mausmooth

1. What is *mausmooth*?

mausmooth is a simple script for *Praat* that helps you extract f0 contours from your sound files, gives you the opportunity to make manual changes to the extracted contours, and plots them in a smooth way. Depending on the aim of your study, when plotting f0 contours it might be useful to provide information on global trends (black line in Figure 1) rather than show the extracted datapoints themselves (red dots in Figure 1), which are often affected by various artefacts, e.g. the interactions with segments and phonation types.

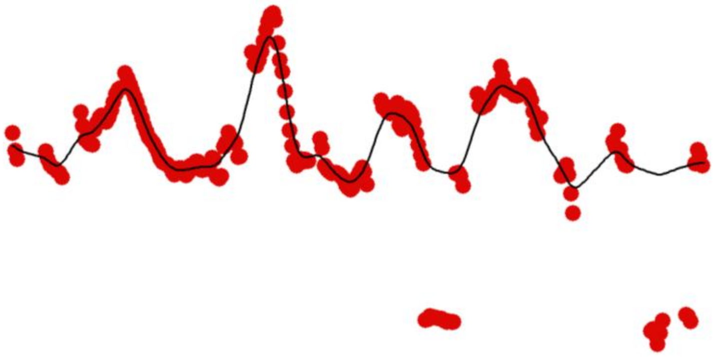


Figure 1. f0 contour for a test utterance, plotted using *Praat* standard extraction method (red speckles) and *mausmooth* (black line).

Praat already offers a selection of parameters to minimize the impact of such artefacts. But even more reliability might be needed when eyeballing large datasets in order to find differences across categories (e.g. contexts, dialects, speakers). Compare the readability of standard speckle plots (left panel in Figure 2) with that of *mausmooth*ed contours (right panel in Figure 2).

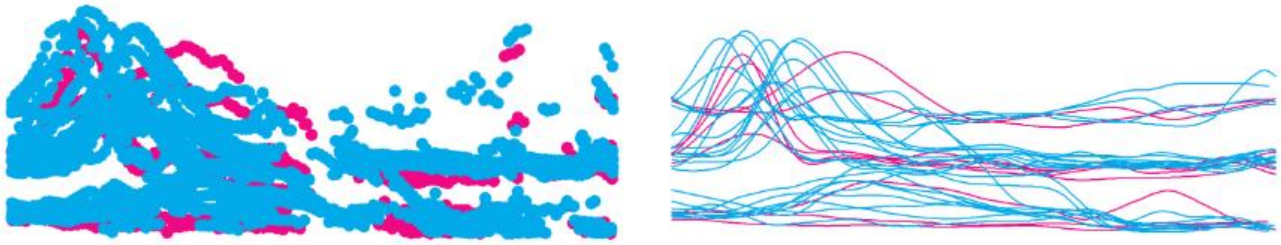


Figure 2. f0 contours for utterances from two categories (cyan vs. magenta) plotted using *Praat* standard extraction and plotting method (left panel), and using *mausmooth* (right panel).

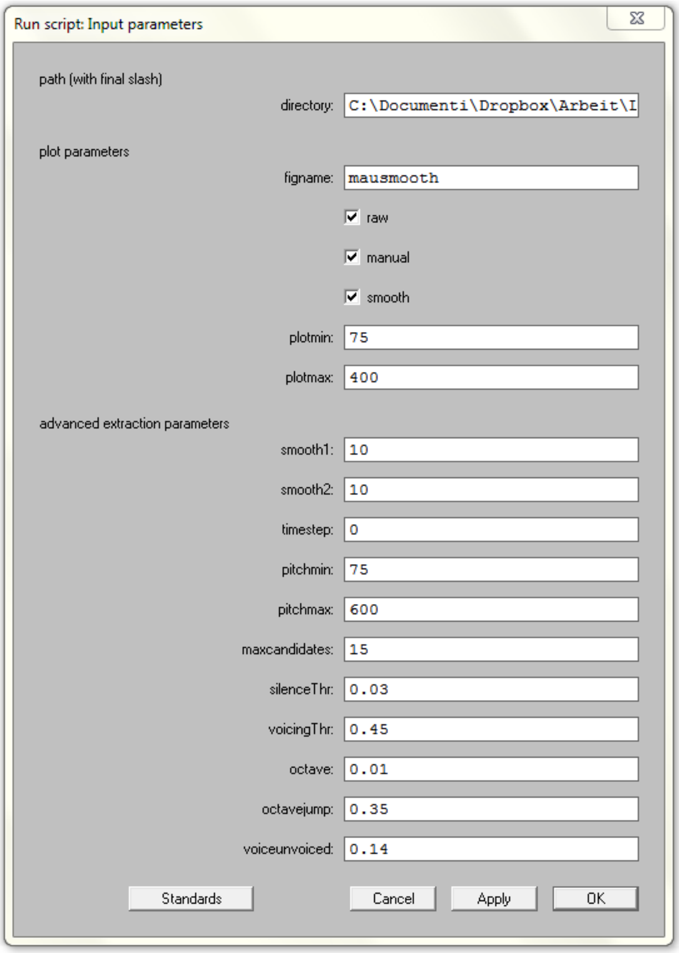
2. How does *mausmooth* work?

2.1. Script basics

As the name suggests, *mausmooth* helps you obtain *MA*nually and *AU*tomatically *SMOOTH*ed f0 contours. It goes through all sound files in a folder, visualizes the extracted f0 values, gives you the time to make manual corrections, and proceeds with smoothing and interpolation. Of course, you don't have to provide manual corrections for all files every single time you run the script. Once provided by the user, manual corrections are stored in individual `.pitch` files (with the name of the corresponding sound file). *mausmooth* only asks for a manual verification when no such `.pitch` files are already provided. This means that you will only be asked once to verify each of your files.

The script opens a form asking basically three things (see Figure 3):

- the **path** to both input and output files: the working DIRECTORY
- the **plot** parameters: the name of the output figure (FIGNAME), the vertical range of the plot (PLOTMIN and PLOTMAX) and, crucially, which combination of RAW, MANUALly corrected and SMOOTHed contours should be plotted. For example, Figure 1 has been created by ticking both raw and smooth boxes, while Figure 2 has raw contours in the left panel and smoothed contours in the right one. The advanced user can further customize the plots (e.g. line thickness, speckles colour, box size) by modifying the script directly.
- the pitch **extraction** parameters, essentially the ones discussed in the *Praat* manual for the *To Pitch (ac)* command, with the addition of two values for smoothing (SMOOTH1 and SMOOTH2), which will be discussed later.



path (with final slash)

directory: C:\Documenti\Dropbox\Arbeit\I

plot parameters

figname: mausmooth

☑ raw

☑ manual

☑ smooth

plotmin: 75

plotmax: 400

advanced extraction parameters

smooth1: 10

smooth2: 10

timestep: 0

pitchmin: 75

pitchmax: 600

maxcandidates: 15

silenceThr: 0.03

voicingThr: 0.45

octave: 0.01

octavejump: 0.35

voiceunvoiced: 0.14

Figure 3. *mausmooth* input form.

Once the input parameters are validated, the script opens the first sound file in the path, and checks whether a manual correction (a `.pitch` file) is already available. If this is the case, the script plots its f0 contour according to the plot parameters. If a manual correction is not already available, the soundfile is opened in the *Sound* window, the f0 contour is extracted and visualized in the *Pitch* window, and the script pauses to let the user provide a manual correction (see Figure 4).

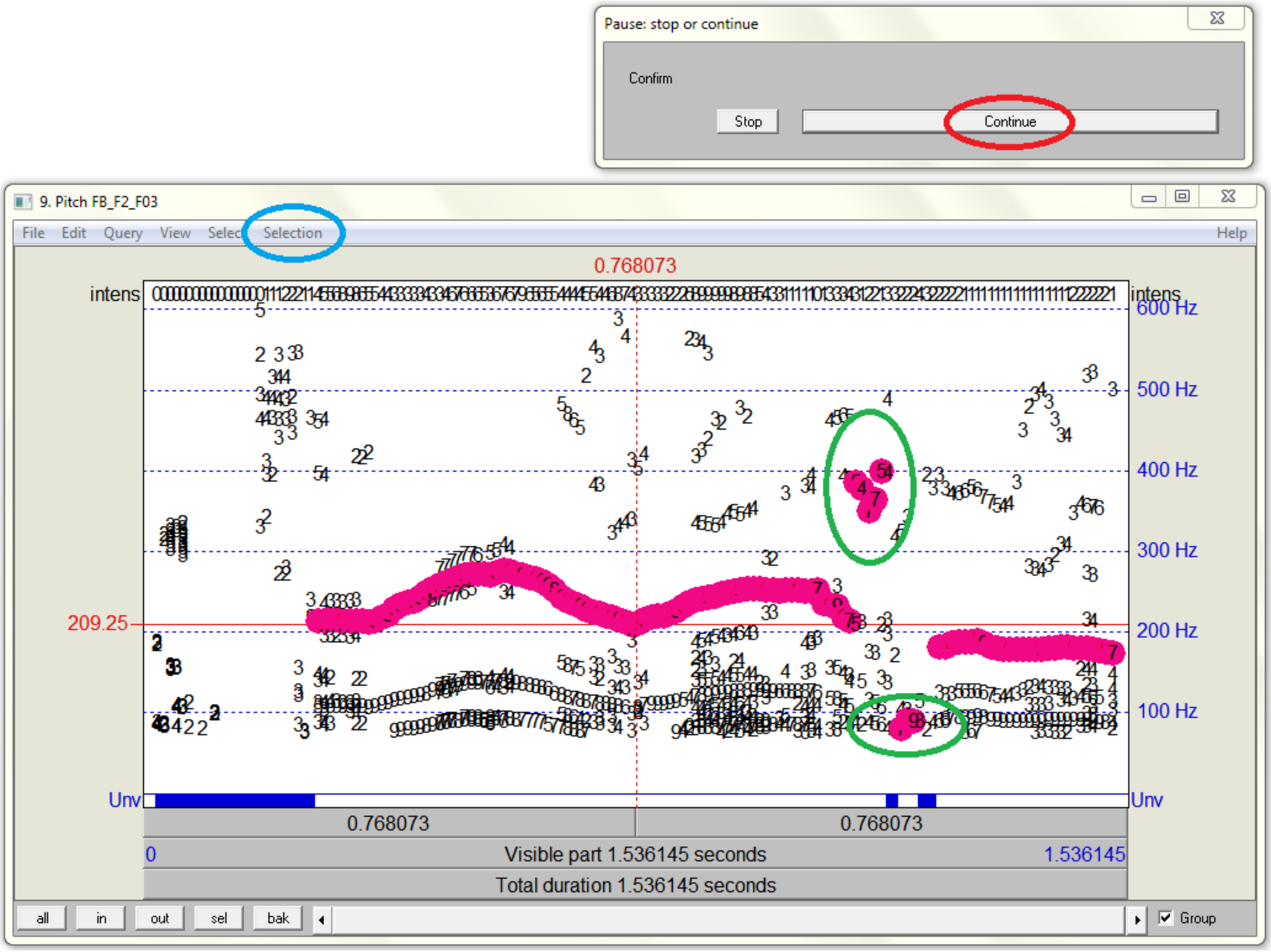


Figure 4. Pitch object editor, waiting for manual changes.

At this point the user can use the *Selection* menu of the *Pitch* window (blue ellipsis in Figure 4) to remove or change spurious pitch points (green ellipses in Figure 4). Once changes are made, the user can click on *Continue* in the *Pause* window (red ellipsis in Figure 4) to resume the script (note that the *Pause* window might end up below the *Pitch* window). The manual corrections are stored in a `.pitch` file, then the contour is smoothed, interpolated and smoothed again (using the parameters provided in the form) and stored in a `.smooth` file. The smoothed contour is then converted to a HEaderless SPreadsheet file (`.hesp`), a simple table format which can be used in other environments (e.g. *R* or *Matlab*). Then the contour is plotted in *Praat* according to the plot parameters. Once the contour is plotted, the script processes the following sound file in the path folder in the same way. When all sound files are processed, the script saves the plot with the filename provided in the input form.

2.2. Correction and smoothing choices

In order to understand how the script works, when using *mausmooth* for the first times it is useful to plot individual sound files by ticking all three boxes (RAW, MANUAL and SMOOTH) in the plot settings. Consider Figure 5:

- RAW: the f0 values extracted by *Praat* are plotted as red dots (as in Figure 1).
- MANUAL: on top of these, the outcome of the manual correction is plotted with silver dots. This has the effect of highlighting only those points which were extracted by *Praat* but were rejected in the manual correction, and which now surface as red.
- SMOOTH: the smoothed contour, which uses the manual correction as input, is overlaid as a black line.

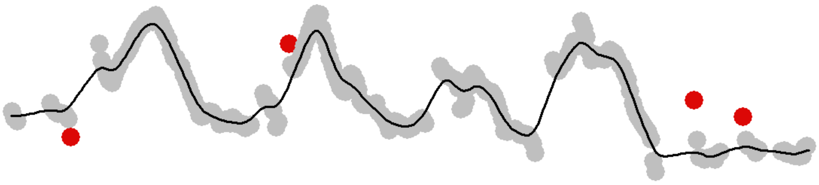


Figure 5. A single utterance plotted using all three styles.

Of course, deciding whether a given f0 point extracted by *Praat* should be retained or discarded is no hard science. In order to take principled choices, at least a basic knowledge of phonetics and signal processing is required, but manual correction is in the end a subjective matter. Some readers for example might find that another couple of points could have been rejected in the example in Figure 5 (perhaps during the course of the first rise). As always, decision criteria should be clearly stated and consistently applied. In my own practice, for example, I tend to manipulate (reject or change) only pitch points during non-modal phonation (e.g. the halved values in the second half of Figure 1) or at segmental discontinuities. In this latter case, I am more comfortable removing isolated points (e.g. the two points in the final flat stretch in Figure 5, which thus surface as red) than points in an environment of similar values (e.g. the three or four points in the first rise-fall movement in Figure 5, which are not removed and thus surface as grey).

Once the mechanics of the script are clear and sufficient familiarity with manual corrections is acquired, the script can be used to plot multiple smoothed contours only (i.e. by ticking only the SMOOTH box in the plot settings). This choice often yields the most readable display of data (see Figure 2, right panel). The degree of smoothing is controlled by a smoothing parameter. Low values (below 10) provide strong smoothing, while higher values (above 25) leave the contour largely unaffected (see Figure 6).

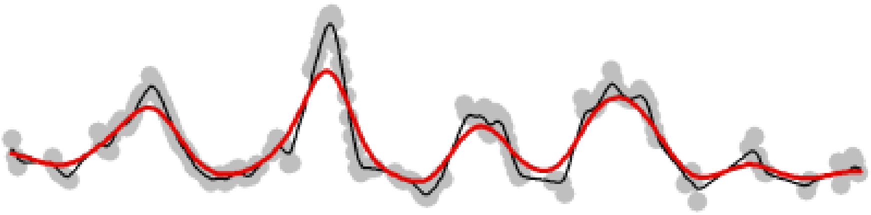


Figure 6. A single utterance, plotted using manually corrected f0 points (silver speckles) and two smoothed contours, obtained by setting both SMOOTH1 and SMOOTH2 to 20 (black line) and 5 (red line), respectively.

Two smoothing values can be specified: one for smoothing the manually corrected f0 points (SMOOTH1) and one for smoothing the contour again after interpolation (SMOOTH2). Of course, the "right" choice of smoothing values depends on the characteristics of the speech material and on the purpose of the analysis. For intonation research, I have often found that it is good to keep SMOOTH2 at 15, and set SMOOTH1 to low values (e.g. 5) when eyeballing the data, in order to retain only the information about the number and the location of the main pitch events, and to high values (e.g. 20) when extracting phonetic features, such as the alignment and the scaling of f0 peaks.

3. How do I acknowledge using *mausmooth*?

mausmooth has been developed by Francesco Cangemi at I*f*L-Phonetik (University of Cologne), integrating useful comments from Bastian Auris, Anna Bruggeman, Samuel Ebner, Henrik Niemann, Timo Röttger and Simon Wehrle. You can acknowledge its use by adding in your paper something along these lines

> f0 contours were extracted, corrected and plotted in *Praat* (Boersma & Weenink 2015), using a customized version of *mausmooth* (Cangemi 2015)

and then adding in your reference lists an entry like the following:

> Cangemi, Francesco (2015). *mausmooth*. Retrievable online at http://phonetik.phil-fak.uni-koeln.de/fcangemi.html

4. Where can I find *mausmooth*?

Here it is. Just copypaste it in your *Praat* scripting window, make all necessary changes, run and have fun.

```
# TITLE:        MAUSMOOTH (Manual and AUtomatic SMOOTHing of f0 tracks)

# INPUT:        given .wav (and potentially .Pitch) file(s)
# ELABORATION:  plot the f0 contour in (combinations of) the three following formats
#               red dots: Praat-extracted f0 values;
#               silver dots: manually corrected f0 points;
#               black line: smoothed and interpolated f0 track
# OUTPUT:       save plots as single .png file, then for each sound save
#               manual correction as .pitch and smoothed contour as .smooth and .hesp

# NOTES:        The script uses a manual correction (.pitch file) if available,
#               otherwise asks user to create a new one (pausing the script for each file)
#               Working on Praat 5.3.77 under Windows 7
# AUTHOR:       fcangemi@uni-koeln.de

# INPUT
form Input parameters
        comment path (with final slash)
        sentence directory C:\Documenti\temp\
        comment plot parameters
        sentence figname mausmooth
        boolean raw 1
        boolean manual 1
        boolean smooth 1
        integer plotmin 75
        integer plotmax 400
        comment advanced extraction parameters
        integer smooth1 10
        integer smooth2 10
        integer timestep 0
        integer pitchmin 75
        integer pitchmax 600
        integer maxcandidates 15
        real silenceThr 0.03
        real voicingThr 0.45
        real octave 0.01
        real octavejump 0.35
        real voiceunvoiced 0.14
endform
Erase all
Select inner viewport: 0.5, 2.5, 0.5, 1.5
Solid line
Line width: 1

# open files
Create Strings as file list: "pitchlist", directory$+"*.Pitch"
Create Strings as file list: "list", directory$+"*.wav"
Sort
nfile = Get number of strings
for i to nfile
```

```
        name$ = Get string: i
        basename$ = name$ - ".wav"
        Read from file: directory$+name$

# ELABORATION 1: EXTRACT
# original f0 track
        To Pitch (ac): timestep, pitchmin, maxcandidates, "no", silenceThr,
           ...voicingThr, octave, octavejump, voiceunvoiced, pitchmax

# manual correction
#       determine if there is already a manually corrected .pitch file
        exists = 0
        selectObject: "Strings pitchlist"
        npitch = Get number of strings
        for k to npitch
                pitchname$ = Get string: k
                pitchbasename$ = pitchname$ - ".Pitch"
                if pitchbasename$ = basename$
                        exists = 1
                endif
        endfor
#       if it exists then open, else provide manual correction and save
        if exists = 1
                Read from file: directory$+basename$+".Pitch"
        elsif exists = 0
                selectObject: "Sound "+basename$
                View & Edit
                selectObject: "Pitch "+basename$
                Copy: basename$
                View & Edit
                pause Confirm
                Save as text file: directory$+basename$+".Pitch"
        endif
        Rename: "manual"

# smooth track
        Smooth: smooth1
        Rename: "smooth"
        Interpolate
        Smooth: smooth2
        Save as text file: directory$+basename$+".smooth"
        Down to PitchTier
        Save as headerless spreadsheet file: directory$+basename$+".hesp"

# ELABORATION 2: PLOT
# with any form-defined combination of
# original in red, corrected in silver, smoothed in black.
# Given the order of plotting, red "surfaces" as manually corrected points
        if raw = 1
                selectObject: "Pitch "+basename$
                Red
                Speckle: 0, 0, plotmin, plotmax, "no"
        endif
        if manual = 1
                selectObject: "Pitch manual"
                Silver
                Speckle: 0, 0, plotmin, plotmax, "no"
        endif
        if smooth = 1
                selectObject: "Pitch smooth"
                Black
                Draw: 0, 0, plotmin, plotmax, "no"
        endif

# OUTPUT
# close loops and clean
        select all
        minusObject: "Strings list"
        minusObject: "Strings pitchlist"
        Remove
        selectObject: "Strings list"
endfor
plusObject: "Strings pitchlist"
Remove

# save output
Save as 600-dpi PNG file: directory$+figname$+".png"
```