ACOUSTIC FRONT-END OPTIMIZATION FOR BIRD SPECIES RECOGNITION

Martin Graciarena, *Michelle Delplanche*, *Elizabeth Shriberg*, *Andreas Stolcke, Luciana Ferrer*

SRI International, Menlo Park, CA, USA
martin@speech.sri.com

ABSTRACT

The goal of this work was to explore the optimization of the feature extraction module (front-end) parameters to improve bird species recognition. We explored optimizing the spectral and temporal parameters of a Mel cepstrum feature-based front-end, starting from common parameter values used in speech processing experiments. These features were modeled using a Gaussian mixture model (GMM) system. We found an important improvement when increasing the spectral bandwidth and increasing the number of filter banks. We found no improvement when switching the filter bank distribution from the perceptually based Mel frequency scale to a linear frequency scale. In addition, no improvement was found when we either reduced or increased the time resolution. On the other hand, we found that the best time resolution is species dependent. We did find great improvements from a species-specific combination of different front-ends with different time resolutions relative to using the same front-end time resolution for all species.

Index Terms— Bird species recognition, acoustic front-end, Gaussian mixture model.

1. INTRODUCTION

The goal of this work is bird species recognition based only on bird song samples. Authors like Härmä [1] have approached the problem of bird species identification by using a specific model of bird song syllables. In [2], Härmä and Somervuo extended this work by using the harmonic structure. In [3], Somervuo and Härmä employed a song-level modeling approach using syllable pair histograms. Recently, Somervuo *et al.* [4] compared three feature representations of bird sounds for automatic species recognition. In [5], Kwan *et al.* explored Gaussian mixture models (GMMs) and hidden Markov models (HMMs) of acoustic features with the goal of bird species classification to aid in the minimization of bird strikes to airplanes in airports. In the publications cited here the feature extraction parameters are set to a certain set of values. It is not clear if those values are optimal for bird species classification.

The goal of this work is to optimize the feature extraction module (front-end) parameters to better model the signal characteristics of bird vocalizations. We will start from a commonly used set of front-end parameters such as the one used in speech recognition and speaker identification experiments, and we will optimize frequency and temporal parameters. We will also explore if the optimal parameters of the front-end are bird species-specific. The motivation is that different species have different anatomies for song production, and therefore may produce songs with different signal characteristics.

2. BIRD SONG DATA

In our experiments we used the following publicly available collections of bird song data:

- Macaulay Library of Natural Sounds, *Bird Songs of California*, Cornell Laboratory of Ornithology, Geoffrey A. Keller, 3-CD, 2003
- Peterson Field Guides: Bird Songs: *Western North America*, *A Field Guide to Western Bird Songs*, Second Ed., Cornell Laboratory of Ornithology Interactive Audio, 1992
- Peterson Field Guides: Bird Songs: *Eastern and Central North America, A Field Guide to Bird Songs*, Third Ed., Cornell Laboratory of Ornithology Interactive Audio, 1990
- *Common Bird Songs* (Audio CD), by Donald J. Borror, Dover Publications, 2003
- *Common Birds and Their Songs* (Book and Audio CD), by Lang Elliott and Marie Read, Houghton Mifflin, 1998
- Stokes Field Guide to Bird Songs: *Western Region* (Audio CD), by Kevin Colver *et al.,* Hachette Audio, 1999

These collections contain bird song vocalizations from multiple species and were captured using different types of recording equipment.

We extracted the bird vocalization segments in the CD waveforms from background signals and discarded very short calls. We used a simple voice activity detection system, with acoustic models trained with bird vocalization data.

3. FRONT-END ANATOMY

The following steps represent the processing in a standard Mel frequency cepstral coefficient (MFCC) front-end used in speech applications [6]:

1. Analog to digital conversion
2. Continuous signal split into overlapping segments (frames)
3. Spectrum computation for each frame
4. Multiplication by filter bank modules
5. Log computation of average energy in each filter bank
6. Inverse discrete cosine transform (DCT) that results in the cepstrum coefficients

The full feature set used in experiments is an extended vector comprising the frame-based MFCC features appended by delta and delta-delta features. These delta and delta-delta features are computed as linear combinations of features from neighboring frames. They approximate the time variation of the frame-based features.

A set of front-end parameters commonly used in speech applications, corresponding to a 16 kHz sampling rate, is shown in Table 1.

Table 1 – Common Speech Applications
MFCC Front-end Parameters

Parameter	Value
Sampling rate	16 kHz
Frequency range	100 Hz – 6.4 kHz
Number of filter banks	24
Filter bank scale	Mel
Frame length	25 ms
Frame separation	10 ms
Number of cepstral features	13
Full feature vector dimension	39

We start optimizing the acoustic front-end with the goal of better modeling the bird song signal characteristics. Comparing speech and bird song signals, we observe that some bird song signals have shorter vocalization durations. In addition, the spectral information is more localized in frequency.

4. SPECIES VERIFICATION SYSTEM

A GMM system [7] was used to model MFCC features computed on bird vocation waveforms. The system is based on the GMM-UBM model paradigm, in which a bird species model is adapted from a universal background model (UBM). Maximum a posteriori (MAP) adaptation was used to derive a bird species model from the UBM. The GMM has 1024 Gaussian components. The front-end uses utterance-level mean and variance normalization [8].

We used a speaker verification paradigm in our experiments. For a given species model we used two types of testing data – one from other samples of the same species (called *true trials*) and the other using samples from other species (called *impostor trials*). In this paradigm the goal is to make a decision on whether to accept or reject the trial samples as being from the same species as the one in the training model. If an impostor trial is accepted, it is called a *false acceptance error*. If a true trial is rejected, it is called a *false reject error*. The equal error rate (EER) is the point at which the percent of false acceptance errors and of false reject errors are equal. Typically, the number of impostor trials is one or two orders of magnitude larger than the number of true trials.

From the bird song datasets we identified ninety-two species with four or more vocalizations. Some of the species used in this study are shown in Table 2. We used samples from these ninety-two species for model training and for true trial testing. We split the rest of the species into two sets, one to be used as a portion of the impostors and the other to train our UBM model. For a given training species we used also the data from the other ninety one species as impostors. Next we split the impostor list into two parts, one for development and the other for testing. This split was made three times using different random seeds. The goal of these different splits is to increase the generalization of the results. Therefore, for each species we had three different groups of impostors, with each impostor group divided into development and test parts. Since the number of target trials per species was small, in the experiments we split randomly the target scores in two sets of the same size, one was used in development and other was used in testing. We also did this target score split three times using different random seeds. The average number of target trials per species was 9 in both dev and test sets. The average number of impostor trials per species was around 1 K for the development set and around 3 K for the test set. The total number of true trials was 1718 for both the dev and test sets and the total number of impostor trials was 100 K for the dev set and 280 K for the test set.

Table 2 – Some Bird Species

alder flycatcher	american bittern	american goldfinch
baltimore oriole	bewicks wren	brown-headed cowbird
bullocks oriole	brewers blackbird	carolina wren
common yellowthroat	dark-eyed junco	eastern bluebird
hooded warbler	evening grosbeak	fox sparrow
indigo bunting	kentucky warbler	great blue heron
lincolns sparrow	northern cardinal	house finch
nashville warbler	pine grosbeak	lark sparrow
parula warbler	winter wren	willow flycatcher
song sparrow	savannah sparrow	pine siskin

5. FRONT-END OPTIMIZATION

In optimizing the front-end parameters with the goal of reducing the bird species recognition error, we start from common MFCC speech front-end parameters. Tables below show the average EER on the development dataset and the test dataset over all species. Whenever a decision must be made regarding comparison of systems, we decide based on the development dataset.

5.1. Bandwidth Optimization

Bird vocalization waveforms were sampled at 44 kHz, which corresponds to an available bandwidth of 22 kHz. Therefore, we explore whether increasing the upper bandwidth frequency will reduce the error rate. In Table 3 we optimize the bandwidth increasing over the common speech 6.4 kHz upper limit.

In Table 3 we show the average EER by species results from increasing the bandwidth to 10 kHz, 13 kHz and 16 kHz. We also explored changing the number of filter banks in each frequency range condition.

Table 3 – Optimizing Frequency Range
and Number of Filter Banks

Frequency Range	Number of Filter Banks	Avg % EER by Species	
		Dev	Test
100 - 6.4 kHz	24	11.00	12.24
100 - 10 kHz	24	10.22	11.02
100 - 10 kHz	32	9.92	11.04
100 - 13 kHz	24	10.95	11.45
100 - 13 kHz	32	9.34	10.60
100 - 13 kHz	41	**8.98**	**10.22**
100 - 13 kHz	52	9.36	10.53
100 - 16 kHz	52	10.45	10.97

Table 3 shows a big EER reduction in both the dev and the test datasets with increased bandwidth. The optimal bird song upper frequency appears to be 13 kHz. Increasing over this limit does not further produce EER reductions. Therefore, we do not need the full available bandwidth of 22 kHz to obtain the best EER performance for bird signals.

We found the best number of filter banks to be 41. If we divide the spectral range by the number of filter banks, we obtain a measure of spectral resolution per filter bank. The best spectral resolution for bird songs is smaller than the spectral resolution of a common 16 kHz speech front-end. This is reasonable if bird song spectral patterns are simpler than speech spectral patterns.

We also found in Table 3 and other experiments not shown here, that reducing the number of filter banks increases the EER. Therefore it seems that the filter bank processing seems advantageous for bird song signals over not using it.

We also tried increasing the lower limit of the frequency range but it did not lead to EER reductions.

5.2. Frequency Scale Optimization

All the experiments so far were done using the Mel perceptual filter bank scale. This frequency distribution is linear up to 1 kHz and follows a logarithmic scale above 1 kHz [6]. This scale is based on properties of human hearing. It may not necessarily be the most advantageous for bird sounds, and therefore alternative frequency distributions are worth exploring.

In Table 4 we first show the EER results of the best configuration in Table 3 using the Mel frequency scale. Next we show the EER result using the same bandwidth and same number of filter banks but now distributed on a linear scale.

Table 4 – Frequency Scale Optimization

Frequency Range	Number of Filter Banks	Scale Type	Avg % EER by Species	
			Dev	Test
100 - 13 kHz	41	Mel	**8.98**	**10.22**
100 - 13 kHz	41	Linear	12.82	13.26

We found from the results in Table 4 that using a linear filter bank scale does not improve performance over using the Mel scale. Therefore, most of the spectral energy seems to be allocated where most of the Mel resolution is concentrated.

5.3. Frame Length Optimization

Bird song signals, in some cases, present faster evolutions compared to the speech signal. Therefore, it would be advantageous to use smaller frame lengths. However, it is not clear if this smaller frame length would be beneficial for all species. Therefore, we explored frame lengths longer and shorter than the 25 ms length commonly used in speech processing applications. In Table 5 we present the EER results on the development and test datasets for different frame lengths from 40 ms to 10 ms. Since the frame advance used is 10 ms, we cannot reduce the frame length below 10 ms without starting to lose signal samples, assuming we fix the sample rate at 10 ms.

We found from Table 5 that the best frame length in both the development and the test datasets is 25 ms, which matches the speech frame length. The test EER remains fairly uniform across a wide range of frame lengths.

One reason that we may not obtain a gain when reducing the frame length may be that for some species we are reducing the EER whereas for others we are increasing the EER, resulting in a cancellation of the benefits. To illustrate this trend we present in Table 6 the average test EER by species for three different frame lengths for some of the bird species shown in Table 2.

From Table 6 we conclude that for some species (pipit, finch, warbler, etc) it is better to use shorter frame lengths while for others (flycatcher, swallow, grosbeak, etc) it is better to use longer frame lengths. For some species (goldfinch, lincolns sparrow, nashville warbler, etc) using 25 ms achieves the lowest EER. Some species like the lincolns sparrow and the nashville warbler have lower than expected EER for the 25 ms frame length, compared to the two other frame lengths. This is possibly an effect of computing the EER with a small number of true samples per species.

Table 5 – Optimizing the Frame Length

Frame Length	Avg % EER by Species	
	Dev	Test
40 ms	10.78	10.95
35 ms	9.90	11.70
30 ms	10.93	11.49
25 ms	**8.98**	**10.22**
20 ms	9.41	10.61
15 ms	10.06	10.65
10 ms	9.54	10.75

Table 6 – Average Test EERs by Species for Three Frame Lengths

Species	EER for Frame Length		
	15 ms	25 ms	35 ms
alder flycatcher	1.56	0.58	**0.53**
american goldfinch	20.46	**19.50**	22.37
american pipit	**15.95**	27.33	32.77
barn swallow	2.05	1.57	**1.29**
carolina wren	**22.22**	**22.22**	38.18
common yellowthroat	10.49	**10.48**	11.27
eastern bluebird	**2.06**	3.19	2.20
fox sparrow	17.82	16.48	**15.84**
house finch	**7.30**	9.45	9.46
kentucky warbler	**0.48**	1.08	13.95
lark sparrow	15.41	**15.30**	16.67
lincolns sparrow	12.16	**2.67**	10.66
nashville warbler	9.25	**1.35**	11.24
northern cardinal	15.55	16.67	**15.53**
parula warbler	1.40	1.17	**0.36**
pine grosbeak	11.62	11.84	**9.80**
song sparrow	**15.34**	21.76	24.44
willow flycatcher	**0.17**	0.21	0.23

6. SPECIES-SPECIFIC FRONT-ENDS

Based on the results from Table 6, we explored the use of species-specific time resolutions and whether we can predict the best frame lengths for a given species in the test dataset using that species' development dataset. We used the EER for each time resolution in the development dataset as a performance indicator to guide the selection of the best-performing time resolutions for a given species.

Selection by EER threshold: We computed the EER for a given species for a range of frame lengths from 10 ms to 40ms in the development dataset. We selected the frame lengths for which the EER in the development dataset fell below a certain EER threshold. This EER threshold was computed as a percent of the difference between the maximum and the minimum EERs in the dev dataset. We then averaged the test scores of the selected frame lengths and computed the EER. This score averaging was done on each development and test datasets. We explored multiple thresholds ranging from one percent, five percent, and so on to using all the front-ends, which corresponds to using one hundred percent.

We first computed the oracle EER result which would show the best possible performance. Specifically, for each species we computed the lowest EER of all the frame lengths in the test dataset. Next we averaged this EER over all species. This corresponds to the oracle result of knowing the best time resolution for each species. If this result was not much different from the average EER, then there would be little hope that selecting the best front-end in the development dataset will reduce the test dataset EER.

Table 7 – Comparing Species-Independent (SI) and Score Combination of Species-Dependent (SD) Front-ends

Front End	System		Avg % EER by Species	
			Dev	Test
SI	Frame Length 25 ms		8.98	10.22
SD	Lowest EER on Test (Oracle)		7.80	5.75
SD	Frame Length Selection by EER Threshold: Average Test Scores from Dev Frame Lengths with EER Lower than Threshold	100%	8.86	10.01
		10%	4.73	8.38
		5%	4.66	**8.01**
		1%	**4.64**	8.05

In Table 7 we present the average EER by species for the development and test datasets. We show in the first column whether the front-end is species independent (SI) or species dependent (SD). The first row in Table 7 shows the best result from Table 5 corresponding to a fixed frame length of 25 ms. Next is the oracle result from selecting the best species-specific frame lengths in the test dataset. Since this result is much better than the baseline 25 ms result, there seems to be hope that using species-specific frame lengths we could reduce the EER. However this big EER reduction can be misleading, as random variations in the test results for multiple resolutions may produce rather optimistic results. Finally, we present the results from predicting the best frame lengths on the development dataset. These results are obtained by averaging the test scores based on selecting the front-end frame lengths with development dataset EERs lower than a certain threshold. We observe that averaging all the front-ends (threshold at 100 percent) produces a small reduction in the EER compared to the SI front-end. This may be due to reducing the variability of score outliers. This EER should be used as reference for comparison with other EER results from this set of experiments. Next we reduce the threshold to select only frame lengths with lower EERs on the development dataset. We observe that not only the dev EER is reduced, as expected since we are optimizing this same dev database, but also the test EER is reduced. We found a great reduction on the test EER at thresholds of 5 percent and 1 percent. The fact that the test EER result from the 5 percent threshold is better than the one from the 1 percent one means that selecting several top performing frame lengths is more robust than just relying on the single best one in the development dataset.

When we compare the result from score averaging and the baseline performance we find that, on average, 57 percent of the bird species have lower EER and 27 percent have higher EER. The rest of the species had no change in EER after the frame length optimization. These species had in general easily distinguishable songs and a very low EER, which stayed constant over all sets of frame lengths.

7. CONCLUSIONS

We explored the optimization of front-end parameters in bird species recognition experiments starting from front-end parameters commonly used in speech applications. We explored the optimization of spectral and temporal parameters of a Mel cepstrum feature front-end. improvement was found from increasing the bandwidth and increasing the number of filter banks. However, no improvement was obtained when we optimized the time resolution. We showed that the best time resolutions are species-specific. Next we found a big improvement from species-specific combination of different front-ends with different time resolutions. These species-specific experiments were done using very few true samples per species. Therefore it would be advisable to validate these results using more samples per species.

In further experiments, other parameters could be optimized, such as the number of cepstral features, delta differences and frame separation. Whether these parameters are also species-specific is something to be explored.

8. ACKNOWLEDGMENTS

We thank Sachin Kajarekar for the fruitful discussions. The views are those of the authors and do not represent the views of the funding agency. This work was supported by Sandia National Laboratories.

9. REFERENCES

[1] A. Härmä, "Automatic Identification of Bird Species Based on Sinusoidal Modeling of Syllables," in *Proc. of ICASSP*, Hong Kong, 2003.

[2] A. Härmä and P. Somervuo, "Classification of the Harmonic Structure in Bird Vocalization," in *Proc. of ICASSP*, Montreal, Canada, 2004.

[3] P. Somervuo and A. Härmä, "Bird Song Recognition Based on Syllable Pair Histograms," in *Proc. of ICASSP*, Canada, 2004.

[4] P. Somervuo, A. Härmä, and S. Fagerlund, "Parametric Representations of Bird Sounds for Automatic Species Recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 6, pp. 2252–2263, Nov. 2006.

[5] C. Kwan, G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, and K.C. Ho, "Bird Classification Algorithms: Theory and Experimental Results," in *Proc. of ICASSP*, Montreal, Canada, 2004

[6] X. Huang, A. Acero, and H-W Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall, 2001.

[7] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker Verification Using Adapted Mixture Models," *Digital Signal Processing*, vol. 10, pp.181-202, 2000.

[8] S. Kajarekar, N. Scheffer, M. Graciarena, E. Shriberg, A. Stolcke, L. Ferrer, and T. Bocklet, "The SRI NIST 2008 Speaker Recognition Evaluation System," in *Proc. ICASSP*, Taipei, Taiwan, 2009.