4.3.4.1). For the DTW frame extension parameter, T, it is found that increasing it from 5 to 30 frames improves the classification accuracy of the machine-segmented bird phrase segments whose time boundaries are less consistent than those obtained from human annotators. The results of the DTW and DTW-SVM-2pass classifiers shown are also obtained with T set to 30.

The classification accuracies in Table 4.7 on machine-segmented test phrases show similar performance trends as those observed in Table 4.6. The proposed DTW-SR-2pass classifier has the highest averaged Acc for both CAVI test sets, followed by DTW-SR-1pass, DTW-SVM-2pass, DTW, SR and SVM.

The classification accuracies on machine-segmented test phrases are lower than those obtained for manually-segmented test phrases. This is mainly due to time boundary errors present in the machine-segmented phrase segments. A shorter segment with missing portions of the test phrase would lead to misclassified to classes that are more similar to the detected sub-segment. For example, the phrase shown in the top left corner of Fig. 4.4 may be classified to the phrase on its right, if some of its beginning and end portions are missing. A longer phrase segment might also be misclassified if there are substantial background noise in the additional frames detected.

4.3.5 Discussion

4.3.5.1 On the importance of DTW

On the manually-segmented test phrases, the SR and SVM classifiers without DTW perform worse than a DTW classifier, with a larger Acc performance degradation observed on phrase segments in Test B than those in Test A. This shows the benefit of using DTW to reduce the mismatch between the training and test data. Error analysis reveals that such data mismatch exists due to individual bird variations in producing phrases from the same class, environmental differences in the source to microphone propagation (due to different sound propagation distance, and reverberation in dense vegetation), and the subjective differences between human annotators in determining the phrase segment boundaries. Fig. 4.19 shows a time mismatch between Mel-spectrograms from the same phrase class when a phrase-duration-dependent frame shift is used to generate Mel-spectrograms with a fixed number of frames. The human-annotated time boundary at the end of the phrase in Fig. 4.19b is marked earlier than that in Fig. 4.19a because the phrase energy towards the end of the spectrogram reached a human-determined intensity threshold at that point.

The importance of DTW in reducing time boundary mismatch is further exemplified on automatically detected and segmented bird phrases. The SR and SVM classifiers without DTW suffer a large performance degradation (30 – 45% absolute decrement in Acc, in contrast to the 5 – 15 % decrement observed in classifiers with DTW) in comparison to their performance on manually-segmented bird phrases. This is because only those machine-segmented phrase segments with time boundaries that are very close to their manually-segmented counterparts are correctly classified.

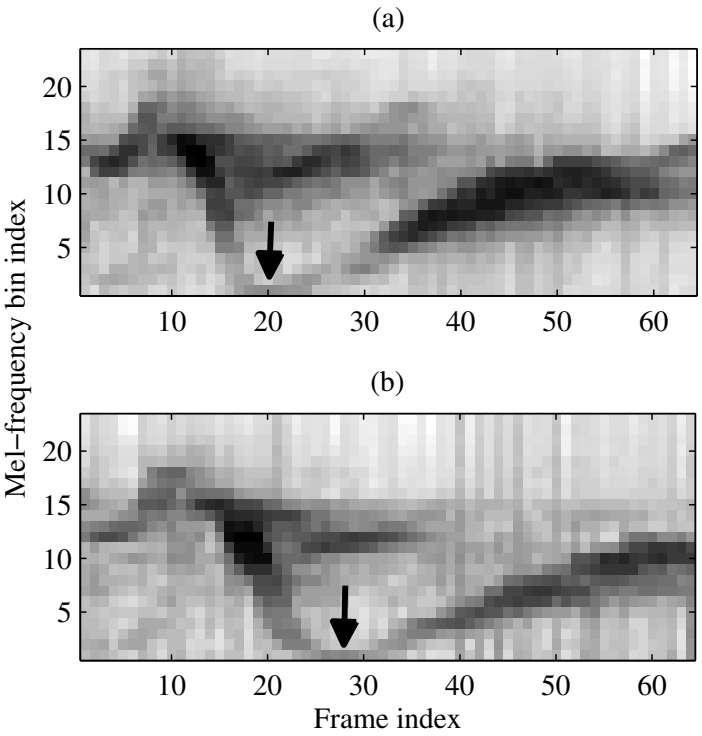

Figure 4.19: Time mismatch observed between Mel-spectrograms of two manually-segmented phrases from the same class that are generated using a phrase-duration-dependent frame shift. The white arrows on these 64-frame Mel-spectrograms in (a) and (b) note the time instances of the low frequency regions in the phrase segments.

4.3.5.2 On the performance gain of DTW-SR-2pass over DTW-SR-1pass

When the results of DTW-SR-1pass classifier are compared with those of the DTW-SR-2pass classifier, it is evident that the largest performance gain is observed in the case when there is only $n = 1$ training sample per phrase class. The 2nd-pass of the SR classification aims to resolve the conflicting class decisions of DTW-SR-1pass and DTW. The DTW classifier compares the test sample to every

training sample separately, and obtains the correct decision if there is a training sample in the correct class that has the highest cosine similarity with the test sample. In contrast, the SR classifier generally classifies correctly if the sparse solution in Eq. (4.3) is concentrated at the coefficients corresponding to the correct phrase class. When there are more than one training samples per class, the SR classification framework has the advantage of using a linear combination of (all or a few) training samples from the correct phrase class (together with some training samples from other classes) to reconstruct a vector that resembles the test feature vector. When there is only one training sample per phrase class, the advantage of DTW-SR-1pass over DTW with a linear combination framework is largely reduced. Even if the single training sample from the correct class is closest to the test sample, but not by a large margin compared to other training sample, there is a larger possibility of finding a sparse linear combination of training samples from multiple classes that resembles the test sample, in which the coefficient corresponding to the correct class is not the largest. Analysis reveals that among the conflicting class decisions in which either DTW or DTW-SR-1pass is correct, DTW-SR-1pass is correct for approximately 77% of them in the case when $n = 5$, and this percentage decreases to 53% when $n = 1$. When only the two conflicting classes' training feature vectors (obtained from frequency bin-normalized Mel-spectrograms) are used in the 2nd-pass of SR classification, the SR classifier is able to derive a correct class decision for 70–90% of these cases.

4.3.5.3 Parameter sensitivity of DTW-SR-2pass's classification performance

The proposed DTW-SR-2pass classification performance is relatively consistent compared to DTW-SR-1pass, across the different parameter values of P (PCA di-

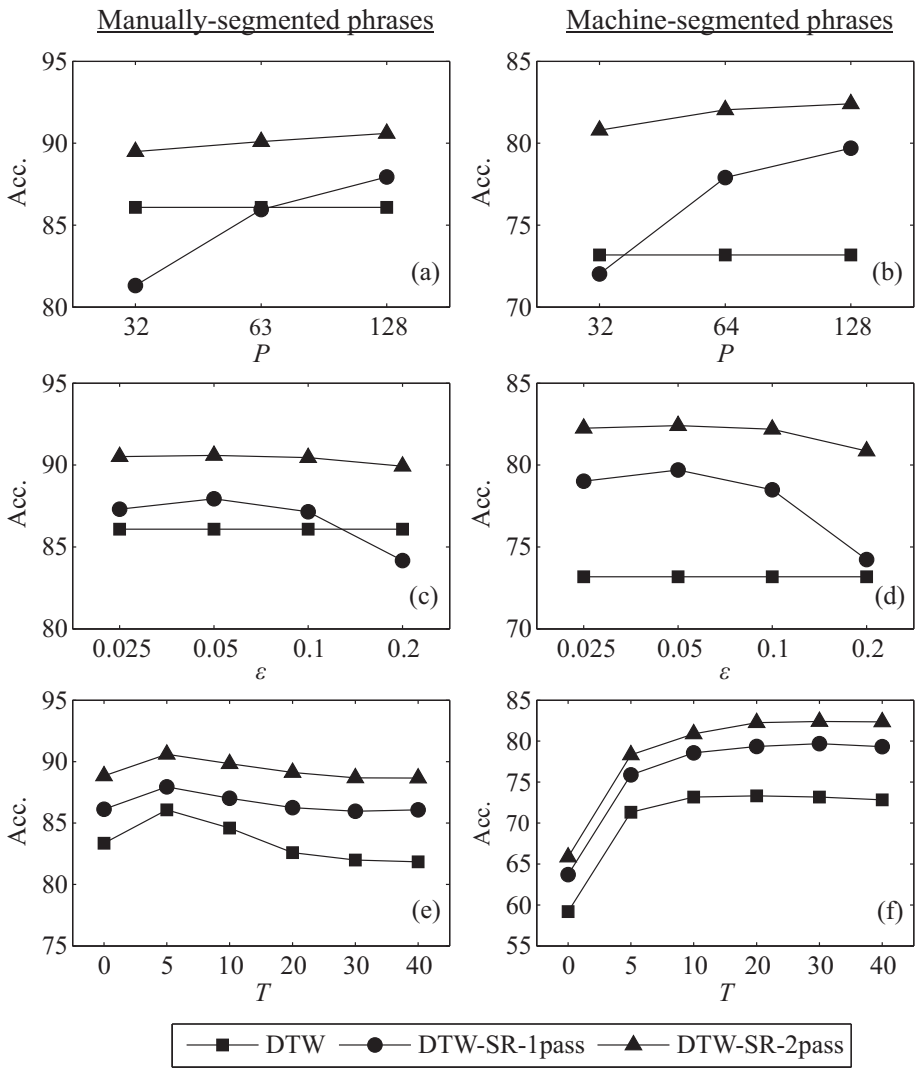

Figure 4.20: Sensitivity of DTW-SR-2pass's classification accuracy to different values of PCA dimension (P), difference tolerance (ε), and DTW frame extension (T). The averaged Acc (averaged across n) obtained on manually-segmented test CAVIs' phrases by the DTW, DTW-SR-1pass, and DTW-SR-2pass classifiers with different values of P, ε, and T, are plotted in (a), (c) and (e), respectively. The classifiers' averaged Acc on machine-segmented test CAVIs' phrases as P, ε, and T vary, are plotted in (b), (d) and (f), respectively.

mension) and ε (difference tolerance) investigated. Figs. 4.20a and 4.20b plot the averaged Acc (averaged across n) achieved with different P values for manually-segmented and machine-segmented test CAVIs' phrase segments (in Test B), respectively, while ε and T (DTW frame extension) are fixed at their optimal value. Similarly, Figs. 4.20c and 4.20d plot the Acc variations with ε (while P and T are fixed at their optimal value), for manually-segmented and machine-segmented test CAVIs' phrase segments, respectively. The Acc of the DTW classifier is constant in Figs. 4.20a–d because its algorithm is independent of P and ε. The Acc range of DTW-SR-2pass is less than 2% in the range of P and ε evaluated, while the Acc range of DTW-SR-1pass is larger. There are also some values of P and ε for which DTW-SR-1pass performs worse than the DTW classifier. The reduced Acc variation observed for the proposed DTW-SR-2pass classifier is mainly due to the 2nd-pass stage, which takes into account the class decision of the DTW classifier, whose performance does not change with P and ε.

Figs. 4.20e and 4.20f plot the averaged Acc achieved by DTW, DTW-SR-1pass and DTW-SR-2pass with different T values (while P and ε are fixed at their optimal value), for manually-segmented and machine-segmented test CAVIs' phrases, respectively. As T varies, the Acc variations of DTW, DTW-SR-1pass and DTW-SR-2pass are similar to one another. Comparing these figures, we conclude that increasing T can result in performance degradation (about 2%) for manually segmented phrases. For machine-segmented phrases, on the other hand, increasing T beyond 0 is beneficial.

4.3.5.4 Real-time performance of DTW-SR-2pass

Table 4.8 shows the real-time (RT) performance of the proposed DTW-SR-2pass classification algorithm (written in MATLAB) on an Intel i7 2.67 GHz processor

with 4 GB RAM (without parallel-computing). It varies from 0.5 to 1.5×RT, depending on the parameter settings, and the number of training samples used. In general, the computation time increases with T because DTW is done on a larger number of frames. The computation time also increases as n and P increase because there are more elements in the dictionary matrix, A, on which the l_1-minimization in Eq. (4.3) is performed. On the other hand, the l_1-minimization requires less computation time when a larger tolerance, ε, is used.

Table 4.8: Real-time (RT) performance in ×RT of the DTW-SR-2pass classifier with various values of ε, P, T, and n.

n=3 and T=5				ε=0.05 and P=128			
$\varepsilon \diagdown P$	32	64	128	$T \diagdown n$	1	3	5
0.025	1.09	1.15	1.19	0	0.53	1.09	1.15
0.05	0.97	1.03	1.04	10	0.56	1.09	1.26
0.1	0.86	0.91	0.92	20	0.59	1.19	1.34
0.2	0.84	0.86	0.87	30	0.65	1.29	1.47

4.3.6 Conclusion

A DTW-SR-2pass classification framework that combines dynamic time warping (DTW) with a two-pass sparse representation (SR) classification is proposed for limited data birdsong phrase classification. Mel-spectrograms of the training samples are dynamically-time-warped to the Mel-spectrogram of each test sample to obtain training Mel-spectrograms that have the same number of time frames as the test Mel-spectrogram. This is followed by feature normalization, which performs frame normalization and amplitude log compression. PCA dimension reduction is performed, and the dimension-reduced feature vectors are used to

perform the 1st-pass SR classification. If the class decision of the 1st-pass SR classification is different from the decision of the DTW classifier that is computed (as a by product) during the DTW stage, a 2nd-pass SR classification is performed using the training samples solely from the two conflicting classes. Frequency bin normalization (instead of frame normalization) is performed on the Mel-spectrograms in the 2nd-pass.

The training set contains a few training samples ($n = 1 - 5$) per phrase class from a few bird individuals, and the test data is split into two sets – Test A which contains phrases sang by the same CAVI individuals found in the training set, and Test B which contains phrases sang by another group of CAVI individuals. Performance evaluations are conducted on both manually-segmented and machine-segmented phrases. Compared to the DTW, SVM-based, and SR (without DTW) classifiers evaluated, the proposed DTW-SR-2pass classifier achieves the highest classification accuracies on both manually-segmented and machine-segmented phrases. DTW helps in reducing the mismatch between the training and test phrase segments. Incorporating the 2nd-pass SR classification leads to improvement in classification performance and reduction in parameter sensitivity compared to using only a one-pass SR classification framework.

CHAPTER 5

Summary and Future Work

In this dissertation, robust algorithms for pitch detection, automatic speech recognition (ASR), and birdsong phrase classification are presented. Noise-robustness of pitch detection is achieved using multi-band summary correlograms, while sparse representations are used to improve the noise-robustness of ASR, and the accuracy of birdsong phrase classification in limited data condition.

5.1 Multi-band summary correlogram (MBSC)-based pitch detector for noisy speech

In Chapter 2, a multi-band summary correlogram (MBSC)-based pitch detection algorithm is described. The pitch detection algorithm performs pitch estimation and voiced/unvoiced (V/UV) detection via novel signal processing schemes that are designed to enhance the MBSC's peaks at the most likely pitch period. Four wideband FIR filters are used to perform frequency decomposition, such that multiple harmonics fall within each subband. This facilitates the use of signal envelopes in all subbands for pitch estimation (including the low-frequency band), since signals that contain F0-spaced harmonics will be amplitude-modulated at the rate of F0. Besides the signal envelopes, the low-frequency band signal is also retained for subsequent processing. Each of these subband signal/envelope streams are furthered filtered using a comb-filterbank, in which each channel con-

sists of a comb-filter with a particular inter-peak separation in an adult human's pitch range. Harmonic-to-Subharmonic Ratio (HSR)-based comb-channel selection and weighting schemes are performed to yield individual stream's summary correlogram. This is followed by stream-reliability-weighting to combine these summary correlograms into a single MBSC. V/UV detection is performed by applying a constant threshold on the maximum peak of the MBSC. Narrowband noisy speech sampled at 8 kHz are generated from Keele (development set) and CSTR - Centre for Speech Technology Research (evaluation set) corpora. Both 4-kHz fullband speech, and G.712-filtered telephone speech are simulated. When evaluated solely on pitch estimation accuracy, assuming voicing detection is perfect, the proposed algorithm has the lowest gross pitch error for noisy speech in the evaluation set among the algorithms evaluated (RAPT, YIN, etc.). The proposed pitch detection algorithm also achieves the lowest average pitch detection error, when both pitch estimation and voicing detection errors are taken into account.

5.2 Jointly-sparse estimated and reference soft-mask representations for noise-robust speech recognition

In Chapter 3, a novel JDictMask feature enhancement scheme that uses jointly-sparse reference soft-mask ($\mathrm{SM}_{\mathrm{ref}}$) and estimated soft-mask ($\mathrm{SM}_{\mathrm{est}}$) representations is presented. $\mathrm{SM}_{\mathrm{ref}}$ is the ratio of clean Mel-spectrum to the noisy Mel-spectrum computed from a clean and noisy utterance-pair in the training data, while $\mathrm{SM}_{\mathrm{est}}$ is the ratio of a denoised Mel-spectrum to the original noisy Mel-spectrum. The denoised Mel-spectrum is obtained by subtracting the noise spectrum (estimated using the minimum statistics noise estimation algorithm) from the original noisy spectrum. The sparse linear combination of $\mathrm{SM}_{\mathrm{est}}$ dictionary

representations that best approximates the $\mathrm{SM}_{\mathrm{est}}$ of the test speech utterance is found by solving an l_1-minimization problem. An enhanced soft-mask is generated by applying the same sparse linear combination to the $\mathrm{SM}_{\mathrm{ref}}$ dictionary representations. This soft-mask is used to enhance the Mel-spectrogram before MFCCs are extracted. Using multi-noise training and dictionary representations made up of exemplar-pairs extracted from clean and noisy utterance-pairs in the training data, the joint soft-mask-based feature enhancement scheme achieves higher word accuracies in the presence of mismatch noise-types and channel characteristics on the Aurora-2 noisy digit recognition task, compared to other existing feature enhancement techniques that utilizes Mel-spectral-based dictionary representations. When the JDictMask feature enhancement scheme is extended to the Aurora-4 large vocabulary noisy speech recognition task, a sparsity-based dictionary learning is used to derive jointly-sparse $\mathrm{SM}_{\mathrm{ref}}$ and $\mathrm{SM}_{\mathrm{est}}$ dictionaries from an initial subset of soft-mask exemplar-pairs extracted from the training set. An additional ambient noise subtraction is performed on the clean Mel-spectrogram to derive a $\mathrm{SM}_{\mathrm{ref}}$ that can better suppress noise at non-speech regions. On the Aurora-4 speech recognition task with multi-noise training, the JDictMask feature enhancement scheme also has the highest averaged word accuracy on speech corrupted with stationary and non-stationary out-of-set noise-types. In addition, it performs well on speech corrupted with relatively stationary in-set noise-types.

5.3 Exemplar-based sparse representation classifier for limited data birdsong phrase classification and/or verification

In Chapter 4, a study on the efficacy of an exemplar-based sparse representation (SR) classifier for limited data birdsong phrase classification and verification is conducted. This exemplar-based SR classification technique is first proposed for a facial image recognition task [71]. The evaluation database in this study contains manually-segmented Cassins Vireo (CAVI) song phrases from two bird individuals in the 2010 collection, and only phrase samples of 32 phrase classes (a.k.a. in-set classes) are used for training. A 64-frame, dynamic range normalized spectrogram is computed from each phrase segment using a duration-dependent frame shift, followed by dimension reduction using PCA, to obtain the feature vector of each phrase segment. The SR classifier finds a sparse linear combination of exemplars (or training feature vectors) that best approximates the test feature vector, by solving an l_1-minimization problem. The phrase class that has the minimum Euclidean distance between the test vector and the vector reconstructed from that class's training vectors, is the SR classifier's decision. A threshold is applied on the sparsity concentration index (SCI) computed by the SR classifier to verify whether a phrase belongs to one of the in-set phrases classes. On the 32-class in-set phrase classification task, the SR classifier outperforms the nearest subspace (NS) and support vector machine (SVM) classifiers when three to seven training samples per phrase are used. On the joint in-set bird phrase classification and verification and classification task, in which the classifier has to classify each phrase into one of 33 phrase classes – 32 in-set classes, and 1 collective out-of-set category – the SR classifier also has the highest classification accuracy compared

to the NS and SVM classifiers, due to good performances in both classification and verification.

Subsequently, an improved exemplar-based SR classification algorithm, DTW-SR-2pass, that is robust to within-class phrase variations from bird individual differences and segmentation variabilities is proposed. It involves dynamic time warping (DTW) and a two-pass SR classification. A fixed frame shift is used to compute the Mel-spectrogram of each phrase segment. The Mel-spectrograms of all training samples undergo DTW such that they have the same number of time frames as the test Mel-spectrogram. DTW improves the similarity between training and test phrases in the presence of individual bird differences in producing phrases of the same class, and helps when the phrase segment time boundaries are inconsistently determined by a human annotator or by machine. DTW also computes a class decision as a by-product, which corresponds to the training sample that is most similar to the test sample. Frame normalization and amplitude log compression is performed on the spectrogram, followed by PCA dimension reduction and the 1st-pass SR classification. When the class decisions from DTW and the 1st-pass SR classification are different, SR classification is performed a second time with dynamically time-warped training Mel-spectrograms from these two conflicting classes. In this 2nd-pass SR classification, frequency bin normalization (instead of frame normalization) is performed on the training and test Mel-spectrograms. For training, one to five samples per phrase class from three of the six CAVI individuals in the 2010 and 2012 collections are used. Song phrases from the remaining three CAVIs are used for testing. Compared to a DTW-based classifier, SVMs, and the simple SR classifier (without DTW) used in the earlier study, the DTW-SR-2pass classifier achieves the highest classification accuracies on manually-segmented phrases, as well as automatically-segmented phrases.

5.4 Future work

5.4.1 Pitch detection

The MBSC-based pitch detector is unable to distinguish between harmonic structures from speech and non-speech. One way to improve the algorithm's pitch detection accuracy in the presence of harmonic noise is to incorporate a speech activity detector that can differentiate harmonic noise from voiced speech. One can also extend the MBSC single-pitch detection algorithm to a multi-pitch detection algorithm by incorporating a pitch continuity tracking scheme to track the pitch variations of overlapping speech from multiple speakers.

5.4.2 Feature enhancement

The current reference and estimated soft-mask dictionary representations are obtained assuming that the clean and noisy training utterance-pairs are time-synchronized and that the noise is additive in nature. This leads to the assumption that the spectrogram amplitudes of the clean training utterance are always smaller than the spectrogram amplitudes in the corresponding noise-corrupted time-frequency regions of the noisy version. As such, the values in the reference soft-mask exemplars and the reconstructed reference soft-mask (that is used for Mel-spectrogram denoising) are bounded to the range between 0 and 1 in the JDictMask feature enhancement scheme. Future work can investigate the use of the JDictMask feature enhancement scheme on noisy speech that has been transmitted through a communication channel. The above assumptions might no longer be true for these speech signals due to transmission delay, automatic gain control, and channel distortion. In this case, the clean and noisy training utterance-pairs would have to be pre-processed prior to reference soft-mask com-

putation. For example, the clean and noisy utterance-pairs would have to be time-synchronized, followed by spectrogram amplitude adjustment or soft-mask dynamic range normalization, to ensure that the reference soft-mask values are nearly zero for noise-dominated regions, and nearly one for noise-free regions.

5.4.3 Birdsong phrase classification

Besides phrase classification of a single bird species, future work can investigate and extend the use of the proposed sparse representation framework to bird songs and/or species recognition. Noise-robustness of the classification algorithm can also be improved by either denoising the spectrogram prior to feature extraction, or extracting only the reliable time-frequency spectrographic features for classification and verification.

References

[1] C. K. Catchpole and P. J. B. Slater, *Bird Song: Biological Themes and Variations*. Cambridge University Press, 2008.

[2] B. McCowan, L. R. Doyle, J. Jenkins, and S. F. Hanser, "The appropriate use of Zipf's law in animal communication studies," *Animal Behaviour*, vol. 69, pp. F1–F7, 2005.

[3] D. Talkin, "Robust algorithm for pitch tracking," *Speech Coding and Synthesis*, pp. 497–518, 1995.

[4] A. Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America (JASA)*, vol. 111, pp. 1917–1930, 2002.

[5] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 22, no. 5, pp. 353 – 362, oct. 1974.

[6] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *Proc. ICASSP*, 2002, pp. 333–336.

[7] T. Nakatani and T. Irino, "Robust and accurate fundamental frequency estimation based on dominant harmonic components," *JASA*, vol. 116, pp. 3690–3700, 2004.

[8] A. Camacho and J. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *JASA*, vol. 124, pp. 1638–1652, 2008.

[9] J. C. R. Licklider, "A duplex theory of pitch perception," *Experientia*, vol. 7, pp. 128–134, 1951.

[10] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," *Auditory Physiology and Perception*, vol. 83, pp. 429–446, 1992.

[11] M. Slaney and R. F. Lyon, "A perceptual pitch detector," in *Proc. ICASSP*, 1990, pp. 357–360.

[12] J. Rouat, Y. Liu, and D. Morissette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech," *Speech Communication*, vol. 21, pp. 191–207, 1997.

[13] M. Wu, D. Wang, and G. Brown, "A multipitch tracking algorithm for noisy speech," *TSAP*, vol. 11, pp. 229–241, 2003.

[14] B. Delgutte, "Representation of speech-like sounds in the discharge patterns of auditory-nerve fibers," *JASA*, vol. 68, pp. 843–857, 1980.

[15] R. Meddis and M. J. Hewitt, "Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification." *JASA*, vol. 89, pp. 2866–2882, 1991.

[16] P. Cariani and B. Delgutte, "Neural correlates of the pitch of complex tones. I. Pitch and pitch salience," *Neurophysiology*, vol. 76, pp. 1698–1716, 1996.

[17] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *TASSP*, vol. 24, pp. 399–418, 1976.

[18] B. Secrest and G. Doddington, "Postprocessing techniques for voice pitch trackers," in *Proc. ICASSP*, 1982, pp. 172–175.

[19] S. Ahmadi and A. S. Spanias, "Cepstrum-based pitch detection using a new statistical v/uv classification algorithm," *TSAP*, vol. 7, pp. 333–338, 1999.

[20] Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals," *IEEE Trans. on Signal Processing*, vol. 39, pp. 40–48, 1991.

[21] B. Secrest and G. Doddington, "An integrated pitch tracking algorithm for speech systems," in *Proc. ICASSP*, 1983, pp. 1352–1355.

[22] I. Luengo, I. Saratxaga, E. Navas, I. Hernaez, J. Sanchez, and I. Sainz, "Evaluation of pitch detection algorithms under real conditions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. 1057 –1060.

[23] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *TASSP*, vol. 24, pp. 201–212, 1976.

[24] J. Shah, A. Iyer, B. Smolenski, and R. Yantorno, "Robust voiced/unvoiced classification using novel features and gaussian mixture model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, pp. 17–21.

[25] F. Beritelli, S. Casale, and S. Serrano, "Adaptive V/UV speech detection based on acoustic noise estimation and classification," *Electronics Letters*, vol. 43, pp. 249–251, 2007.

[26] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, 2008.

[27] R. Baraniuk, E. Candes, M. Elad, and Y. Ma, "Applications of sparse representation and compressive sensing," *Proceedings of the IEEE*, vol. 98, pp. 906–909, 2010.

[28] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[29] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, pp. 4655–4666, 2007.

[30] V. Mitra, H. Franco, M. Graciarena, and A. Mandal, "Normalized amplitude modulation features for large vocabulary noise-robust speech recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4117–4120.

[31] M. R. Schädler, B. T. Meyer, and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 131, pp. 4134–4151, 2012.

[32] B. Strope and A. Alwan, "A model of dynamic auditory perception and its application to robust word recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, pp. 451–464, 1997.

[33] C.-P. Chen and J. A. Bilmes, "Mva processing of speech features," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, pp. 257–270, 2007.

[34] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," in *Proc. Interspeech*, 2002, pp. 29–33.

[35] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *IEEE ICASSP*, 1979, pp. 208–211.

[36] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, vol. 32, pp. 1109–1121, 1984.

[37] E. S. document, *Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced Front-end feature extraction algorithm; Compression algorithm*, ETSI Std. ES 202 050 v1.1.5, ES 202 050 v1.1.5, 2007.

[38] J. van Hout and A. Alwan, "A novel approach to soft-mask estimation and log-spectral enhancement for robust speech recognition," in *IEEE ICASSP*, 2012, pp. 4105–4108.

[39] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *IEEE ICASSP*, 2013, pp. 7092–7096.

[40] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, pp. 75–98, 1998.

[41] J. C. Segura, Á. de la Torre, M. C. Benítez, and A. M. Peinado, "Model-based compensation of the additive noise for continuous speech recognition. experiments using the Aurora II database and tasks." in *INTERSPEECH*, 2001, pp. 221–224.

[42] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[43] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[44] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[45] B. J. Borgstrom and A. Alwan, "Utilizing compressibility in reconstructing spectrographic data, with applications to noise robust ASR," *IEEE Signal Processing Letters*, vol. 16, pp. 398–401, 2009.

[46] J. F. Gemmeke, H. Van Hamme, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 272–287, 2010.

[47] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2067–2080, 2011.

[48] W. Li, Y. Zhou, N. Poh, F. Zhou, and Q. Liao, "Feature denoising using joint sparse representation for in-car speech recognition," *IEEE Signal Processing Letters*, vol. 20, pp. 681–684, 2013.

[49] H. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA Tutorial and Research Workshop (ITRW) for Automatic Speech Recognition: Challenges for the new Millenium*, 2000, pp. 181–188.

[50] J. F. Gemmeke and T. Virtanen, "Artificial and online acquired noise dictionaries for noise robust ASR," in *INTERSPEECH*, 2010, pp. 2082–2085.

[51] M. Fujimoto, K. Takeda, and S. Nakamura, "CENSREC-3: An evaluation framework for Japanese speech recognition in real car-driving environments," *IEICE Transactions on Information and Systems*, vol. 89, pp. 2783–2793, 2006.

[52] T. S. Brandes, "Automated sound recording and analysis techniques for bird surveys and conservation," *Bird Conservation International*, vol. 18, pp. S163–S173, 2008.

[53] D. J. Mennill, "Individual distinctiveness in avian vocalizations and the spatial monitoring of behavior," *Ibis*, vol. 153, pp. 235–238, 2011.

[54] A. Härmä, "Automatic recognition of bird species based on sinusoidal modeling of syllables," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2003, pp. 545–548.

[55] C.-H. Lee, C.-C. Han, and C.-C. Chuang, "Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 1541–1550, 2008.

[56] V. M. Trifa, A. N. G. Krischel, and C. E. Taylor, "Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models," *J. Acoust. Soc. of Am.*, vol. 123, pp. 2424–2431, 2008.

[57] M. Graciarena, M. Delplanche, E. Shriberg, and A. Stolcke, "Bird species recognition combining acoustic and sequence modeling," in *Proc. IEEE ICASSP*, 2011, pp. 341–344.

[58] A. N. G. Kirschel, M. L. Cody, Z. T. Harlow, V. J. Promponas, E. E. Vallejo, and C. E. Taylor, "Territorial dynamics of Mexican Ant-thrushes *Formicarius moniliger* revealed by individual recognition of their songs," *Ibis*, vol. 153, pp. 255–268, 2011.

[59] J. A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study," *J. Acoust. Soc. of Am.*, vol. 103, pp. 2185–2196, 1998.

[60] L. Ranjard and H. Ross, "Unsupervised bird song syllable classification using evolving neural networks," *J. Acoust. Soc. of Am.*, vol. 123, pp. 4358–4368, 2008.

[61] K. Sasahara, M. Cody, D. Cohen, and C. Taylor, "Structural design principles of complex bird songs: A network-based approach," *PLOS ONE*, vol. 7, p. e44436, 2012.

[62] B. C. Pijanowski, L. J. Villanueva-Rivera, S. L. Dumyahn, A. Farina, B. L. Krause, B. M. Napoletano, S. H. Gage, and N. Pieretti, "Soundscape Ecology: The Science of Sound in the Landscape," *BioScience*, vol. 61, pp. 203–216, 2011.

[63] K. Ito, K. Mori, and S. Iwasaki, "Application of dynamic programming matching to classification of budgerigar contact calls," *JASA*, vol. 100, pp. 3947–3956, 1996.

[64] S. E. Anderson, A. S. Dave, and D. Margoliash, "Template-based automated recognition of birdsong syllables from continuous recordings," *J. Acoust. Soc. of Am.*, vol. 100, pp. 1209–1219, 1996.

[65] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[66] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[67] J. Laaksonen, "Subspace classifiers in recognition of handwritten digits," Ph.D. dissertation, Helsinki University of Technology, 1997.

[68] M. A. Acevedo, C. J. Corrada-Bravoc, H. Corrada-Bravob, L. J. Villanueva-Riverad, and T. A. Mitchell, "Automated classification of bird and amphibian calls using machine learning: A comparison of methods," *Ecological Informatics*, vol. 4, pp. 206–214, 2009.

[69] L. Zhang, F. Lin, and B. Zhang, "Support vector machine learning for image retrieval," in *Proc. IEEE Int. Conf. on Image Processing*, 2001, pp. 721–724.

[70] C. Willis, "Hyperspectral image classification with limited training data samples using feature subspaces," in *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 5425, 2004, pp. 170–181.

[71] A. Yang, J. Wright, Y. Ma, and S. Sastry, "Feature selection in face recognition: A sparse representation perspective," *UC Berkeley Tech Report UCB/EECS-2007-99*, pp. 1–17, 2007. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-99.html

[72] L. Tan, K. Kaewtip, M. Cody, C. Taylor, and A. Alwan, "Evaluation of a sparse representation-based classifier for bird phrase classification under limited data conditions," in *Proc. Interspeech*, 2012, pp. 2522–2525.

[73] K. Kaewtip, L. N. Tan, A. Alwan, and C. E. Taylor, "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification," in *IEEE ICASSP*, 2013, pp. 768–772.

[74] Z. Chen and R. C. Maher, "Semi-automatic classification of bird vocalizations using spectral peak tracks," *J. Acoust. Soc. Am.*, vol. 120, pp. 2974–2984, 2006.

[75] C.-H. Lee, S.-B. Hsu, J.-L. Shih, and C.-H. Chou, "Continuous birdsong recognition using gaussian mixture modeling of image shape features," *IEEE Trans. on Multimedia,*, vol. 15, pp. 454–464, 2013.

[76] I. Recommendation, *G.712: Transmission performance characteristics of pulse code modulation channels*, ITU Std., 1996.

[77] L. N. Tan and A. Alwan, "Noise-robust F0 estimation using SNR-weighted summary correlograms from multi-band comb filters," in *Proc. ICASSP*, 2011, pp. 4464–4467.

[78] P. Loughlin and B. Tacer, "On the amplitude-and frequency-modulation decomposition of signals," *JASA*, vol. 100, pp. 1594–1601, 1996.

[79] R. Drullman, "Temporal envelope and fine structure cues for speech intelligibility," *JASA*, vol. 97, pp. 585–592, 1995.

[80] D. J. Hermes, "Measurement of pitch by subharmonic summation," *JASA*, vol. 83, pp. 257–264, 1988.

[81] K. Walker and S. Strassel, "The rats radio traffic collection system," in *ISCA Odyssey*, 2012.

[82] F. Plante, G. Meyer, and W. A. Ainsworth, "A pitch extraction reference database," in *Proc. European Conf. on Speech Communication and Technology (Eurospeech)*, 1995, pp. 837–840.

[83] P. Bagshaw, S. Hiller, and M. Jack, "Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching," in *Proc. Eurospeech*, 1993, pp. 1003–1006.

[84] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, pp. 247–251, 1993.

[85] H.-G. Hirsch, "Fant – filtering and noise adding tool," 2005. [Online]. Available: http://dnt.kr.hs-niederrhein.de/download.html

[86] K. Sjölander and J. Beskow, "WaveSurfer - An Open Source Speech Tool," in *Proc. InterSpeech*, 2000, pp. 464–467.

[87] W. Chu and A. Alwan, "Reducing F0 Frame Error of F0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *Proc. ICASSP*, 2009, pp. 3969–3972.

[88] K. Oh and C. Un, "A performance comparison of pitch extraction algorithms for noisy speech," in *Proc. ICASSP*, 1984, pp. 85–88.

[89] M. E. Frerking, *Digital signal processing in communication systems*. Springer, 1994.

[90] "DARPA RATS program website." [Online]. Available: http://www.darpa.mil/Our_Work/I2O/Programs/Robust_Automatic_Transcription_of_Speech_(RATS).aspx

[91] M. Graciarena, A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. H. L. Hansen, A. Janin, B.-S. Lee, Y. Lei, V. Mitra, N. Morgan, S. O. Sadjadi, T. Tsai, N. Scheffer, L. N. Tan, and B. Williams, "All for one: Feature combination for highly channel-degraded speech activity detection," in *INTERSPEECH*, 2013, pp. 709–713.

[92] R. Martin, "Bias compensation methods for minimum statistics noise power spectral density estimation," *Signal Processing*, vol. 86, pp. 1215–1229, 2006.

[93] M. Brookes, "Voicebox: Speech Processing Toolbox for MATLAB," 2011. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[94] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2009, pp. 689–696.

[95] ——, "Online learning for matrix factorization and sparse coding," *J. Machine Learning Research*, vol. 11, pp. 19–60, 2010. [Online]. Available: http://spams-devel.gforge.inria.fr/downloads.html

[96] L. N. Tan and A. Alwan, "Feature enhancement using sparse reference and estimated soft-mask exemplar-pairs for noisy speech recognition," *accepted to IEEE ICASSP*, 2014.

[97] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[98] D. L. Donoho, V. C. Stodden, and Y. Tsaig, "About SparseLab (version 2.1)," 2007. [Online]. Available: http://sparselab.stanford.edu/

[99] E.-G. technical specification, *European digital cellular telecommunication system (Phase 1); Transmission planning aspects for the speech service in GSM PLMN system*, ETSI Std. GSM 03.50, version 3.4.0,, GSM 03.50, version 3.4.0, 1994.

[100] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, *The HTK book (for HTK version 3.4)*. Cambridge University Engineering Department, 2006.

[101] J. F. Gemmeke and H. van Hamme, "Advances in noise robust digit recognition using hybrid exemplar-based techniques," in *INTERSPEECH*, 2012, pp. 2134–2137.

[102] N. Parihar, J. Picone, D. Pearce, and H. G. Hirsch, "Performance analysis of the Aurora large vocabulary baseline system," in *EUROSPEECH*, 2004, pp. 553–556.

[103] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 357–362.

[104] CMU pronouncing dictionary. Carnegie Mellon University. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[105] C. B. Goguen and R. C. David, "Cassin's Vireo (*Vireo cassinii*), The Birds of North America Online (A. Poole, Ed.)," Ithaca: Cornell Lab of Ornithology, Cornell Lab of Ornithology, 2002. [Online]. Available: http://bna.birds.cornell.edu/bna/species/615

[106] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (Version 5.2.22) [Computer program]," 2011. [Online]. Available: http://www.praat.org/

[107] E. Candes and J. Romberg, "l_1-MAGIC: Recovery of sparse signals via convex programming," 2005, http://users.ece.gatech.edu/ justin/l1magic/. [Online]. Available: http://users.ece.gatech.edu/~justin/l1magic/

[108] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[109] T. Wu, C. Lin, and R. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[110] E. Oja, *Subspace methods of pattern recognition*. Research Studies Press, 1983, vol. 142.

[111] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, p. 153157, 1947.

[112] L. N. Tan, G. Kossan, M. L. Cody, C. E. Taylor, and A. A., "Dynamic time warping and sparse representation classification for birdsong phrase classification using limited training data," submitted.

[113] C. D. Meliza, S. C. Keen, and D. R. Rubenstein, "Pitch- and spectral-based dynamic time warping methods for comparing field recordings of harmonic avian vocalizations," *J. Acoust. Soc. Am.*, vol. 134, pp. 1407–1415, 2013.

[114] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, pp. 623–635, 1980.

[115] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, pp. 890–912, 2008.

[116] ——, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[117] C.-H. Lee, Y.-K. Lee, and R.-Z. Huang, "Automatic recognition of bird songs using cepstral coefficients," *Journal of Information Technology and Applications*, vol. 1, pp. 17–23, 2006.

[118] S.-S. Chen and Y. Li, "Automatic recognition of bird songs using time-frequency texture," in *IEEE International Conference on Computational Intelligence and Communication Networks (CICN)*, 2013, pp. 262–266.

[119] M. Lindermuth, "Harma syllable segmentation in MATLAB." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/29261-harma-syllable-segmentation