Fast Chirplet Transform feeding CNN, application to orca and bird bioacoustics

Herve Glotin[*] **Julien Ricard**
Department of Computer Science
Toulon University
Toulon 83130 - France
glotin@univ-tln.fr
julien.ricard@gmail.com

Randall Balestriero
Ecole Normale Superieure Cachan
61, avenue du Président Wilson
94235,France
randallbalestriero@gmail.com

Abstract

Advanced soundscape analysis or machine listening are requiring efficient time frequency decompositions. The recent scattering theory is offering a robust hierarchical convolutional decomposition, nevertheless its kernels need to be fixed. The CNN can be seen as the optimal kernel decomposition, nevertheless it requires large amount of training data. This paper aims to show that Chirplet kernels are providing good constant Q time-frequency representation which yields to a better CNN classification than usual log-Fourier representation. We first recall the main advantages of the Chirplet concerning bioinspired auditory processing. Then the contributions of this paper are (1) to give a new fast implementation of the Chirplet by decreasing its complexity. (2) We validate fast Chirplet computation on nearly real-time over long series of orca online monitoring recordings, and on bird songs on hundreds of birds species. (3) We demonstrate that the Chirplet is improving convolutional neural net classification on complex overlapping bird calls challenge compared to usual Mel representation. Validations are conducted on a subset of the Amazon bird species of the LifeClef 2016 classification challenge.

1 Introduction

Representation of Bioacoustic sequences started with 'Human' speech in the 70'. Speech has been represented and processed during 50 years, yielding to standard and efficient Mel Filter Cepstral Coefficients (MFCC). Today new paradigms come from environmental monitoring and species classification, like bird and whales [3,5,8]. Most of the challengers used Fourier decomposition, and small number of systems have been designed by Convolutional Neural Net (CNN) [13] from the raw audio as are the most advanced human speech recognizer.

Several evidences suggest that auditory cortical neurons are tuned to complex time varying acoustic features and that auditory cortex consists of several fields that decompose sounds in parallel. One main difficulty is that such decomposition may vary across species and usual computational auditory cortical processing cannot capture such complexity.

A difficulty in learning such optimal filters by CNN is the lack of clean bioacoustic data samples in which many species overlap at the same time.

This paper summarize the properties of Chirplet decomposition and gives a new fast implementation allowing to run it online. Third, we illustrate the SNR enhancement on real recordings of whale and

[*]http://glotin.univ-tln.fr *not* We thank SABIOD MI CNRS and Institut Universitaire de France for their support.

birds, and we give a perspective with CNN classification showing that Chirplet performs better than Mel Fourier representation.

2 Chirplet

2.1 Neurophysiological motivations

Cortical neurons represent sound efficiently [1] and electrophysiological studies have demonstrated that ring patterns in specific neural regions can often be predictably correlated with particular sound features [2], even if the underlying neural codes that give rise to such correlations remain unclear. However the main properties of auditory cortex are [1]: complex patterns of sound feature selectivity, species-specific signal decomposition, Dynamic modulation of response characteristics.

The Chirplet transform appears to be well suited for our purposes: it subsumes both Fourier analysis and wavelet analysis, providing a broad framework for mapping one-dimensional sound waveforms into an n-dimensional auditory parameter space. It offers the processing that have been described in previous section for different auditory fields, i.e. cortical regions with systematically related response sensitivities.

Moreover, Chirplet spaces are highly over-complete because there are an infinite number of ways to segment a time-frequency plane, the dictionary is redundant: this corresponds well with the overlapping, parallel signal processing pathways of auditory cortex.

2.2 Formal definition of Chirplet

A chirplet can be seen as a complex sinus with increasing or decreasing frequency modulated by a Gaussian window. It is a generalization of wavelets which are a special case of chirplet with a sinus of constant frequency. When extending the support of the modulated Gaussian, one can also retrieve the Fourier basis. As a result, and as presented in [12], the chirplet transform is a generalization of many known time-frequency representations. We first present briefly the wavelet transform framework to extend it to chirplets. Given an input signal x one can compute a wavelet transform [14] through the application of multiple wavelets ψ_λ. A wavelet is an atom with compact support in time and frequency domain which integrates to 0. The whole filter bank is derived from a mother wavelet ψ_0 and a set of dilation coefficients following a geometric progression defined as $\Lambda = \{2^{1+j/Q}, j = 0, ..., JQ-1\}$ with J being the number of octave to decompose and Q the number of wavelets per octave. As a result, one can create the filter-bank as the collection $\{\psi_0(\frac{t}{\lambda}) := \psi_\lambda, \lambda \in \Lambda\}$. After application of the filter-bank, one ends up with a time-scale representation, or scalogram, $Ux(\lambda, t) := |(x \star \psi_\lambda)(t)|$ where the complex modulus was applied in order to remove the phase information and contract the space. It is clear that a wavelet filter-bank is completely characterized by its mother wavelet and the set of scale parameters. However,

Chirplets define a general class of filters with wavelets and Fourier bases as part of this large functional space [4]. These filters are generated from a Gaussian window determining the time support, and a complex sinus with non constant frequency over time with center-frequency f_c. Since the scope of the parameters leads infinitely many different possible filters, we have to restrain ourselves and thus create only a fixed chirplet filter-bank allowing fast computations.The parameters defining these filters include the time position t_c, the frequency center f_c, the duration Δ_t and the chirp rate c :

$$g_{t_c,f_c,\log(\Delta t),c}(t) = \frac{1}{\sqrt{\sqrt{\pi}\Delta t}} e^{-\frac{1}{2}\frac{(t-t_c)^2}{\Delta_t^2}} e^{j2\pi(c(t-t_c)^2+f_c(t-t_c))}. \quad (1)$$

We now present the strategy we adopted to select a priori the parameters used for our chirplet filters.

2.3 Proposition of a fast algorithm for Chirplet computation

The parameter space is basically of infinite dimension. Similarly to continuous wavelet transform however, it is possible to use some a priori knowledge in order to create a finite bank-filter. For example, wavelets are generated by knowing the number of wavelets per octave and the number of octave to decompose. As a result, we used the same motivation in order to reduce the number of possible Chirplets required.

The goal here is not to compute an invertible transform but rather provide a redundant transformation highlighting transient structures. As a result we chose to chose keep the same framework as for wavelets with the Q and J parameters. Standard parameters to analyze bird songs are $J = 6$ and $Q = 16$ with a sampling rate of 44100Hz. In addition, the chirp rate was proportional and the duration was included in it too. Finally, since we are interested in frequency modulations, we compute the ascendant and descendant chirp filters as one being the symmetrized version of the other.

As a result, we chose to use a more straightforward analytical formula defined with a starting frequency F_0, an ending frequency F_1 and the usual wavelet like parameters σ being the bandwidth. Finally another hyper-parameter fixed constant for the whole bank-filter generation is the parameter p defining the order of the chirp. The case $p = 1$ leads to a linear chirp, $p = 2$ to a quadratic chirp. The starting and ending frequencies are chosen so that they approximately cover one octave and are directly computed from the λ parameters. The λ parameters define the scales.

$$\Lambda = \{2.0^{1+i/Q}, ii = 0, ..., J \times Q - 1\} \tag{2}$$

$$F_0 = \frac{Fs}{2\lambda}, \lambda \in \Lambda \tag{3}$$

$$F_1 = \frac{Fs}{\lambda}, \lambda \in \Lambda \tag{4}$$

$$\sigma = 2\frac{d}{\lambda}, \lambda \in \Lambda \tag{5}$$

3 Fast Chirplet Transform

We propose a new Fast Chirplet Transform (FCT) taking advantage of the a priori knowledge for the filter-bank creation and the fast convolution algorithm discussed in 3.2. Therefore, we first create the Chirplet with the ascendant and descendant versions in once with:

```
INPUT: F0,F1,Fs,sigma,p
OUTPUT: coefficients_upward,coefficients_downward
if(p):
    w=cos(2*pi*((F1-F0)/((p+1)*sigma**p)*t**p+F0)*t)
else:
    w=cos(2*pi*((F0*(F1/F0)**(t/sigma)-F0)*sigma/log(F1/F0)))
coefficients_upward=w*exp(-((t-\sigma/2.0)**2)/(2*sigma**2))
coefficients_downward=flipud(coefficients_upward).
```

3.1 Filter bank

Then we generate the whole filter-bank by using the previous algorithm with the defined λ and hyper-parameters:

```
INPUT: J, Q, Fs, sigma, p
lambdas           = 2.0**(1+arrange(J*Q)/float(Q))
start_frequencies  = (Fs /lambdas)/2.0
end_frequencies    = Fs /lambdas
distances          = 2.0*d/flipud(lambdas)
filters=list()
for f0,f1,d in zip(start_frequencies,end_frequencies,distances):
    filters.append(chirplet(Fs,f0,f1,d,p))
return filters.
```

Finally, we also used the scattering framework. Mallat et al. [10,11] developed the scattering coefficients Sx which are the result of a time-averaging on the time-frequency representation Ux bringing local time-invariance. This time-averaging is computed through the application of a scaling function ϕ which is usually a Gabor function with specified standard deviation. As a result, one computes these coefficients as $Sx(\lambda, t) = (|x \star \phi_\lambda| \star \phi)(t)$. Similarly, we performed local time-averaging on the chirplet representation in the same manner.

We present some possible filters in Fig. 2.

3.2 Optimal FFT windowing for FCT

The third step in our FCT consists in the reduction of the convolution task. The asymptotic complexity of the Chirplet transform is $O(N.\log(N))$ with N being the size of the input signal. This is the same asymptotic complexity as for the continuous wavelet transform and the scattering network. However, it is possible to reach lower asymptotic complexity simply by a division of the convolution task. usually the convolutions are carried through application of an element-wise multiplication of the signal and the filter in the frequency domain and then compute the inverse Fourier transform to end up with $x \star \psi_\lambda$.

However, if we denote by M the length of the filter ψ_λ it is possible to instead perform multiple times this operation on different overlapping chunks of the signal to then concatenate the results to obtain at the end the same convolution result but now in $O(N.\log(M))$. Finally a last improvement induced by this approach is to allow easy tackling of signals with a length just above a power of 2 which otherwise would require to be padded in order to obtain a FFT with real $O(N.\log(N))$ complexity through the Danielson-Lanczos lemma [9].

Applying this scheme allowed to compute the convolutions between 3 to 4 times faster. The variations came from the distance between N and the closest next power of 2 depending on the desired chunk size.

4 Chirplet versus FFT on real bioacoustic scenes of orca and whales

We defined a simple procedure to optimize the appropriate basis functions of the Chirplets according to the maximisation of the SNR gain of the time frequency targets in the Fourier domain versus in the Chirplet decomposition. We validated also the efficiency of our fast implementation on real bioacoustic recordings. First we processed on 10 usual CPUs (4 year old) in 2 days the whole 999 bird species data set from LifeClef bird challenge (16 kHz Sampling Rate, 16 bits, nearly 100 hours of recordings). Second, we processed in 7 days the equivalent of 1 month of continuous recordings of orca whale from Orcalab.org ONG project (22 kHz SR, 16 bits). Chirplet samples are depicted Fig. 1 and 2.

5 CNN on Chirplets outperforms CNN on Mel FFT

We train CNNs [7] on the Lasagne Theano plateform (LISA Montreal Univ.), on a subset of LifeClef 2016 bird classification challenge that was extracted for ENS Ulm data challenge 2016, on the 3 species numbered [26,45,9], for a total of 15 minutes of recordings including testing and training data sets. The baseline MFCC plus randomforest classifier on this task gives an accuracy of 73% (source: ENS data challenge org.)

We then train two different CNNs. A baseline CNN is trained on a simple log of the simple 64 channel mel scale of FFT spectrum (we simply run the usual tool from Columbia univ. http://pydoc.net/Python/librosa/0.2.0/librosa.feature/). A second CNN is trained on our Chirplet representation as depicted above. The parameters of both CNN are similar, with 64 frequency bands each (we remove top and bottom band from the Chirplet to set to 64 bands only). Then the input layer is 64 x 86, the Conv layer of 20 filters of size 8 x 10. All activation functions are relu. Then we maxpool 2 x 2, follow the 20 filters of size 8 x 10, maxpooling, dense layer (200), dropout at 10%, with a final softmax dense layer with 3 classes and same dropout. Each CNN is trained by cross-entropy, L2 reg., with a learning rate set of 0.001.

In Fig. 3 we give the accuracy of these two CNNs with similar number and values of hyperparameters. The CNN on Chirplet, which is not ended, seems to give a better accuracy than the CNN trained on the Mel representation (which has already nearly completely ended its learning). More training will be conducted to confirm these first results.

6 Discussion and Conclusion

We have demonstrated that our novel implementation of Fast Chirplet Transform (FCT) can be computed nearly online on large data acoustic set. Second we shown that it offers a better representation than MFCC, but more interestingly than Mel filters. This can be interpreted because of the SNR increase that we motivated and illustrated above. The Fourier representation is representing all acoustic waves, but the Chirplet are enhancing the signal having strong time-frequency energy flow, which are characterising bioacoustic / communicative calls.

A perspective of this seminal work would be to integrate Chirplet parametrization into the CNN training itself, as a constrained embedded layer. Then the CNN would benefit of the optimal SNR due to adapted Chirplet representation. This might be done by developing a framework similar to a Wavelet Neural Network [6] but with Chirplet activation functions.

References

[1] E. Mercado C. Myers, M. Gluck. (2000). Modeling auditory cortical processing as an adaptive chirplet transform. Neurocomputing, 32, 913-919.

[2] N. Kowalski, D. Depireux, S. Shamma. (1996). Analysis of dynamic spectra in ferret auditory cortex: II, Prediction of unit responses to arbitrary dynamic spectra, J. Neurophys. 76, 3524-3534.

[3] A.Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B.Fisher, H. Müller. (2015). LifeCLEF: multimedia life species identification challenges, in Experimental IR Meets Multilinguality, Multimodality, and Interaction. pp 462-483, Springer International Publishing

[4] M. Steve, S. Haykin. (1995). The chirplet transform: Physical considerations. *Journal of Signal Processing, IEEE Transactions on*bf 43.11:2745-2761.

[5] H. Glotin, Y. LeCun, S. Mallat, T. Artières, C. Tchernichovski, X. Halkias. (2014). Proc. of the 1st wkp on Neural Information Processing for Bioacoustics NIPS4B, joint to NIPS, Alberta.

[6] A. Hojjat, X. Jiang. (2006). Dynamic fuzzy wavelet neural network model for structural system identification. Journal of Structural Engineering 132.1: 102-111.

[7] Y. LeCun, Y. Bengio. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), Chicago

[8] LIFECLEF 2016, http://www.imageclef.org/lifeclef/2016

[9] W. Press, B. Flannery, S. Teukolsky, W. Vetterling. (1989). Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, pp. 407-411.

[10] B. Joan, S. Mallat. (2013). Invariant scattering convolution networks. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35.8: 1872-1886.

[11] A. Joakim, S. Mallat. (2014). Deep scattering spectrum. Signal Processing, IEEE Transactions on 62.16: 4114-4128.

[12] Mann, Steve, and Simon Haykin. "The chirplet transform: A generalization of Gabor's logon transform." Vision Interface. Vol. 91. 1991.

[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[14] Mallat, Stéphane. A wavelet tour of signal processing. Academic press, 1999.

[15] Mann, Steve, and Simon Haykin. "Adaptive chirplet transform: an adaptive generalization of the wavelet transform." Optical Engineering 31.6 (1992): 1243-1256.

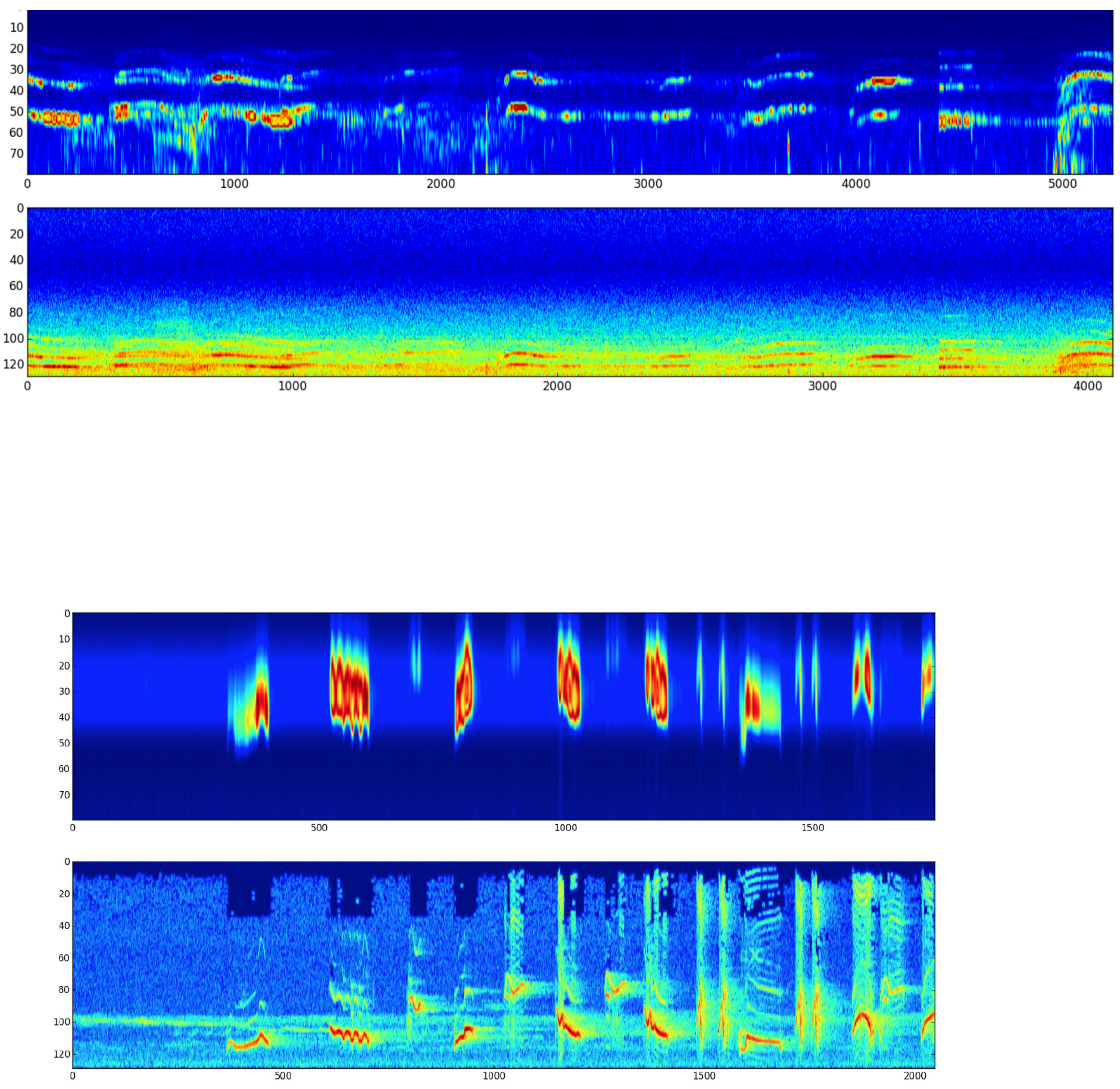
Figure 1: Top : Chirplet of orca call with p=3, j=4, q=16, t=0.001, s=0.01, with usual FFT spectrogram, showing the calls (in red) enhanced in the Chirplet compared to the FFT. Bottom the same for bird calls from Amazonia [8]. (Sampling rate 16 kHz, 16 bits.)

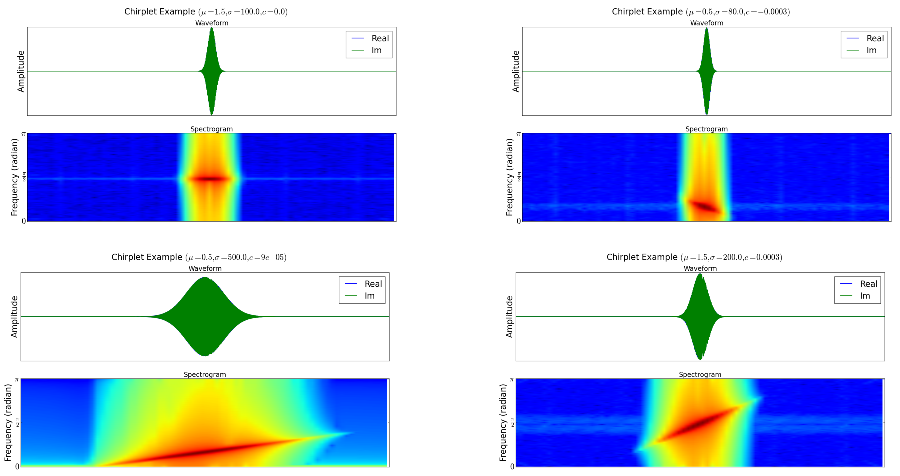

Figure 2: Some chirplets displayed in the physical domain and in the time-frequency domain through a spectrogram. The first one reduces to a wavelet since the chirp rate is 0. One can see the importance of the time duration and the chirp rate and well as the center frequency depending on what one wishes to capture.

.

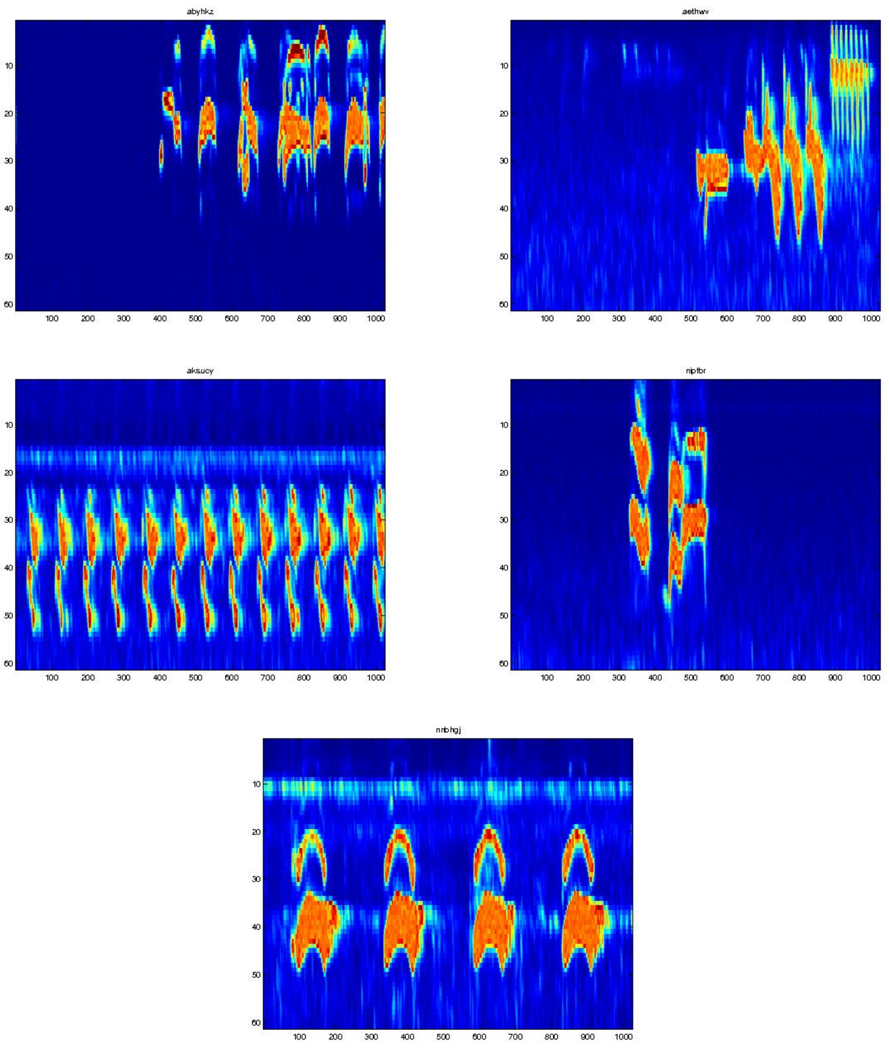

Figure 3: Chirplet representations of 5 species of amazonian birds ([8] challenge). The calls are the high SNR red regions. The species are international codes, from top to bottom, right to left : abyhkz, aethwv, aksucy, nipfbr, nnbhgj

.