Segmenting sounds: simpler version

LSS 31 3 2019.

This work is based on work I did a long time ago, but updated a bit since then. Original references are [Sound Segmentation Using Onsets and Offsets](http://www.cs.stir.ac.uk/~lss/Papers/JNMR1994.pdf), Smith L.S.

Journal of New Music Research, 23, 1, 11-23, March 1994.

Interestingly, re-reading my old papers (yes, one forgets what one write after more than 20 years) I see that in a later paper, for NIPS 1995 (*Onset based sound segmentation*, pp 729--735, in Touretzky D.S., Mozer M.C., Hasselmo M.E. (eds) Advances in Neural Information Processing Systems 8 (Proceedings of the 1995 Conference), MIT Press, 1996), I added a layer of integrate-and-fire neurons. I haven’t taken that approach here, though it might be a good idea, because it would integrate the signals across adjacent channels (where “adjacent” means whatever the topology of the network implies). But I digress: the aim of this document is to explain what this system does.

The overall concept here is:

1. Bandpass the incoming sound signal into a number of bands, using the gammatone filter-bank. One can adjust the number of bands, the lowest and highest frequencies of the centres of each band, and the bandwidth of the bands. These are set in *findsegments\_all*, with default values, over-ridable using the varargin mechanism. See 'mincochfreq', 'maxcochfreq', 'n\_erbs'.
2. Apply the onset-offset filter to each band. This is achieved by
   1. Rectifying the signal
   2. Smoothing the rectified signal (convolution)
   3. Applying the onset-offset filter (convolution)

See lines 101-103 of findsegments\_1.m. Note that the rectified signals are summed prior to smoothing or the application of the onset/offset filter. If one was going to use onset neurons, one would not do this, but would smooth and then apply the onset offset filter to each band. This would give a set of onset/offset signals, one for each band. (I must have done this in the 1990’s but I’ve no idea where the code is).

1. Use the onset/offset signal generated to do the segmentation. Currently, this is done by
   1. Finding the onset peaks (peaks of the positive signal) and the offset peaks (peaks of -1 \* onsetoffset signal).
   2. Use these to segment the signal in a slightly ad-hoc way.

This all works reasonably well. I enclose an extra function,