

A Segmentation Algorithm for Zebra Finch Song at the Note Level

Ping Du and Todd W. Troyer

Neuroscience and Cognitive Science Program, Dept. of Psychology

University of Maryland, College Park, MD 20742

Abstract

Songbirds have been widely used as a model for studying neuronal circuits that relate to vocal learning and production. An important component of this research relies on quantitative methods for characterizing song acoustics. Song in zebra finches - the most commonly studied songbird species - consists of a sequence of notes, defined as acoustically distinct segments in the song spectrogram. Here we present an algorithm that exploits the correspondence between note boundaries and rapid changes in overall sound energy to perform an initial automated segmentation of song. The algorithm uses linear fits to short segments of the amplitude envelope to detect sudden changes in song signal amplitude. A variable detection threshold based on average power greatly improves the performance of the algorithm. Automated boundaries and those picked by human observers agree to within 8 msec for >83% of boundaries.

Key words: *Zebra Finch, Song Analysis, Song Segmentation, Peak Detection*

Introduction

Songbirds have been widely used as a model for studying neuronal circuits that relate to vocal learning and production. With the exponential growth in data storage capability it is now feasible to study of song development using a data-intensive approach, recording all vocalizations sung by juvenile birds during the initial period of vocal practice. While the first studies using this approach have been published (e.g. [2,8]), more detailed analysis algorithms are required to fully characterize the range of birdsong behavior. The size of the data sets involved put a premium on computational resources.

This paper presents an efficient algorithm for song segmentation suitable for high throughput analysis. The algorithm is optimized for the song of the zebra finch, the most commonly studied songbird species. A song usually starts with a few introductory notes, followed by several repeats of a single motif, defined as a stereotyped sequence of syllables. Syllables are defined as acoustic productions separated by gaps of silence, and correspond to the pattern of respiration during singing [3]. Syllables have been proposed to be the basic unit of song production [1], and the silence between syllables makes them easy to segment automatically. For these reasons, most quantitative approaches to song analysis have used syllables as the basic unit of analysis (e.g. [4,5,6,7]).

However, several factors suggest that song analysis might be best accomplished at the level of notes, defined as segments of song separated by rapid transitions in the spectrogram. Behaviorally, Cynx [1] reported that although birds that are startled by a strobe flash most commonly stop at syllable boundaries, some birds did stop in the middle of syllables and in these cases the song was interrupted at note boundaries. Moreover, by definition notes are periods of song in which the

spectro-temporal features of song are relatively constant, suggesting that this is the appropriate level at which to base automated song analysis routines.

The starting point for the analysis is the observation that note boundaries most often correspond to sudden changes in the amplitude envelope of the song. (Fig. 1). Therefore, as a first attempt at song segmentation, we developed an algorithm to detect these sudden changes in amplitude, and compared the performance of this algorithm to human segmentation.

Methods

Songs were recorded from male zebra finches housed in small soundproof chambers and digitized at 24.414 kHz and scaled to fit on an arbitrary amplitude scale ranging from -1 to 1 . The signal was high pass filtered above 500 Hz and the amplitude envelope of the song signal was obtained by calculating root mean square (RMS) amplitude using a 128 point (~ 5.2 ms) sliding square window. Windows are overlapped by 50%, yielding a resolution of 2.6 ms. This amplitude envelope forms the starting point for our algorithm.

For each point on the amplitude envelope curve, several points immediately preceding and following that point are chosen (see below). On each side a least squares fit is performed with a straight line anchored at the center point (Fig. 1C). The angle between these two lines gives a measurement of sudden changes in the shape of the amplitude envelope. The angle is measured from the right line to the left line clockwise, thus ranges from 0 to 2π . If the angle is above π , as shown in the second example in Fig. 1C, it indicates a sudden increase in the slope of the amplitude envelope at that time point.

The selection of points for the linear fit and the angle calculation require that time and amplitude be expressed using a comparable metric. To accomplish this, both time and amplitude were expressed as normalized (unitless) quantities. Time was expressed in units of resolution (2.6 msec) and amplitude was expressed as a fraction of the average amplitude over the entire song. Thus a line that went from the zero amplitude at one time bin to the average amplitude of the song in the next time bin would have a slope of 45 degrees. The window for choosing the number of points for the linear fit was determined as a fixed length along the curve, determined as the summed Euclidean distance between adjacent points using normalized coordinates. If the curve is flat at the point of interest, most of the distance will be horizontal and a greater number of time points will be fit. When the amplitude envelope changes rapidly, a significant portion of the distance along the curve will be vertical, and a smaller number of time points will be selected for the linear fit. This adaptive window size aids in the detection of rapid changes in amplitude.

Segment boundaries are set as above threshold peaks in the angle curve. First, a threshold function is specified (see below). For each period of the curve that is continuously above the threshold, the time corresponding to maximal angle within the period is assumed to correspond to a note boundary (Fig. 1, D & E). Using a fixed threshold has the problem that changes in amplitude often scale with the overall amplitude. For example, a change in amplitude at low power might indicate a note boundary whereas the same change in louder portions of the song would not. Based on this observation, a significant improvement in performance is obtained by increasing the angle threshold at high amplitude periods. We used the following threshold function (Fig. 1C):

$$Threshold(t) = (1 + B) \times \pi + k \times AE(t)$$

where B is the threshold base level, $AE(t)$ is amplitude envelope vector and k is the amplitude scale factor. Finally, based on human segmenting experience, if a boundary is found to be too close to the onset or offset of syllables, it is usually ignored. In the algorithm, any peaks within 5 msec of a syllable boundary are ignored.

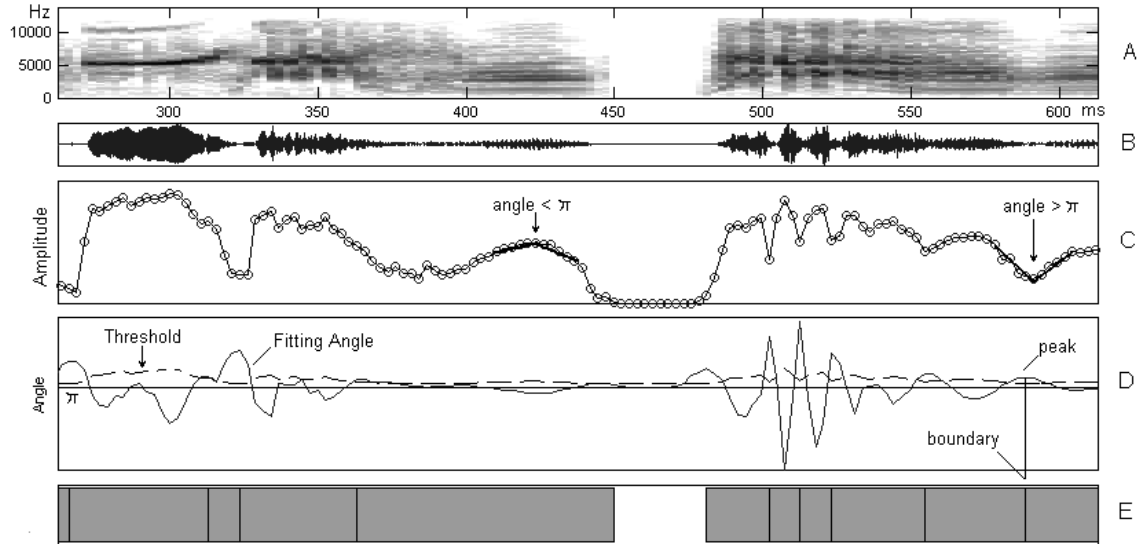


Fig. 1. Note segmentation algorithm. A. Spectrogram (time-frequency plot) for a section of a song. B. Song waveform (high-pass filtered at 500 Hz). C. Amplitude Envelope (RMS). D. Solid line: Angle calculated from linear fitting. Dashed line: Angle threshold adjusted by power level. E. Segments determined by the algorithm.

Results

To evaluate the performance of the algorithm, we compared the results to human segmentation. This testing dataset includes a large repertoire of song types: 76 sample songs, each from a different bird, were selected and manually segmented. The hand segmentation was performed by the author

with the aid of customized, Matlab-based software named SongBrowser. Only the amplitude envelope is shown during labeling so that other information such as the spectrogram will not affect the decisions of the human observer. Hand segmentation resulted in a data set with 5087 segments, with 2995 segment boundaries falling in the middle of syllables and the remaining 2092 segment boundaries corresponding to syllable boundaries. Average segment length was 38.1 msec. In addition to marking the time of presumptive note boundaries, each boundary was given a confidence level ranging from 1 to 3, with 1 being least confident and 3 being most confident.

Performance of the algorithm is governed by three parameters: L , fixed length along the curve for linear fitting; B , the base level of threshold; and k , the amplitude adjustment factor. We used the average length of segments as measure of the overall sensitivity of segmentation: a more sensitive criterion will yield more segments and hence a shorter average length, a less sensitive algorithm will yield a longer average length.

Other performance measures relied on a process of determining the “best matched” boundary. This is a directional process, i.e. one can start with the boundaries set by the automated algorithm (referred to as an “automated boundary”) and determine the closest hand segmented boundary (a “hand boundary”). Alternatively, one can start with the set of hand boundaries and determine the closest automated boundary. Since the hand and automated segmentation schemes share syllable boundaries, the starting set of matches only includes interior boundaries. However, the closest matching boundary could be a syllable boundary. For example, if the automated algorithm determined that a boundary is set at 6ms before the end of a syllable, but a human did not put a boundary there, then the nearest match to the automated boundary is the syllable-ending boundary and the match distance is 6 ms.

A second performance measure used was the average distance to the best match. To get an overall measure, we computed the average distances calculated from hand-to-automated matches and from automated-to-hand matches, and averaged these two values. Histograms of the two processes for optimal performing parameters (see below) are shown in Fig. 2. From these histograms plus an ad hoc examination of a number of song spectrograms, we determined that 8 msec was a reasonable cut off to determine a “match.” If the best match boundary was more than 8 msec distant, then this was deemed a “mismatch.” Our third performance measure was the average match percent, calculated as the average of the hand-to-automated and automated-to-hand match percentages.

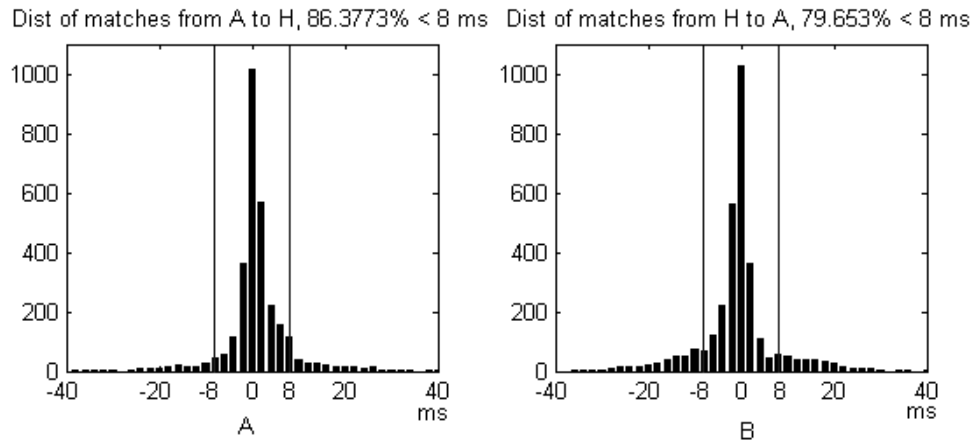


Fig. 2. Histograms of distance between matched pairs. Left. 86.4% of all hand boundaries match an automated boundary to within 8 msec. Right. 79.7% of all automated boundaries match a hand boundary to within 8msec.

In the effort to find parameters that yield optimal performance, we ran a batch process in the 3-dimension parameter space. To compare with fixed window algorithms (not shown) the length L along the curve for determining window length was stepped in units of $\sqrt{2}$. This corresponds to steps of one time bin when the slope of the amplitude envelope is 45 degrees. Parameters searched

were as follows: L from $4 \times \sqrt{2}$ to $8 \times \sqrt{2}$ in steps of $\sqrt{2}$, B from 0.01 to 0.05 in steps of 0.01, and k from 0.2 to 0.8 in steps of 0.2, for a total of $5 \times 5 \times 4 = 100$ combinations. Optimal performance as measured by the average distance measure was 4.476 msec and was achieved at $L = 5 \times \sqrt{2}$, $B = 0.01$ and $k = 0.6$. The dependence on window length L (using optimal values of B and k for each L) is shown in Fig. 3. At the optimal value, the auto-to-hand average distance was 4.83 msec and the hand-to-auto average distance was 4.12 msec. Optimal performance, as measured by the average match percent measure, was 83.5% and was achieved at the same parameters that were optimal for the distance criterion. At the optimal value, the auto-to-hand percent match was 80.09% and the hand-to-auto percent match was 86.78%. The overall average segment length was 38.193 msec (compared to 38.107 msec for hand segments).

At these parameters, we also broke down the hand-to-auto performance as a function of confidence level. Hand boundaries labeled with confidence levels of 1, 2, and 3, (least, moderate and most confident) had hand-to-auto match percentages of 74.2%, 87.6% and 93.4% respectively. The corresponding average match distances were 9.59 msec, 4.42 msec, and 3.09 msec.

One common error that we noticed was that automated algorithm often segmented low amplitude segments near the beginning/end of syllables that the human observer lumped in with the first/last note of the syllable. To quantify this we determined that of all auto-to-hand mismatches, 48.7% had the closest matching hand boundary fall at the edge of a syllable (31.2% at the beginning and 17.5% at the end; percentages were calculated based on the optimal parameter set).

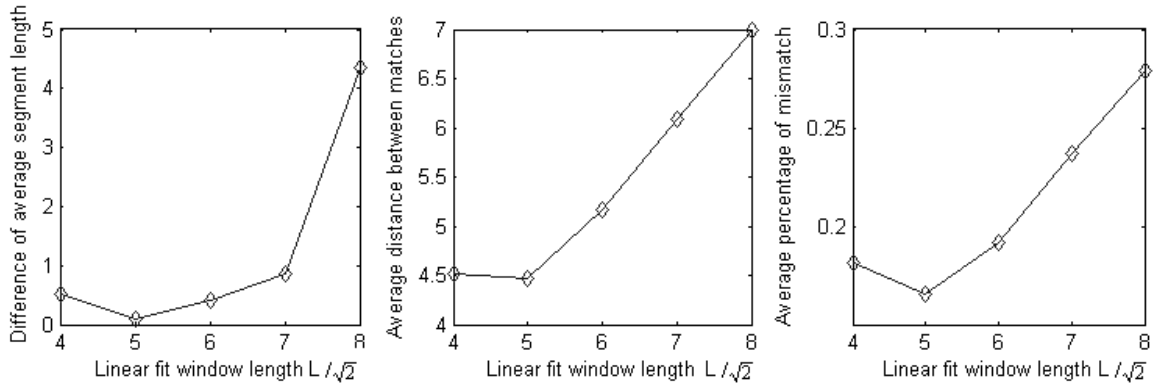


Figure 3. Performance of automated segmentation as a function of the size of the linear fitting window.

Finally we note that although most changes in spectral structure are accompanied by a change in amplitude, there are a number of exceptions (e.g. Figure 1 near 400 msec). Because our algorithm is based solely on amplitude changes, such changes are not detected.

Conclusion

In this study we presented an automated method to segment zebra finch song at the note level. The algorithm uses linear fitting algorithms applied to the RMS amplitude envelope, and is therefore extremely computationally efficient. In the field of human speech processing, there are several mature methods for segmenting speech into phonemes. However, birdsong, especially the zebra finch song we study here, has a simpler and more invariant amplitude structure relative to speech. Thus even though our algorithm relies solely on the amplitude envelope, it was able to match segmentation performed by a human observer with greater than 83% accuracy using a tolerance of 8 msec. Furthermore, nearly half of the auto-to-hand mismatches occurred when an auto boundary is matched to a syllable boundary, suggesting that a significant portion of the errors in the algorithm

are due to misclassification of the hard to segment, low amplitude sounds found at the beginning and end of many syllables.

The algorithm holds substantial promise as a front end for more computationally intense algorithms analyzing the full spectro-temporal representation of song. Thus far we have assessed the performance of the algorithm by averaging hand-to-auto and auto-to-hand performance measures. Parameters were chosen as a compromise to minimize a combination of “false positive” and “false negative” results. In the context of a more sophisticated algorithm, the role of the current algorithm would be to identify candidate boundaries that could then be evaluated using more stringent criteria, such as a rapid change in spectral features. In this role, the algorithm should be run with a low threshold so that nearly all “true” note boundaries would be included as candidates. The algorithm could also be run with a very high threshold. Candidate boundaries that passed this high threshold could be confidently classified as note boundaries based on amplitude information alone. Current work is underway to assess the feasibility of this approach.

Acknowledgements

Support contributed by a Sloan Research Fellowship, and NIMH 1R21MH66047-0.

References

- [1] Jeffery Cynx, Experimental Determination of a Unit of Song Production in the Zebra Finch, *J of Comparative Psychology*, Vol 04 , No 1 (1990) 3-10.

- [2] S. Deregnacourt, P.P. Mitra, O. Feher, C. Pytte, O. Tchernichovski, How sleep affects the developmental learning of bird song. *Nature*, Vol. 433, No. 7027. (2005) 710-716.
- [3] M. Franz, and F. Goller, Respiratory units of motor production and song imitation in the zebra finch. *J Neurobiol.* 51(2) (May 2002) 129-141.
- [4] Petr Janata, Quantitative assessment of vocal development in the zebra finch using self-organizing neural networks. *J. Acoust. Soc. Am.* 110 (5), Pt. 1, (Nov. 2001).
- [5] 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. *J. Acoust. Soc. Am.* 103 (4) (April 1998).
- [6] Christopher B, Sturdy, Leslies S. Phillmore, and Ronald G. Weisman. Note Types, harmonic Structure, and Note Order in the Songs of Zebra Finches. *Journal of Comparative Psychology* Vol 113 No. 2 (1999) 194-203.
- [7] O. Tchernichovski, P.P. Mitra, Towards quantification of vocal imitation in the zebra finch. *J Comp Physiol.* 188 (2002) 867–878.
- [8] O. Tchernichovski, P.P. Mitra, T. Lints, , F. Nottebohm, Dynamics of the vocal imitation process: how a zebra finch learns its song. *Science*, 291 (2001) 2564-2569.