Automatic extraction bat call parameters for automatic species identification

Author: Volker Runkel and ecoObs GmbH

Introduction

Acoustic detection of bats and identification of species based on recorded calls is one of the most powerful tools for assessing bat activity within the landscape. Thus it is widely used throughout the world. While call quality has the highest impact on the species identification, the parameters extracted from calls are of similar importance. Most common is the extraction of parameters after transferring the time domain into the frequency domain by using FFT or similar algorithms. From this representation call parameters are then extracted. Alternatively the parameters can be extracted from the time-domain. Any automatic process has in common, that it must run without user interaction. Before measurements can be taken, such parts of the recording have to be identified that hold possible bat calls.

Using parameters with little descriptive power or such overlapping highly between species will lower identification results. Similarly the number of parameters should be high enough to allow distinction of species. Depending on the used statistical or mathematical algorithm there may be an upper limit to the number of parameters to avoid overfitting. Furthermore nowadays tools are available that can use images of calls - usually spectrograms - and by means of Machine Learning algorithms identifies species based on the spectrogram images. Again the call quality and the recording quality have a strong influence on the accuracy of these methods.

In this document we will show how to automatically measure bat calls and use a slightly simplified version of a algorithm we have developed in 2004. We will not cover the actual algorithm in full detail, but give an overview of a possible approach that results in stable measurements for most calls. The sources exist in a github repository (CallMeasurements2_0) this document is part of.

Requirements for automatic call measurements

The first step to extract parameters of bat calls is to identify which part of the recorded sound is a bat call. While this may seem simple, it nevertheless is not always easy to achieve using an automatic system. While a simple algorithm may use amplitude information only, it may miss calls that are of low amplitude or it overachieves and will result in many non bat call noises to be found. While the first is only a small problem and has the advantage of only returning loud calls, the second situation leads to non bat calls measured nonetheless and reduces the accuracy of species identification.

Recording quality is influencing the automatic parameter extraction rather strongly. The better the SNR the easier the extraction of parameters can be done despite the used algorithm. So the goal must be to create recordings of high quality if automatic species identification should give reliable results. This also is true for manual identification, yet manual analysis allows usage of calls with less good quality as well to some degree. Apart from the SNR the amount of echoes and overall disturbances within the sound file plays a crucial role. Calls with echoes overlapping the actual call will be very hard to measure automatically. In many situations this will result in very poor if any measurements.

We will now show an example of a rather simple yet powerful algorithm we have developed for measuring bat calls.

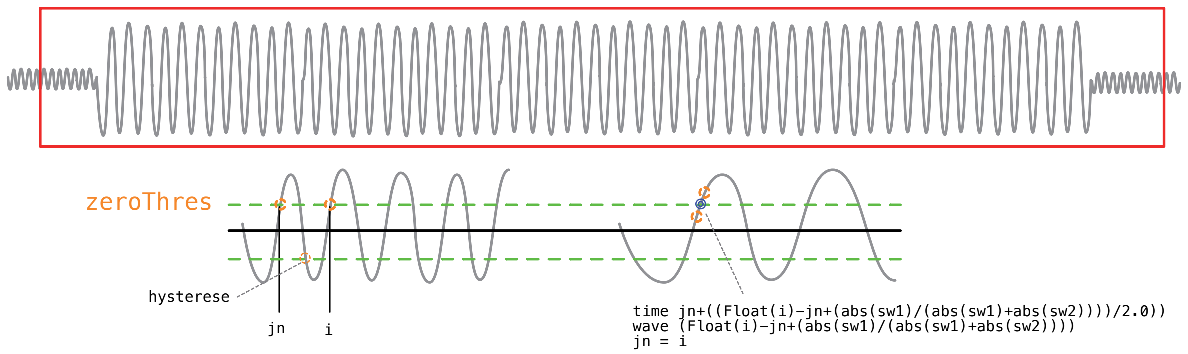
Time domain based parameter extraction

Current computers are powerful enough to calculate FFTs very very fast and thus allowing to use tools, we have dreamed of only 10 to 15 years ago. Nevertheless not always the most sophisticated and computing intensive algorithm will give the best results. Often other solutions work well and use much less computing power. We will show you an algorithm working directly in the time domain. That way it does not need to use FFT or similar computing intensive methods.

The call finder source code or working is not included in the project. We added a library that is closed source and is called from the example application. The call finding algorithm can easily be substituted by another library of your choice. It needs to deliver call positions for the measurement algorithm to work.

Building a zero-crossing table

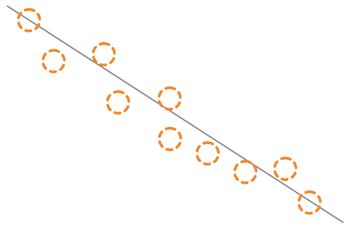
The algorithm is looping over all found possible bat calls within a recording. For each segment of sound that may contain a bat call, a zero crossing analysis is done. To improve accuracy a hysteresis is used and the zero line is moved to avoid measurements of noise. The results are further improved by analysing the proportions of the two samples - one before and one after the crossing - and the crossing is calculated at subsample accuracy.



The image shows the process of sound analysis and building a table of zero crossings. It also shows the noise cancelation and hysteresis implementation.

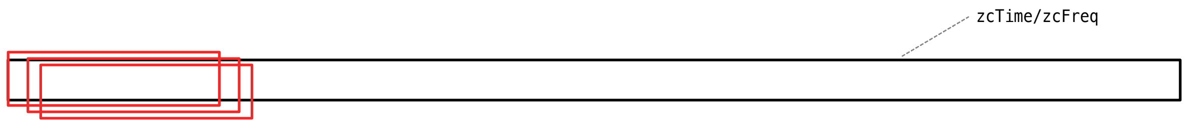
Regression analysis

The following process calculates regressions and the deviation of the samples within small time windows. These are moved through the zero-crossing table data and only data with high reliability is kept. This keeps mostly sinusoidal sound data and eliminates most of the broadband noise data. The resulting data is used to find actual calls and extract measurements. Using a simple regression function allows a rather powerful removal of unwanted wave from the actual data.



Finding call start and call end

The call start and end are evaluated using a regression on the stored frequency/time data. Subsequent regressions are calculated over small windows of the stored data. The window is moving in single steps through the full zerocrossing table.



A certain rsquare value needs to be obtained for data resembling an actual call. Each possible call is stored for measurement interpolation.

Interpolation of call measurements

The call start and end are evaluated using a regression on the stored frequency/time data. The call is then interpolated based on the available frequency/time data from the previous steps. For the representation of a call measurements are interpolated in 0.1ms steps. These can be plotted to show the call curvature. Furthermore various other derived values are calculcated. For example histogram values for 1 kHz bins (part of the frequency range in 2 kHz bins) as well as locations of most/least steepness. These values then all can be used for species identification. Further details can be seen in the source code.