7-3-07

Shifting formants using STRAIGHT

Eric Oglesbee

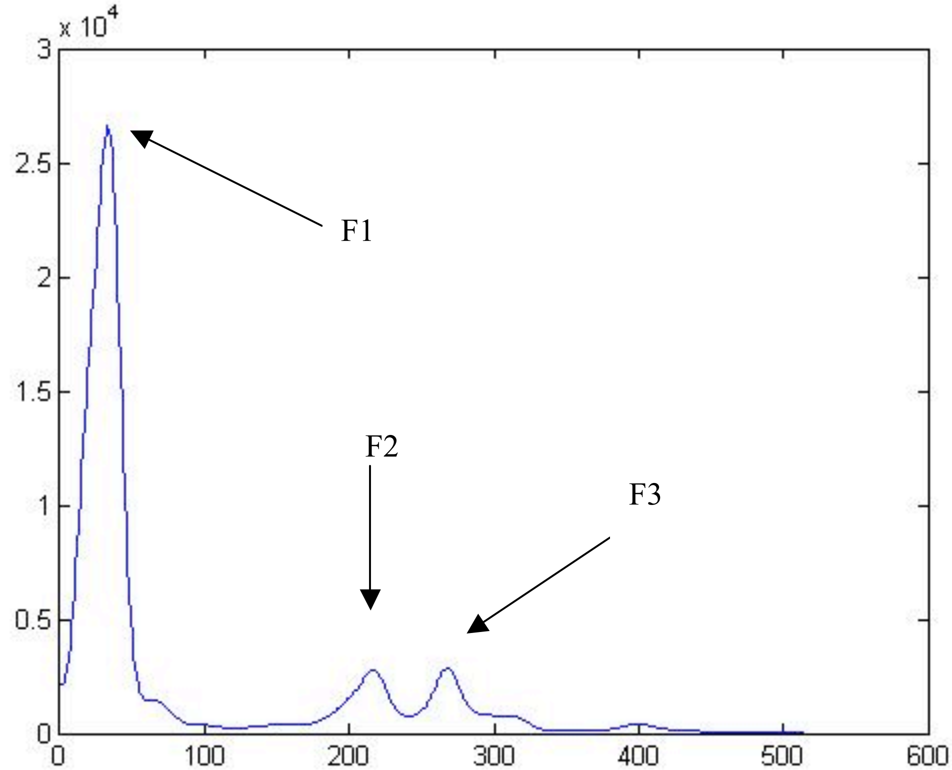
Note: This document assumes that you are using the "batch" version of STRAIGHT as opposed to the GUI version. These instructions assume a reasonable familiarity with both Matlab and the STRAIGHT manuals provided by Hideki Kawahara on his website. These instructions are provided as a service, without any guarantee of support.

1. Background: Principles behind shifting formants in STRAIGHT

STRAIGHT is essentially a large, multi-channel vocoder system with two major input components: an F0 contour, and a "smoothed spectrogram" for capturing spectral characteristics. When a STRAIGHT analysis is performed on a sound file, an F0 contour and smoothed spectrogram are created. Left unaltered, these two components can be used within STRAIGHT to synthesize a sound file that is virtually indistinguishable from the original. In order to shift formants (or make any kind of spectral modification), the original smoothed spectrogram is replaced with a modified version that is used during the synthesis process.

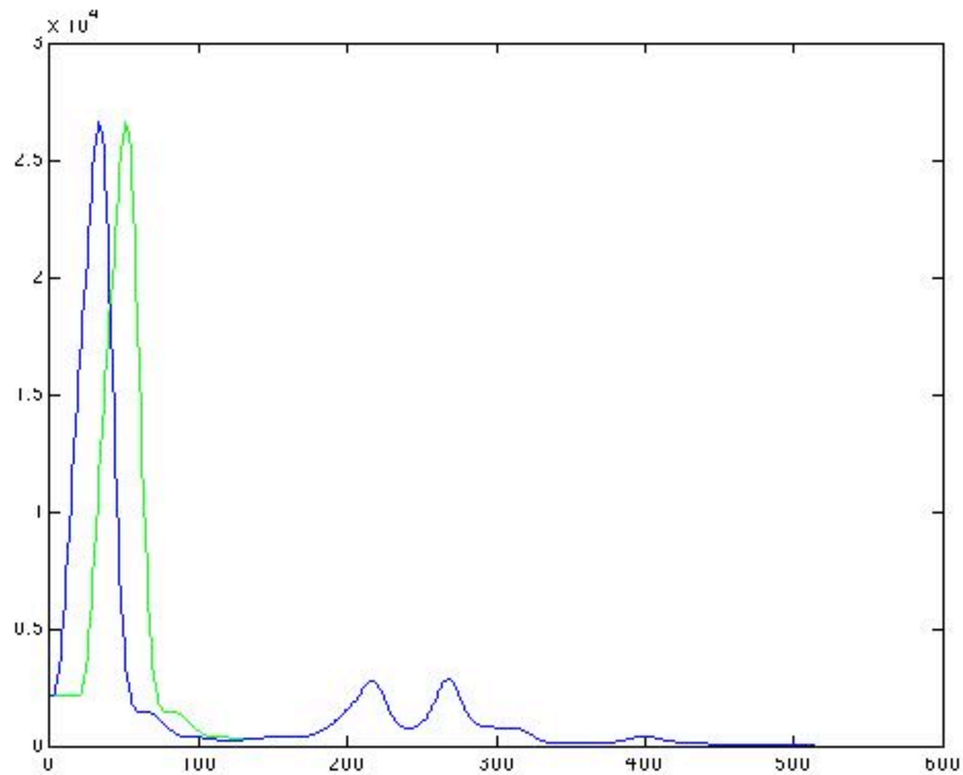
2. Smoothed spectrogram form

The smoothed spectrogram generated by the STRAIGHT analysis is stored in the variable "n3sgram". Instructions for how to get access to this variable can be found in Kawahara's STRAIGHT manual. The n3sgram variable is simply an n x m matrix, where the number of rows (n) corresponds to the number of frequency channels (fft size divided by 2), and the number of columns (m) is equal to the number of analysis steps used to create the smoothed spectrogram. Using the default settings, there are 512 rows, and each column represents a 1 ms step. The entries in the n3sgram matrix are the amplitude values for each frequency/time combination. The figure below is a plot of a single column from an n3sgram matrix. The fft bands are across the horizontal axis, with amplitude on the vertical axis. Notice the peaks for F1, F2, and F3.



3. Shifting script

The m-file “func3d_shift_0703.m” shifts a single formant peak by a fixed frequency value for a given time frame. The figure below illustrates a shift in F1 within a single time slice. The original smoothed spectral slice is shown in blue, and the shifted slice is shown in green.



Note that the only change was that the F1 peak was shifted to a higher frequency.

A short description of each of the input variables for the shifting script is given below:

`PERCENT_MAX and PERCENT_MIN:` The approach used in our lab for formant shifting was built around the dual ideas that (1) a range of stimuli would be generated, and (2) that the shift increments would be expressed as a percentage of the original formant peak value. Hence, the variables PERCENT_MAX and PERCENT_MIN are used to specify the formant values of the endpoints of the continuum that is being created as a proportion of the original formant value. For example, suppose you wanted a continuum with one endpoint at 95% of the original formant value, and the other endpoint at 110% of the original value. In this case,

PERCENT_MAX = .10
PERCENT_MIN = -.05

IMPORTANT NOTE: The actual step size in Hz is calculated using an estimation of the average formant frequency over the time interval in which the formant is being shifted. For example, suppose you have a sound file that is 200 ms long, but you are only shifting the formants in the 50 – 120 ms portion of the sound file. The shifting algorithm will calculate the average of the formant peak frequency in every spectral slice from 50 – 120 ms. The endpoints of the generated stimulus continuum will calculated relative to this value. In order to accurately predict the output values for the shifted formants, it is necessary to know this average formant value that is calculated by the script. I have placed a comment in the shifting script with instructions for obtaining this value.

`STEP`: Indicates the number of steps (including endpoints) desired for the stimulus continuum. In this implementation, steps are spaced equally in Hz.

`file:` Text file containing original smoothed spectrogram (include .txt extension)

`sf:` Sampling Rate (in Hz)

`TST:` Time (in ms) when the formant shift is to start

`TEND:` Time (in ms) when the formant shift is to end

`n_formant:` The number of the formant to be shifted (i.e. 1 = F1, 2 = F2, etc…). A better way of thinking of this though is that you are choosing which major peak in the smoothed spectrogram is to be shifted. Sometimes, the first major peak is actually associated with F0 and not F1, in which case you would want to input "2" in order to shift the first formant. The shifting script uses the following default ranges for the different formants:

F0MAX = 250;
F1MAX = 1000; %default is 1000
F2MAX = 2500;
F3MAX = 3500;
V4MAX = 4000;

Sometimes it is helpful to adjust these default ranges to more fine grained levels in order to capture the peak that you want to shift.

4. Shifting Example
The following is m-code for a 15 step continuum used in our lab:

```
%Script to make 7 steps of "n3sgram"  chang's shifting
% routine. At matlab prompt from previously stored base
% ns3gram in Straight analysis
% DKP  2/7/05
% DKP  4/11/05  Used positive first step from all formants
%that then goes negative. 7 steps, starting with

% ENO 2/17/06   Modified to produce a total of 15 steps for
discrimination
% portion of Sentclassfinal Experiment 3
```

```
new = func3d_shift_0703(.1335,.0141,15,'n3bid9d.txt',11025,155,352,1);

n3bid9d_f1_2afc_00 = new(:,:,1);
save n3bid9d_f1_2afc_00.txt n3bid9d_f1_2afc_00 -ascii

n3bid9d_f1_2afc_01 = new(:,:,2);
save n3bid9d_f1_2afc_01.txt n3bid9d_f1_2afc_01 -ascii

n3bid9d_f1_2afc_02 = new(:,:,3);
save n3bid9d_f1_2afc_02.txt n3bid9d_f1_2afc_02 -ascii

n3bid9d_f1_2afc_03 = new(:,:,4);
save n3bid9d_f1_2afc_03.txt n3bid9d_f1_2afc_03 -ascii

n3bid9d_f1_2afc_04 = new(:,:,5);
save n3bid9d_f1_2afc_04.txt n3bid9d_f1_2afc_04 -ascii

n3bid9d_f1_2afc_05 = new(:,:,6);
save n3bid9d_f1_2afc_05.txt n3bid9d_f1_2afc_05 -ascii

n3bid9d_f1_2afc_06 = new(:,:,7);
save n3bid9d_f1_2afc_06.txt n3bid9d_f1_2afc_06 -ascii

n3bid9d_f1_2afc_07 = new(:,:,8);
save n3bid9d_f1_2afc_07.txt n3bid9d_f1_2afc_07 -ascii

n3bid9d_f1_2afc_08 = new(:,:,9);
save n3bid9d_f1_2afc_08.txt n3bid9d_f1_2afc_08 -ascii

n3bid9d_f1_2afc_09 = new(:,:,10);
save n3bid9d_f1_2afc_09.txt n3bid9d_f1_2afc_09 -ascii

n3bid9d_f1_2afc_10 = new(:,:,11);
save n3bid9d_f1_2afc_10.txt n3bid9d_f1_2afc_10 -ascii

n3bid9d_f1_2afc_11 = new(:,:,12);
save n3bid9d_f1_2afc_11.txt n3bid9d_f1_2afc_11 -ascii

n3bid9d_f1_2afc_12 = new(:,:,13);
save n3bid9d_f1_2afc_12.txt n3bid9d_f1_2afc_12 -ascii

n3bid9d_f1_2afc_13 = new(:,:,14);
save n3bid9d_f1_2afc_13.txt n3bid9d_f1_2afc_13 -ascii

n3bid9d_f1_2afc_14 = new(:,:,15);
save n3bid9d_f1_2afc_14.txt n3bid9d_f1_2afc_14 -ascii
```

Each of the saved n3sgrams would then be used to replace an original n3sgram in order to synthesize the shifted stimuli (once again, see Kawahara's instructions for getting access to the n3sgram variable in synthesis).

5. Shifting visualization tools

In order to double check that the shifting process worked properly, the m-file n3movieComparison can be used to create a "movie" where each frame of the movie is a single spectral slice of a smoothed spectrogram. By displaying two smoothed spectrograms simultaneously, it is possible to quickly check for erroneous shifts. The variables for the script are:

filename1: a variable (NOT a filename) containing an n3sgram. Will be displayed in green.

filename2: a variable (NOT a filename) containing an n3sgram. Will be displayed in blue.

tStart: The time (in ms) where the comparison is to start.

tEnd: The time (in ms) where the comparison is to end.