BIRD-PHRASE SEGMENTATION AND VERIFICATION: A NOISE-ROBUST TEMPLATE-BASED APPROACH

Kantapon Kaewtip[1], Lee Ngee Tan[1], Charles E.Taylor[2], Abeer Alwan[1]

[1]Department of Electrical Engineering, [2]Department of Ecology and Evolutionary Biology, University of California, Los Angeles, California, USA

kkaewtip@ucla.edu, tleengee@ee.ucla.edu, taylor@biology.ucla.edu, alwan@ee.ucla.edu

ABSTRACT

In this paper, we present a birdsong-phrase segmentation and verification algorithm that is robust to limited training data, class variability, and noise. The algorithm comprises a noise-robust, Dynamic-Time-Warping (DTW)-based segmentation and a discriminative classifier for outlier rejection. The algorithm utilizes DTW and prominent (high energy) time-frequency regions of training spectrograms to derive a reliable noise-robust template for each phrase class. The resulting template is then used for segmenting continuous recordings to obtain segment candidates whose spectrogram amplitudes in the prominent regions are used as features to a Support Vector Machine (SVM). The algorithm is evaluated on the Cassin's Vireo recordings; our proposed system yields low Equal Error Rates (EER) and segment boundaries that are close to those obtained from manual annotations and, is better than energy or entropy-based birdsong segmentation algorithms. In the presence of additive noise (-10 to 10 dB SNR), the proposed phrase detection system does not degrade as significantly as the other algorithms do.

Index Terms: bird phrase detection, limited data, dynamic time-warping, SVM, noise-robust, template-based.

1. INTRODUCTION

Birdsongs typically comprise a sequence of smaller units such as syllables and phrases. Automatic phrase or syllable detection systems of bird sounds are useful in several applications [1]. However, bird-phrase detection is challenging due to segmentation error, duration variability, limited training data, and background noise. In real recording environments, the data can be corrupted by background interference, such as rain, wind, or vocalizations of other animals, such that phrase detectors detect non-target segments [2],[3].

Most phrase detection or classification tasks consist of two components; first segmentation and then classification [2]-[10]. Several studies have proposed segmentation algorithms for birdsongs using an energy-based or entropy-based approach [8],[9]. The energy-based segmentation algorithm first locates a local maximum time-frequency bin and expands the time interval until the energy is less than a pre-defined threshold [8]. The entropy-based segmentation algorithm uses the assumption that bird calls are usually sparse, while the background noise is relatively white, i.e. short-time entropy dips when a signal is detected and rises when the signal is not detected [9],[10]. Such segmentation approaches are sensitive to background noise that have high energy and high entropy such as other animals vocalizing [8]-[10]. A classification-based segmentation has been proposed by using Random Forests to determine pixels that contain bird signals [7]. This approach requires manual annotation of binary masks for each time-frequency index which can be a difficult and consuming task. In addition, it is expensive to train all time-frequency indices with a variety of phrase duration and noise conditions (noise levels and noise types).

Template-based approaches (e.g., DTW) are appealing because the segmentation can be performed by discarding frames that are not similar to the template. Hidden Markov Models (HMMs) require many training examples to estimate their parameters, while DTW can be trained with a few training samples. Several DTW algorithms can perform segmentation in continuous recording by searching an optimal lattice path (i,j,k) where i is a frame index of the test recording that aligns with a frame index j of a template k [12] - [15]. This approach requires many templates per class to represent the variability (silence, noise, or garbage models). The computation increases significantly with the number of training samples. Moreover, it is sensitive to noise and demands "a low-clutter, low noise environment" [15]. Such a template-based approach may perform accurate segmentation by introducing a noise-robust component but it might not be sufficiently discriminative [3].

Several approaches employ discriminative classifiers such as Support Vector Machines (SVMs) and decision trees, all of which also need accurate segmentation (e.g., energy-based segmentation or manual annotations) as preprocessing [5]-[7], [11]-[13]. These classifiers require the same feature dimension for all training and test samples, and it is usually achieved by extracting descriptive features such as frequency-range, spectral flatness, time-duration or by resampling the spectrogram segments to equal length. However, this processing is sensitive to accurate segment boundaries and feature-shifting if no time alignment is performed [3].

In summary, template-based approaches can perform reliable segmentation but such generative models generally lack discriminative power. Discriminative-model approaches, on the other hand, train their models such that they focus on class separablility. In this paper, we take advantage of both approaches; the template-based approach is enhanced by integrating with an independent discriminative classifier. Specifically, we extend our previous noise-robust DTW-based phrase classifier which originally requires pre-segmentation to a phrase detection algorithm that can automatically perform template-based segmentation quite reliably even in noisy conditions [3]. The computational requirement is also less than other template-based approach–only one template is required for each class and silence or garbage models are not needed. The extra benefit is that the dimension of the generated segments are always the same as the template and the features are properly-aligned with respect to the model reference; the spectrogram values can be readily used as feature vectors to a discriminative classifier.

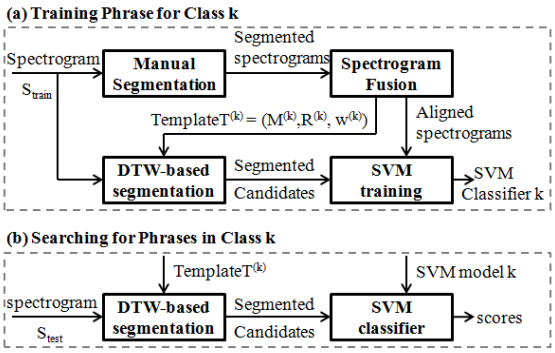

Fig. 1: Overview of the proposed algorithm

2. PROPOSED ALGORITHM

The algorithm consists of three main components (Fig. 1): Template Derivation (Section 2.2), DTW-based segmentation (Section 2.3) and SVM classification (Section 2.4). The training and testing procedures (Sections 2.4 and 2.5) are performed individually for each phrase class.

2.1 Preprocessing

The spectrographic feature extraction is similar to our previous work but with different resolution parameters to reduce the computation complexity of our system [3]. Short-time 512-point FFT was performed using a 25ms Hamming window advanced by 10 ms; only the magnitude information is retained. To reduce dimensionality, a 64-uniformly-spaced rectangular filterbank is applied and the first 7 bins corresponding to frequencies below 1 kHz are discarded, resulting in 57-frequency-bin spectrograms.

2.2 Deriving class templates

The Template Derivation algorithm has a similar implementation as our previous work with additional intermediate variables used to train SVM [3]. For a given phrase *k*, the class samples are obtained from manual annotations. Spectrograms of the segments samples are passed to the Spectrogram-Fusion Algorithm (SFA) which derives a template that represents common features among training samples of the same class. A template is a collection of three attributes; a spectrogram reference $M^{(k)}$, a prominent region $R^{(k)}$ and a frame-weighting function $w^{(k)}$. The spectrogram reference represents the time-frequency energy pattern of clean signals. The prominent region indicates which pixels are used to compute the similarity measure for DTW and the subsequent discriminative classification. The purpose of these regions is to exclude low-energy regions which are susceptible to background noise. A large improvement in classification accuracy has been observed at low SNR conditions when the prominent region is included [3]. The frame weighting function (which sums to 1) assigns more weights to reliable frames based on short-time correlation. In addition to obtaining the template, SFA also returns the aligned spectrograms that are used to derive the template. We will use these aligned spectrograms to train an SVM classifier in a later stage.

2.3 DTW-based segmentation

Our algorithm uses the derived template (Section 2.2) as a sliding elastic window to detect a target pattern from a continuous recording. This algorithm is a modified version from the DTW procedure used in the SFA. Here, DTW is used to find the locally optimal time warping function between a segment of test spectrogram S and a reference template $M^{(k)}$.

DTW-based segmentation (input: $M^{(k)}$, $R^{(k)}$, $w^{(k)}$, S)

- The superscript $\square^{(k)}$ indicates that the $\square$ is a specific attribute of phrase class k
- i and j are the time indices of the reference $M^{(k)}$ (with N_M frames) and test spectrogram S (with N_S frames), respectively.
- C(i,j) is the cosine similarity between the i^{th} frame of $M^{(k)}$ and the j^{th} frame of S.
- P(i,j) is the intermediate cumulative score.
- The operator $\odot$ denotes the element-wise multiplication.

1) $C(i,j) = \theta(M^{(k)}{}_i \odot R^{(k)}{}_i, X_j \odot R^{(k)}{}_i)$

2) $P(1,j) = C(1,j)$

$P(2,j) = \max\{P(1,j)+w_2C(2,j), P(1,j-1) + w_2C(2,j)\}$ for j>1

Recursive step

$$P(i,j) = \max \begin{cases} P(i-1,j-2)+0.5w_iC(i,j-1)+0.5w_iC(i,j) & \textbf{Path 1} \\ P(i-1,j-1) + w_iC(i,j) & \textbf{Path 2} \\ P(i-2,j-1) + w_{i-1}C(i-1,j) + w_iC(i,j) & \textbf{Path 3} \end{cases}$$

3) $[p_1\ p_2\ p_3\ \dots\ ]$ = peak locations of function $P(N_M,j)$ whose values are greater than 0.75 and are mutually separated by at least N_M.

4) For each p_i, backtrack the optimal path and assign each template frame with a frame of S for Path 2 and Path 3. For path 1, the vector assigned to the template frame is obtained by averaging the spectra of frame j and j-1.

Note that: 1) The cosine similarity is not computed over the entire frequency range, but only on the range determined by the prominent region indicated by the reference frame $R^{(k)}{}_i$. 2) In computing the cumulative score, the contribution of each reference frame is weighted differently based on the frame weighting function *w*. 3) The cumulative score at the last reference frame measures the overall similarity of the optimal path at that point; the range of similarity is from 0 to 1. Instead of backtracking from each frame in the test spectrogram, backtracking is performed from frames that correspond to peaks in $P(N_M,j)$ only frames with peak values higher than 0.75, and with a peak separation of at least the frame number of the template are selected 4) The optimal paths are backtracked, starting from those selected peaks. Each reference frame *i* will match with one or two frames from the test spectrogram. In case of two frames (Path 1), frame *i* will be matched with the average of the spectra of frame *j* and *j-1*. As a result, each candidate segment will have the same number of frames as template $M^{(k)}$.

2.4 SVM training

The verification task uses SVM as a discriminative classifier to reject outlier candidates. We selected an SVM because it is effective for limited training data (4 samples) and the model can be efficiently represented [2]. To train an SVM classifier, the positive (in-class) class comprises the aligned spectrogram segments from the SFA and valid segment candidates from the DTW-based segmentation; the negative class consists of solely invalid candidates (Fig. 1). A segment is considered valid if it satisfies the Relative Segmentation Error Constraint (RSEC) criteria which is explained in Section 3.3. Features for each

training instance are obtained by concatenating the spectrogram values within the prominent region. The resulting SVM classifier and the corresponding template are used for verification and DTW-based segmentation, respectively, in the testing procedure.

2.5 Testing Procedure

To search a given phrase class from a continuous recording, the spectrogram is first computed. The resulting spectrogram is passed through the DTW-based segmentation to obtain segment candidates, which align to the template of the given phrase class. For each segment, spectrogram values within the prominent region are vectorized to obtain a feature vector in the same way as the training procedure. Each feature vector is then classified using the SVM model derived in the training procedure to determine whether the segment is a target phrase or an outlier (i.e., another phrase class or noise).

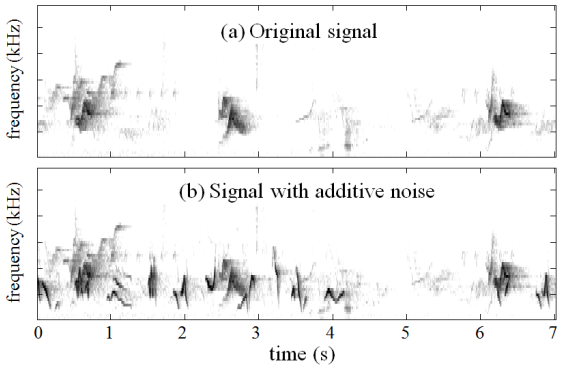

Fig. 2 Spectrograms without and with noise added

3. EXPERIMENTAL SETUP

3.1 Database

All experiments in this paper used recordings described in [2]. Song fragments (phrases) for classification were obtained from recordings of Cassin's Vireo (*Vireo cassinii*) in 2010. The database has 13 recordings which we split into two sets for cross validation. Each experiment randomly selected 6 files for training and the remaining 7 files were used for testing. Four rounds of experiments were repeated and the results were averaged. Phrase classes with at least 4 occurrences found in training recordings were selected from the classes on the training set. Depending on the random partition of training and testing sets in each experiment, the number of classes ranges from 31 to 35; the number of training sample per classes range from 4 to 50; the total number of target phrases are 1771. The recordings and annotations for this study are available online at http://taylor0.biology.ucla.edu/al/bioacoustics/.

To evaluate noise-robustness, we simulated noisy birdsongs by adding background noise at various signals-to-noise ratio (10, 5, 0, -5, and -10 dB). The background noise was recorded in the same environment, when the target bird species is not singing. There are total of 7 noise files (20 minutes long). These files contain birdsongs from other species as well as ambient noise. For a given recording, the noise file ID and time location were selected randomly to match the length of the recording. The noise portion is scaled to generate a pseudo signal-to-noise ratio of a given SNR. An example of the additive noise effect is shown in Fig 2. Note that this SNR value represents the upper bound of the true SNR because the original files are not completely noise-free. These generated noisy signals are used only in testing; the training uses only original recordings (fairly clean signals).

3.2 Comparison Algorithms

Two main contributions of our phrase detection system include noise-robust segmentation as well as the integration of the template-based segmentation and a discriminative classifier for outlier rejection. The optimal classifiers can be explored in a subsequent study; this paper focuses on the first contribution because accurate segmentation is crucial for most automatic phrase detection or transcription algorithms. For this reason, we compared our algorithm with baseline systems that employ an energy-based or entropy-based segmentation [8]-[9]. Since the recordings used in this experiments are generally long and the threshold of the energy-based method depends on the global maximum energy in the recording, the segmented outputs are mostly inaccurate [16]. We saw improvements when a long recording is divided into subintervals as inputs to the algorithm so we used this modification throughout this experiment. For the entropy-based segmentation, we kept all the default parameters except the frequency range to match the database [9]-[10]. For comparison algorithms, the DTW-based segmentation in Fig 1(a) and 1(b) is replaced with two sequential operations; segmentation from the algorithm followed by DTW that are used in SFA. In other words, we applied DTW to the segment candidates; using the template derived from manual annotation, such that the time dimension of the DTW-aligned spectrograms of all segment candidates is equal to that of the template. We also applied the prominent region to those comparative algorithms since it gives a significant improvement in noise. The accuracy of the segment boundaries are evaluated by comparing them with manual annotations. Using a linear kernel with high-dimensional features of limited training data gave the best performance for all algorithms. We applied a power of 0.1 to feature values for dynamic scale compression. The detection threshold on the confidence score of the SVM is varied to obtain the Equal Error Rate (EER).

3.2 Evaluation Framework

To evaluate segmentation algorithms, several criteria have been proposed; for example, a segmentation is considered correct if the detected segment overlaps with a manually annotated phrase interval [9]. However, this criterion can generate unreliable labels; for instance, a detected segment that has only one overlapping frame will be considered valid. For a more meaningful evaluation, we consider a segmentation valid if the boundary offsets relative to the manual annotation is less than a threshold, which we call Segmentation Error Constraint (RSEC). Under RSEC = ε, a segment is considered valid if and only if $\frac{|y_{start}-x_{start}|}{y_{stop}-y_{start}} \leq \varepsilon$ and $\frac{|y_{stop}-x_{stop}|}{y_{stop}-y_{start}} \leq \varepsilon$, where y_{start} and y_{stop} are the start and stop times of the manually-annotated segment, respectively; x_{start} and x_{stop} are the start and stop times of the segmentation algorithm, respectively. For example, with ε = 0.5, a given segment is considered correct if the boundary errors of both ends are less than or equal to half the phrase duration of the manual segmentation. If ε = 0, the boundary errors of both ends are zeros, i.e. this segmentation is identical with the manual annotations; however, this case is difficult to achieve in practice. We present the performance curve with varying RSEC to determine the robustness to RSEC of each algorithm. The Equal Error Rate is used to measure the performance of the segmentation algorithms.

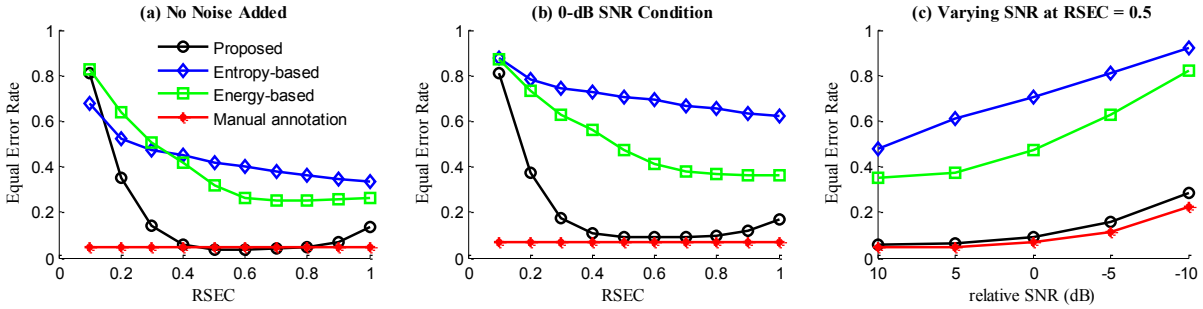

Fig. 3 The Equal Error Rate of each algorithm (all plots have the same legends). Figs. 3(a) and 3(b) show the performance of each algorithm with different Relative Segmentation Error Constraint (RSEC) criteria under clean and 0-dB SNR conditions, respectively. Fig. 3(c) show the performance trend with different SNR conditions at RSEC of 0.5 (ε = 0.5).

4. RESULTS AND DISCUSSION

Fig 3a shows the EER of each algorithm evaluated with RSEC ranging from 0.1 to 1. The performance of perfect boundary segments is also shown for lower bound. The EER curve of the perfect segmentation is flat because its relative boundary error is always 0 hence the EER does not change with different RSEC criteria. It is constant at 4.45% because there are false alarms and misses from the SVM classification; note that the results represent the final output of the entire system. The EER of all algorithms suffer from segmentation errors at low RSEC; it is difficult to achieve perfect segment boundaries because the energy at boundary frames varies. Different stopping threshold of each algorithm will generate different boundaries, hence a small deviation is acceptable.

When a higher RSEC is allowed, the EER of each algorithm decreases up until a certain point where it starts to increase again because of some unreliable labels are generated by the high RSEC criteria. For instance, if the starting segmentation boundary from an algorithm is delayed by 80%, there will be at most 20% overlap with the manual annotation. If we allow RSEC of 0.9 (ε =0.9), this segment will be considered a positive segment but only 20% of common portion will not sufficiently represent the given phrase. Note that the EER curve of our algorithm almost reaches the EER from manual annotation at RSEC between 0.5 and 0.8. This does not mean that the algorithm is perfect; it only tells us that under this RSEC criteria, the EER are about the same. In 0 dB condition, the performance of our algorithm degrades less than other algorithms, compared to the clean condition.

The EER of each algorithm increases as the relative SNR decreases and at a significant rate after 0 dB SNR (Fig 3c). At SNRs higher than 5 dB, the EER of the proposed algorithm is close to that when manual segmentation is used. At 0 SNR dB condition, where energy of target birdsong and background noise are equal, the proposed algorithm still yields low ERR of 0.1, validating its noise-robustness.

Notice that some EER points are higher than 0.5, which is the upper bound of binary detection (pure guess). This is possible because this task of joint segmentation and verification is more challenging than a binary detection. In order to be counted as a correct prediction, two criteria must be satisfied, the deviation boundaries of both ends must be less than RSEC and the SVM must classify the segment correctly. The entropy-based algorithm does not perform well because the background noise contains high-entropy signals such as vocalizations of other bird species. The energy-based segmentation algorithm generally performs better than the entropy-based segmentation but it also suffers in noisy conditions. This is expected because background noise is also of high energy, especially in low SNR conditions. Most errors of both algorithms occur when the algorithms merge target phrase with a neighboring background noise.

The proposed algorithm, on the other hand, cuts off the segment at each end of the template reference. This segmentation approach is similarity-based in the sense that it aligns with the template reference according to its frame-wise similarity. The prominent regions and frame weighting function additionally facilitate the alignment procedure to focus on the relevant regions while discarding noise. However, when noise level is high, the proposed algorithm degrades as well because noise is also present in the prominent regions. In the context of segmentation, the template-based segmentation has a limitation that it requires a template of that particular phrase class. If there is no prior class label, our algorithm cannot perform SFA to derive the template for segmentation. The entropy-based and energy-based segmentation, on the other hand, do not require any template so they are suitable for analyzing data that has not been annotated. However, for a phrase detection where the detection of particular phrases is of interest, the proposed template-based algorithm has been shown to be robust to segmentation error and background noise.

5. CONCLUSIONS

We proposed a segmentation and verification algorithm for bird phrase detection. The algorithm performs template-based segmentation using DTW to account for duration variability. It uses the prominent regions to account for background noise. Segment candidates are then passed to a discriminative classifier (in this case, SVM) to reject outliers. On a Cassin Vireo's database, the proposed algorithm obtained the lowest EER in most cases compared to energy-based and entropy-based segmentation algorithms. In fairly clean recordings, our proposed algorithm achieves an EER of approximately 5.71 % with relatively accurate segmentation boundaries. Using noisy recordings with low SNRs, the performances of comparative algorithms degrade dramatically compared to the proposed algorithm which keeps the performance trend close to the lower bound. The advantages and limitations of our algorithm are also discussed. Our future work will include an investigation of other discriminative classifiers and extending this work to automatic transcription.

6. ACKNOWLEDGEMENTS

This study was supported in part by NSF Award No.0410438 and IIS-1125423. We thank George Kossan for his assistance with phrase identification and Rafael Torres for modifying of the Harma Syllable Segmentation MATLAB code.

7. REFERENCES

[1] T. Scott Brandes, “Automated sound recording and analysis techniques for bird surveys and conservation,” Bird Conservation International, vol. 18 , pp. S163–S173, 2008.

[2] 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," Interspeech, 2012.

[3] Kantapon Kaewtip, Lee Ngee Tan, Abeer Alwan, Charles E.Taylor, "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification", ICASSP 2013, pp. 768-772.

[4] Seppo Fagerlund, "Bird species recognition using support vector machines," EURASIP Journal on Advances in Signal Processing, 2007, 2007.

[5] Miguel A. Acevedoa, Carlos J. Corrada-Bravoc, Héctor Corrada-Bravob, Luis J. Villanueva-Riverad, and T. Mitchell Aidea, “Automated classification of bird and amphibian calls using machine learning: A comparison of methods,” Ecological Informatics, Vol. 4, pp. 206–214, 2009

[6] Forrest Briggs, Fern Xiaoli, and Raich Raviv. "Technical Report (Not Peer Reviewed): Acoustic Classification of Bird Species from Syllables: an Empirical Study."

[7] Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Z. Fern, and Raviv Raich, Sarah J. K. Hadley, Adam S. Hadley, and Matthew G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," Acoustical Society of America Journal 131 (2012): 4640.

[8] Harma, Aki. "Automatic identification of bird species based on sinusoidal modeling of syllables." In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, vol. 5, pp. V-545. IEEE, 2003.

[9] Wang, Ni-Chun, Ralph E. Hudson, Lee Ngee Tan, Charles E. Taylor, Abeer Alwan, and Rung Yao. "Change point detection methodology used for segmenting bird songs." In Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on, pp. 206-209. IEEE, 2013.

[10] Wang, Ni-Chun, Ralph E. Hudson, Lee Ngee Tan, Charles E. Taylor, Abeer Alwan, and Kung Yao. "Bird phrase segmentation by entropy-driven change point detection." In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 773-777. IEEE, 2013.

[11] Vlad M. Trifa, Alexander N. G. Kirschel, Charles E. Taylor, and Edgar E. Vallejo, “Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models,” The Journal of the Acoustical Society of America (JASA), vol. 123, 2424–2431, 2008.

[12] Joseph A. Kogan and Daniel Margoliash , “Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study,” Journal of the Acoustical Society of America (JASA), vol. 103, pp. 2185–2196, 1998.

[13] Wei Chu and Daniel T. Blumstein, “Noise robust bird song detection using syllable pattern-based hidden Markov models,” IEEE ICASSP, pp. 345–348, 2011.

[14] Ken Ito, Koich Mori, and Shin-ichi Iwasaki, ‘‘Application of dynamic programming matching to classification of budgerigar contact calls,’’ Journal of the Acoustical Society of America (JASA), vol. 100, 3947–3956, 1996.

[15] Anderson, Sven E., Amish S. Dave, and Daniel Margoliash. "Template-based automatic recognition of birdsong syllables from continuous recordings." The Journal of the Acoustical Society of America 100, no. 2 (1996): 1209-1219.

[16] Michael Lidenuth, "Harma Syllable Segmentation" http://www.mathworks.com/matlabcentral/fileexchange/29261-harma-syllable segmentation/content/harmaSyllableSeg/harmaSyllableSeg.m.