ch_xcorr.m

Chris Hummersone*

June 5, 2016

This document briefly explains the workings of the `ch_xcorr` algorithm.

Algorithm

```
function [ccg,ic] = ch_xcorr(hc_L,hc_R,frameCount,frame_length,noverlap,maxlag,tau,...
    varargin)
```

```
% SYNTAX
%
%   CCG = CH_XCORR(HC_L,HC_R,FRAME_LENGTH,NOVERLAP,MAXLAG,TAU)
%   CCG = CH_XCORR(...,INHIB)
%   CCG = CH_XCORR(...,IC_T,NORM_FLAG)
%   CCG = CH_XCORR(...,INHIB_MODE)
%   [CCG,IC] = CH_XCORR(...)
```

For the purposes of this document:
`HC_L` $= \mathbf{h}_L$
`HC_R` $= \mathbf{h}_R$
`FRAME_LENGTH` $= M$
`NOVERLAP` $= N$
`TAU` $= \tau$
`INHIB` $= \iota$
`IC_T` $= \Theta_\chi$
`FRAME_COUNT = floor((max(size(HC_L))-MAXLAG-1)/(FRAME_LENGTH))-NOVERLAP+1` $= L$

The algorithm takes two matrices of data `hc_L` and `hc_R` (such as the output of a peripheral ear model) and divides it into frames of length `frame_length` (in samples). From these data, cross-correlograms (averaged cross-correlations) are calculated for each frame. The cross-correlations and cross-correlograms can be calculated in numerous ways, with or without normalisation, and with optional inhibition.

Specifically, cross-correlations are calculated in frequency channel index i, sample index n and lag index m using the input data in the following way:

$$\hat{\mathbf{c}}(i,m,n) = \frac{\acute{\mathbf{c}}(i,m,n)}{\sqrt{\mathbf{a}_L(i,m,n)\mathbf{a}_R(i,m,n)}} \tag{1}$$

where

$$\acute{\mathbf{c}}(i,m,n) = \frac{1}{\tau}\mathbf{h}_L\big(i,\max(n+m,n)\big)\mathbf{h}_R\big(i,\max(n-m,n)\big) + \left(1-\frac{1}{\tau}\right)\acute{\mathbf{c}}(i,m,n-1), \tag{2}$$

*christopher.hummersone@surrey.ac.uk

$$\mathbf{a}_L(i,m,n) = \frac{1}{\tau}\mathbf{h}_L^2\big(i, \max(n+m,n)\big) + \left(1 - \frac{1}{\tau}\right)\mathbf{a}_L(i,m,n-1), \tag{3}$$

$$\mathbf{a}_R(i,m,n) = \frac{1}{\tau}\mathbf{h}_R^2\big(i, \max(n-m,n)\big) + \left(1 - \frac{1}{\tau}\right)\mathbf{a}_R(i,m,n-1) \tag{4}$$

and τ is the time constant of the exponentially decaying window in samples. This is used to extract the Interaural Coherence (IC) χ as:

$$\chi(i,n) = \max_m \hat{\mathbf{c}}(i,m,n) \tag{5}$$

However, the data used in subsequent stages of the algorithm depends on `norm_flag`. Subsequent stages may use the normalised cross-correlation $\hat{\mathbf{c}}$ (if `norm_flag` $= 1$), or the un-normalised cross-correlation $\acute{\mathbf{c}}$ (if `norm_flag` $\neq 1$).

Following this, inhibition is applied, although this procedure depends upon the `inhib_mode` supplied to the function. The default (`'multiply'`) is a multiplication mechanism:

$$\mathbf{c}(i,m,n) = \iota(i,n)\acute{\mathbf{c}}(i,m,n) \qquad (\texttt{norm\_flag} \neq 1) \tag{6}$$

or

$$\mathbf{c}(i,m,n) = \iota(i,n)\hat{\mathbf{c}}(i,m,n) \qquad (\texttt{norm\_flag} = 1) \tag{7}$$

Alternatively, a subtractive procedure (`'subtract'`) may be used:

$$\mathbf{c}(i,m,n) = \max\left(\acute{\mathbf{c}}(i,m,n) - \frac{1}{\tau}\iota(i,n), 0\right) \qquad (\texttt{norm\_flag} \neq 1) \tag{8}$$

or

$$\mathbf{c}(i,m,n) = \max\left(\hat{\mathbf{c}}(i,m,n) - \frac{1}{\tau}\iota(i,n), 0\right) \qquad (\texttt{norm\_flag} = 1) \tag{9}$$

If no inhibition is specified then $\iota(i,n) = 1$ $(\forall\ i,n)$.

The cross-correlograms $\mathbf{C}$ for frame l are calculated by averaging only the inhibited cross-correlations within a given frame for which the corresponding IC value exceeds a threshold value Θ_χ:

$$\mathbf{C}(i,l,m) = \begin{cases} 0 & \text{if } \Psi = \varnothing \\ \dfrac{1}{|\Psi|}\displaystyle\sum_{d\in\Psi}\mathbf{c}(i,d,m) & \text{otherwise} \end{cases} \tag{10}$$

where $\{\Psi \in n : (l-1)M + 1 \leq n \leq lM, l \leq L - N + 1, \boldsymbol{\chi}(i,n) \geq \Theta_\chi\}$ and $\varnothing$ is the empty set. This can effectively be bypassed by setting $\Theta_\chi = 0$.

Note

It is recommended that if `norm_flag` $= 1$ then $\tau \gg 1$. As can be seen above, as $\tau \to 1$ then $\mathbf{a}_{\{L,R\}}(i,m,n-1) \to 0$, and hence $\hat{\mathbf{c}}(i,m,n) \to 1$.