

Guidelines for Using Mview

Quick Tip

Most Mview functions have corresponding hotkeys assigned to them. These can be seen next to their corresponding items in Mview's menus, and make operating Mview much faster if learned. Extra documentation is also available by typing *help mview* into the MATLAB command window.

Opening the Program

- 1) Open MATLAB.
- 2) Set the current directory to the Mview directory by filling in the text box labeled "Current Directory" on the toolbar of the main MATLAB window.
- 3) Start Mview by entering the command:

```
mview('data', 'dproc', {'dproc1', 'dproc2', ...}, 'lproc', 'lp_findgest');
```

'data' is the name of the desired data set¹ (this can be a list, in which case multiple Mview windows will open), and 'dproc1' and 'dproc2' specify any preprocessing to be performed on the data before starting Mview. The dproc field is optional, so none or any number of data processes can be specified. Available processes are described in the section below called Data Processes. 'lproc' is the labeling procedure used by mview. The two available choices are 'lp_findgest', which produces temporal landmarks for a gesture, and 'lp_extrema', which finds the timestamps of the minimum and maximum spatial point (x, y, or z coordinate) over the currently selected portion of a trajectory and the spatial values at those timestamps. They are described in the sections below called Using lp_findgest and Using lp_extrema. As an example, typing:

```
mview('M_devb*', 'dproc', {'mdp_PosOnly2D'}, 'lproc', 'lp_findgest');
```

would start Mview on all data sets whose name starts with M_devb, and would strip out all but the x and y positional signals in the data for each receiver panel (i.e., lateral movement and orientation signals are excluded from the Karstens 3D datasets). This is what you need to do for most analyses you are likely to do using this data. But if instead of 'mdp_PosOnly2D' you specified 'mdp_PosOnly' this would display three trajectories per receiver panel, x, y, z. Crucially, x is lateral movement, y is anterior-posterior and z is

¹ In order for data to be loaded by Mview, it must be visible to MATLAB. It must either be somewhere on MATLAB's path (the path always includes the current working directory as set in step 2, so if your data is in the same folder as mview then this would make things work), or it must be imported directly into the workspace. To import data, select "Import Data..." from the File menu. Navigate to the desired data file and select it. Finally, click "Finish" at the bottom of the resulting Import Wizard window.

up-down movement. To get rid off the lateral you need to specify 'mdp_PosOnly2D' as was done above.

Trajectories represent sequences of discrete measures, where each data point / frame corresponds to a 5 ms temporal slice. What are the axes in the upper left panel of Mview? The y or up-down axis shows the vertical movement of the receiver. The x or anterior-posterior axis shows the horizontal movement of the receiver. In the Munich Karstens system, advancement in the x-dimension is shown by a dip in the x-position trajectory or by lower values in the x-position, and retraction is shown by a peak or higher values in the x-dimension. This is unlike in the Haskins 2D system where horizontal retraction is shown by smaller (more negative) values. Within each receiver panel, the x trajectory is shown with the bright color, and the y with the faint color.

4) An Mview window should now open, displaying the relevant data. You may reconfigure lp_findgest or lp_extrema when Mview first starts up (a configuration window will pop up), or just click OK if the default configuration is acceptable.

5) Note that you can call Mview without specifying the labeling procedure in the command line. That is, instead of

```
mview('M_devb*', 'dproc', {'mdp_PosOnly2D'}, 'lproc', 'lp_findgest');  
or mview('M_devb*', 'dproc', {'mdp_PosOnly2D'}, 'lproc', 'lp_extrema');
```

you can call the program without committing to any of the two labeling procedures by using this statement

```
mview('M_devb*', 'dproc', {'mdp_PosOnly2D'});
```

Then, when you are ready to label with a particular procedure, go to Labels → Labelling Behavior → Select and choose lp_findgest or lp_extrema, as you wish.

Using lp_findgest

1) If desired, clean up the Mview window by hiding unnecessary trajectories. From the Configure menu select "Temporal Layout...". To hide a trajectory, select it in the "Displayed" column, and click the "x" button. To show a trajectory, select it in the "Loaded" column and click the ">" button. The "^" and "v" buttons allow you to arrange the display order of the trajectories. From temporal layout window, it is also possible to limit a trajectory to a particular axis, by clicking on the checkboxes in the "Components" area. Limiting a trajectory to a particular axis also ensures that the labeling procedure only operates on that axis.

2) Find the desired trajectory to label (only positional trajectories, those whose names end in POS, are valid targets for labeling).

3) Find the gesture to label. For info on what a gesture is, see the section titled Anatomy of a Gesture. It may be helpful to play the sound associated with the data. This is done

with the "Play" item in the MVIEW menu. It may also help to zoom in on a particular part of the signal, the "frame" around the gesture. You can frame a selection by clicking on the left edge of the channel window (the waveform on top of all the trajectories and the audio signal) and moving it with the mouse to the left edge of the area you wish to enclose. Again using the mouse, click on the right edge of the channel window and move the mouse to the right edge of the area you wish to enclose. This process will leave a gray strip of the signal selected. To zoom in on this selected area, use "Update" from the MVIEW menu.

4) Locate the gestural plateau. This is the point of maximum oral tract constriction, and will typically be a peak in the positional signals (and a corresponding minimum in the associated velocity signals if they are visible). To find this plateau, it is useful to move time cursor over the positional signal, while looking at the numeric read-outs at the bottom left corner of the Mview window.

5) Right-click on the positional trajectory at the gestural plateau. This will cause the automatic labeling procedure to generate the following set of labels (temporal landmarks in milliseconds) for the gesture. For more information on these landmarks, see the section titled Anatomy of a Gesture. For information on how the labeling procedure works, see the section titled Labeling Procedure:

GONS - the time point of the gestural onset

PVEL_to - the time point of maximum articulator velocity before the plateau

NONS - the time point of the onset of the gestural nucleus

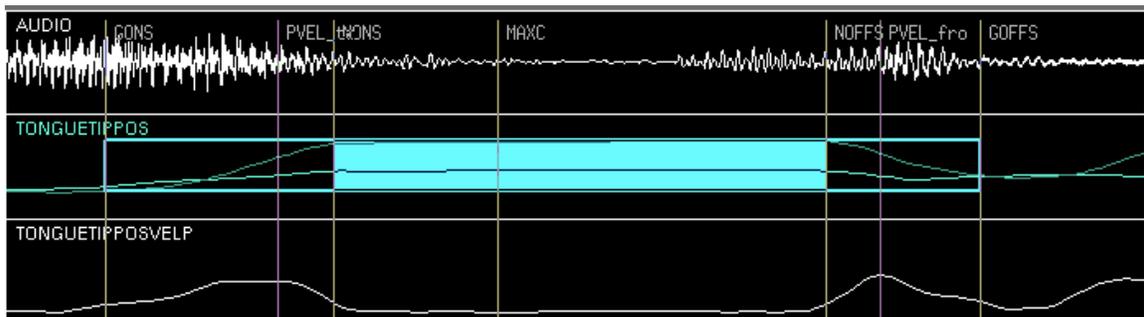
MAXC - the time point of the maximum constriction

NOFFS - the time point of the end or offset of the gestural nucleus

PVEL_fro - the time point of maximum articulator velocity after the plateau

GOFFS - the time point of gestural offset

In the following screenshot, for example, the TONGUETIPPOS trajectory has been labeled according to its tangent velocity (also pictured). *PVEL_to* and *PVEL_fro* are marked by purple lines while the remaining landmarks are marked by yellow lines:



6) Once a set of labels is automatically generated, it can be manually edited by selecting "Edit Labels..." from the Labels menu. To edit, select a label from the list and click the Edit button. To remove a set of labels, select "Clear All Labels" from the Labels menu.

Using lp_extrema

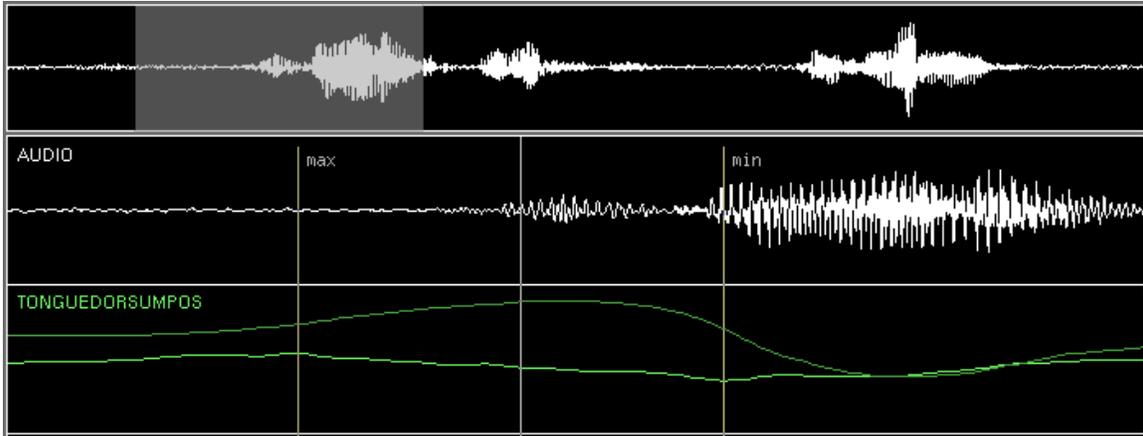
See “Opening the Program” section above on how you can call lp_extrema.

- 1) Select which axis the labeling procedure will operate on, either by clicking on the appropriate checkbox (x, y, or z) in the lp_extrema configuration dialog that pops up when Mview first opens, or by manually opening the configuration dialog by going to "Labeling Behavior>Configure..." from the Labels menu.
- 2) If desired, clean up the Mview window by hiding unnecessary trajectories using the “Temporal Layout...” configuration menu. From temporal layout window, it is also possible to limit a trajectory to a particular axis, by clicking on the checkboxes in the "Components" area. However, unlike for lp_findgest, visually limiting a trajectory to a particular axis does not affect which axis lp_extrema operates on (only the configuration dialog described in 1 above does that).
- 3) Find the desired trajectory to label (only positional trajectories, those whose names end in POS, are valid targets for labeling).
- 4) Frame a selection on the trajectory that contains the desired minimum and maximum spatial values. Again, to frame a selection, click on the left edge of the channel window (the waveform on top of all the trajectories and the audio signal) and move it with the mouse to the left edge of the area you wish to enclose. Again using the mouse, click on the right edge of the channel window and move the mouse to the right edge of the area you wish to enclose. This process will leave a gray strip of the signal selected.
- 5) Right-click anywhere on the positional trajectory. This will cause the automatic labeling procedure to generate labels and display their locus on the trajectory:

min - offset in ms of minimum value when it occurs in the data

max - offset in ms of maximum value when it occurs in the data

In the following screenshot, for example, the TONGUEDORSUMPOS trajectory has been labeled on the x-axis (the bright green signal). Note that the trajectory has been zoomed so that only the gray framed selection is visible:



6) Once a set of labels is automatically generated, it can be manually edited by selecting "Edit Labels..." from the Labels menu. To edit, select a label from the list and click the Edit button. To remove a set of labels, select "Clear All Labels" from the Labels menu.

Exporting Labels

Select "Export Labels..." from the Labels drop-down menu. Designate a file to store the labels in. By default this file should have a .lab extension. You can repeat the export operation to send more labels to the same file. If you do multiple exports to the same file, mview will tell you that the file you are attempting to export your labels to already exists and will ask you if you want to replace it. Say yes – you are not going to lose the info already in the file, since mview simply appends the new label information to the old one.

If using `lp_findgest`, exporting labels appends the information shown below to the seven landmarks located by the labeling procedure discussed in 5) of "Using `lp_findgest`". Note the difference between *PVEL_to* and *PV_to*. They are very different: the former refers to the time point of peak velocity in ms (milliseconds), and the latter refers to speed in cm/sec.

ONV - velocity at GONS (in cm/sec)

NONV - velocity at NONS (in cm/sec)

MCV - velocity at MAXC (in cm/sec)

NOFFV - velocity at NOFFS (in cm/sec)

OFFV - velocity at GOFFS (in cm/sec)

PV_to - velocity at PVEL_to (in cm/sec)

PV_fro - velocity at PVEL_fro (in cm/sec)

AMP_to - amplitude of closing phase, distance between GONS and NONS positions

AMP_fro - amplitude of opening phase, distance between NOFFS and GOFFS positions

PD - peak displacement (euclidean distance between GONS and MAXC positions)

For more information see the section titled Anatomy of a Gesture.

If using *lp_extrema*, exporting a label set provides the following information:

MIN - the minimum value in mm on the desired axis within the framed selection

MIN_OFFSET - offset in ms of minimum value when it occurs in the data

MAX - the maximum value in mm on the desired axis within the framed selection

MIN_OFFSET - offset in ms of maximum value when it occurs in the data

HEAD - left edge in ms of framed selection on which labeling was performed

TAIL - right edge in ms of framed selection on which labeling was performed

Converting Data to Excel Format for Statistical Analysis

1) Open Excel. From Excel, open the output file you designated. Be sure to set the *Files of type* option to *All Files* so that your file appears in the list of options.

2) The data must be converted to a format that is Excel-like. Inside the *Original data type* box, select the delimited option. Select *Next*. Inside the *Delimiters* box, select *Tab* and *Space*. Select *Finish*.

3) Line up the words with their values by inserting cells inside Excel accordingly. Add a row with the names of the labeled values.

4) Save the Excel file using an Excel ending.

Data Processes

mdp_PosOnly - strip all but positional signals from data

mdp_PosOnly2D - strip all but positional signals from data, and remove z-component if present

mdp_Accels - add acceleration magnitude signals for each positional signal

mdp_Accelsx - add x-axis acceleration signal for each positional signal

mdp_Accelsy - add y-axis acceleration signal for each positional signal

mdp_Velocities - add velocity magnitude signal for each positional signal

mdp_Velocitiesx - add x-axis velocity signal for each positional signal

mdp_Velocitiesy - add y-axis velocity signal for each positional signal

mdp_LipAperture - add lip aperture signal (distance between upper and lower lip)

Anatomy of a Gesture

A gesture can be defined as “a member of a family of functionally equivalent articulatory movement patterns that are actively controlled with reference to a given speech-relevant goal (e.g. bilabial closure)” Saltzman & Munhall 1989, p. 334. Defined in this way, gestures involve movement. Movements are parts of gestures or gestural components. But not all movements are gestural components. The movement of the Tongue Tip during the beginning of words like *tip*, *dip*, and *nip*, for example, would be a component of that gesture. The movement of the Lower Lip that also occurs during Tongue Tip raising, however, is not a part of the gesture. That movement is a passive consequence of raising the Tongue Tip. Conversely, when the Lower Lip moves as a consequence of the bilabial closure in the beginning of words like *path*, *bath*, and *math*, the lip movement is a gestural component.

For each gesture, there is a movement component towards the target (Tongue Tip constriction in the example above), and a movement component away from the target. These are referred to as the ‘closing’ or ‘to’ phase and ‘opening’ or ‘fro’ phase, respectively. These phases can be further punctuated by temporal landmarks and associated kinematic parameters.

The to phase begins at a point called the gestural onset (GONS), when the articulator is first mobilized and begins to accelerate. The articulator speeds up until it begins to slow down in preparation for reaching its target. This change in acceleration is the point of maximum velocity during the to phase (referred to as PVEL_{to} above). The to phase continues until the target is just about reached, a point called the onset of the gestural nucleus (NONS). The gestural nucleus itself is the interval of time the articulator spends at the target position. It contains the time point when the vocal tract is maximally constricted (MAXC). The gestural nucleus is also known as the hold phase in articulatory phonetics.

The three basic kinematic parameters of the to phase are the peak velocity, the amplitude or distance traveled, and the duration of movement. First, the maximum velocity during the phase, measured at the change in acceleration described above (PV_{to}). Second, the amplitude of the closing movement, *AMP_{to}*, measured as the distance the articulator travels from the gestural onset to the nucleus onset (*AMP_{to}*). Third, the duration of the to phase is the time elapsed between the gestural and nucleus onsets. This latter one is the only parameter that is not directly reported in the output of *lp_findgest*. But it is trivially computable by subtracting two of the automatically reported landmarks.

The fro phase is symmetric in structure to the to phase. It begins at the point called the nucleus offset (NOFFS), when the articulator leaves its target position or begins to release away from its hold phase. The articulator speeds up until it begins to slow down in preparation for reaching a rest position. This change in acceleration is the point of

maximum velocity during the fro phase (PVEL_fro). Opening continues until the articulator has reached a rest position, a point known as the gestural offset (GOFFS).

As with the to phase, the three basic kinematic parameters of the fro phase are: the maximum velocity during the phase (PV_fro), the amplitude of the opening movement (AMP_fro) measured as the distance the articulator travels from the nucleus offset to the gestural offset, and the duration of the fro phase (time elapsed between the nucleus and gestural offsets).

Finally, the duration of a gesture as a whole is the time between the gestural onset and the gestural offset.

lp findgest Procedure

The gestural identification algorithm proceeds as follows:

1) Find the nearest velocity minimum to the point where the user clicked on the positional trajectory being labeled. More precisely, find all the velocity minima over the entire trajectory (points where the first derivative is 0 and second derivative is positive), and select the one that is the fewest milliseconds away from the point where the user clicked. This corresponds to MAXC. The velocity used could be either the x, y, or tangential velocity (the magnitude of the $\langle x, y \rangle$ velocity vector) trajectory associated with the positional trajectory of the articulator in question. Which of the three is used depends on which axes of the positional trajectory are currently visible (see section on labeling above for info on display options). The velocity trajectories used could be precalculated and included with the Mview data set being looked at. If no precalculated velocity trajectory is present in the data², Mview computes a velocity trajectory by differentiating the positional signal of the articulator.

2) Find the velocity peaks (maximums) before and after MAXC. These correspond to PVEL_to and PVEL_fro, respectively.

3) Find the gestural and nucleus onsets and offsets. The gestural onset (GONS) is the point before PVEL_to when the velocity drops below a certain small percentage of PVEL_to (this percentage can be set by going to "Labeling Behavior>Configure..." from the Labels menu, or in the configuration window Mview pops up at startup). The nucleus onset (NONS) is the point after PVEL_to when the velocity drops below the threshold percentage of PVEL_to. The nucleus and gestural offsets (NOFFS and GOFFS) are found the same way, using the same velocity threshold value, only the threshold applies to PVEL_fro instead.

When labeling, Mview attempts to ignore insignificant fluctuations or bumps in velocity (of the receiver attached on some articulator) resulting from noise or independent vocal

² Currently, precalculated velocities are only available for subject OB's data (from MCA3 2006 datasets).

tract activity (e.g., movement of a different articulator). The parameters which control this behavior are in the following section of the code file `DelimitGest.m`:

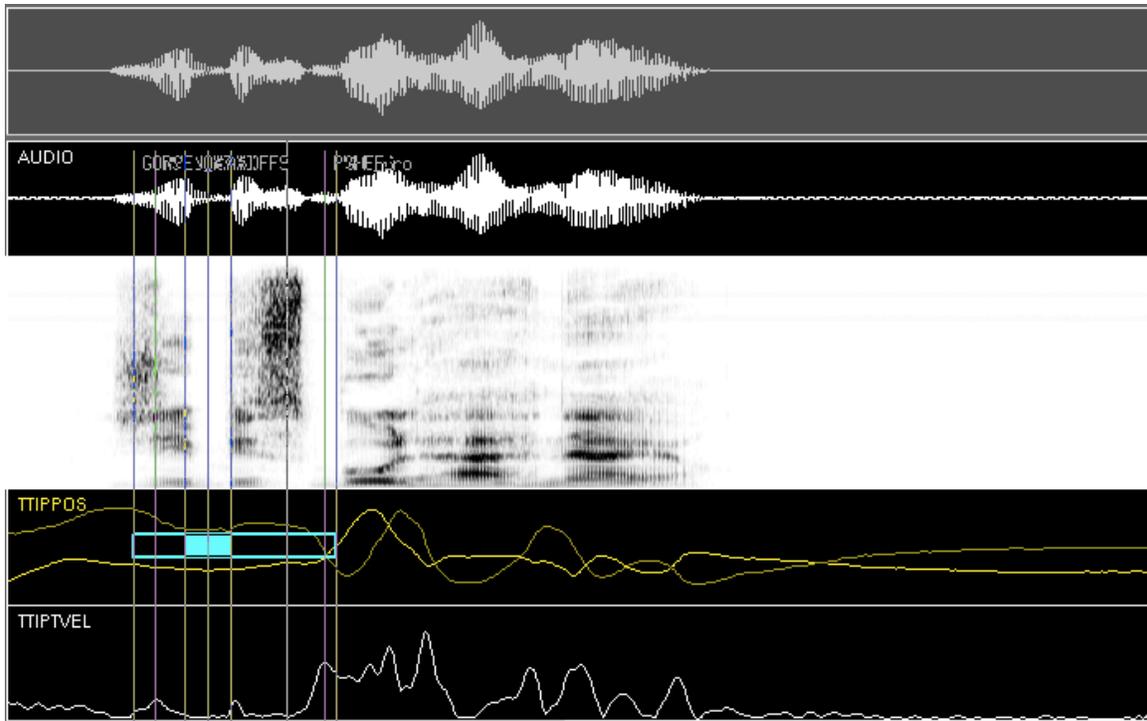
```
% defaults  
thresh = .2;  
onsThr = .2;  
offsThr = .15;
```

`thresh` is the threshold percentage used to find gestural and nucleus onsets and offsets as described above (e.g. GONS is the point when the velocity drops below 20% of `PVEL_to`). `onsThr` and `offsThr` determine what percentage of the maximum velocity over the currently framed selection of a file (framing a selection is described above in part 3 of the section “Using `lp_findgest`”) a peak needs to be in order to be labeled `PVEL_to` or `PVEL_fro`³.

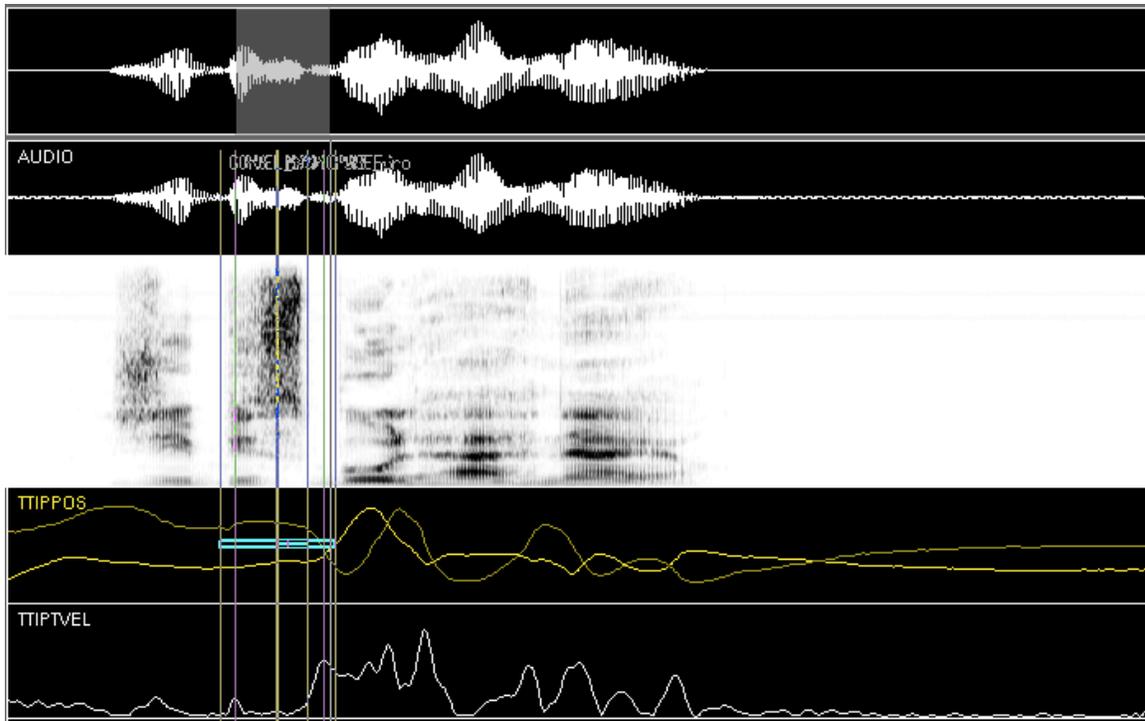
We now turn to discuss cases where the gestural identification procedure above gives the wrong results. When the user clicks near the plateau of a gesture (a velocity minimum) `Mview` tries to find a velocity peak before and after that minimum. Most of the time, the default thresholds produce correct results. However, when searching for the two peaks, `Mview` might skip over the real ones because they are too small, and accidentally find the peaks of neighboring gestures instead. To fix such behavior, the `onsThr` and `offsThr` can be lowered as needed, or, preferably, the gesture can be framed more precisely.

As an example, we can take a look at a token from subject CZ containing a /s/ gesture (the first consonant of [sbulha] in the dataset `CC_1_2_0_sbulha_10_0057.mat`). When we attempt to label the /s/ gesture (the noisy period on the spectrogram about a third of the way through the utterance) with the entire file selected, we get inaccurate gestural landmarks that contain much of the material before the /s/ gesture. When compared to the maximum velocity over the entire file (in this case 24.1cm/sec), the real `PVEL_to` of the /s/ is not large enough (it falls below the threshold `onsThr`) and `Mview` treats it as a noisy fluctuation.

3 If there is no framed selection, potential `PVEL_to` and `PVEL_fro` labels are compared to the maximum velocity of the entire file. This “global” labeling used to be the `Mview` default, and was used in the past by Mark Tiede for his “perfect memory” project, and in *Gafos, Adamantios, Philip Hoole, Kevin Roon, and Chakir Zeroual. In press. Variation in timing and phonological grammar in Moroccan Arabic clusters. Laboratory Phonology 10, Paris, 29 June 2006.*



However, when the /s/ gesture is more accurately framed, the gestural landmarks found by Mview are much more accurate. This time, the real PVEL_to of the /s/ gesture is large enough compared to the maximum velocity in the selection frame (in this case, 17.0cm/sec) to be considered significant.



Displaying the palate trace

For subject CZ, the data for the palate trace are in *cz_palate_refhead.mat* from the CZ 2004 experiment (mca1) and *mca3_cz_palatetrace.mat* from the CZ 2006 experiment (mca3). To display CZ's palate trace from the 2004 experiment (mca1), first load the palate data in Matlab's Workspace by going to File – Open ... and choosing *cz_palate_refhead.mat*. This will load a data structure called *data* in Matlab's Workspace. The command *whos data* in the Matlab command window will tell us that this structure is a 141x3 array – it has 141 rows and 3 columns of numbers – which you can see if you give the command *data*. Without going into much detail, these numbers correspond to positions in the vocal tract and they are in cm, so we need multiply by 10 to get mm. We have also cut off column 1 of this data structure by using the expression *data(:,[2 3])* because this column corresponds to the lateral dimension.

```
mview('F_ft1_104', 'palate', data(:,[2 3])*10, 'dproc', 'mdp_PosOnly2D', 'lproc', 'lp_findgest');
```

To display CZ's palate trace from the 2006 experiment, load the palate data in matlab's workspace (as done above) and then pass the data as a parameter to *mview* as shown below.

```
mview('CC*', 'palate', data(:,[2 3]), 'dproc', 'mdp_PosOnly', 'lproc', 'lp_findgest');
```

Note that *mdp_PosOnly2D* does not work as a dproc for the datasets from the 2006 experiments. This is because the lateral movement has been pruned from the Matlab datafiles.

Selecting specific dimensions of data signals for analysis

MVIEW provides the option of displaying positional signals in 3 dimensions: x (horizontal), y (vertical), and z (lateral). Under certain conditions, however, it may be desirable to restrict analysis to a subset of these 3 dimensions. The *lp_findgest* function delineates gestures based on the velocity profile associated with the displayed spatial signal. Because of this, the parse that *lp_findgest* returns for a given token may crucially depend on the spatial dimensions displayed. If all three dimensions are displayed, then motion in any dimension will contribute to the magnitude of the composite velocity signal.

In the absence of a particular theoretical or practical motivation, the most conservative approach to parsing gestures is to incorporate all motion in 3-D space. Linguists typically operate, however, under specific assumptions about the dimensions which define particular action events. Phonological contrasts, for example, are typically defined in terms of the constriction location and constriction degree within the mid-sagittal plane. On this assumption, lateral movement in the z-dimension is orthogonal to the action events of interest. Analysts may therefore choose to isolate the mid-sagittal plane by quantifying movement exclusively along the x- (horizontal) and y- (vertical) dimensions. This effectively factors out the noise that non-linguistic movement in the lateral dimension may introduce into the velocity profile. Similarly, there may be theoretical reasons to restrict a particular analysis to the vertical or horizontal dimensions.

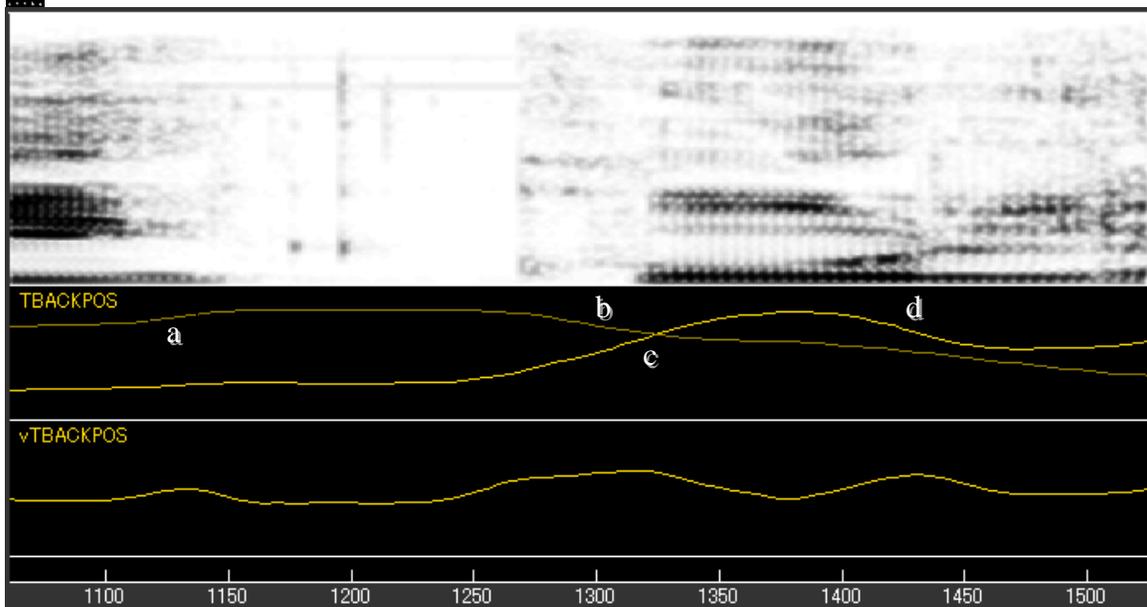
In addition to theoretical motivations, there are also cases in which limiting analysis to single dimensions is necessary for practical reasons. The example below discusses the sequence [kul] which involves temporally overlapping gestures that share an articulator (as in MB's productions of **kulha_04_0077**, **kulha_05_0103**, **kulha_06_0140**, **skulha_04_0080**, **skulha_05_0110**, **skulha_06_0137**, **meskulhaC_04_0073**, **meskulhaC_05_0105**, **meskulhaC_06_0138**). Since *lp_findgest* uses velocity peaks to identify gestures, those velocity peaks which incorporate contributions from multiple action events are potentially problematic. The danger is that *lp_findgest* uses a peak that is pushed in one direction or another by temporally overlapping gestures. Sometimes it is possible to disentangle the parse of overlapping gestures by using different velocity signals for each gesture.

In producing the sequence [kul], the TONGUE BACK first rises to form a constriction against the soft palate required for [k]; it then lowers somewhat and retracts to a posterior target for the back vowel [u] before moving forward again to form the [l] TONGUE TIP constriction at the alveolar ridge. TONGUE BACK motion in the production of [kul] can therefore be decomposed into 4 distinct events:

- (a) vertical movement toward the hard palate (movement toward [k] closure)
- (b) vertical movement away from the hard palate (movement away from [k] closure & movement toward [u])
- (c) horizontal movement toward the vowel target (movement toward [u])
- (d) horizontal movement away from the vowel target (movement away from [u] & movement toward [l])

Figure 1 shows the motion of the tongue back during the production of [kulha] from a recording session of subject MB, datafile: kulha_05_0103, found in the folder MBABdatafiles which is part of the mview software packet. Each of the distinct action events listed above are labeled in the Figure (bold is x-, faint is y-). In addition to spatial information, Figure 1 also shows the tangential velocity associated with the movement. This measure of velocity incorporates motion in both the x- and y- dimensions. Inspection of the tangential velocity signal reveals that the four distinct articulatory events correspond to just three distinct velocity peaks.

Figure 1: (from subject MB, datafile kulha_05_0103) Acoustic signal (top panel); Positional signal for the tongue back in both x- and y- dimensions (middle panel); Tangential velocity signal for the tongue back (bottom panel).



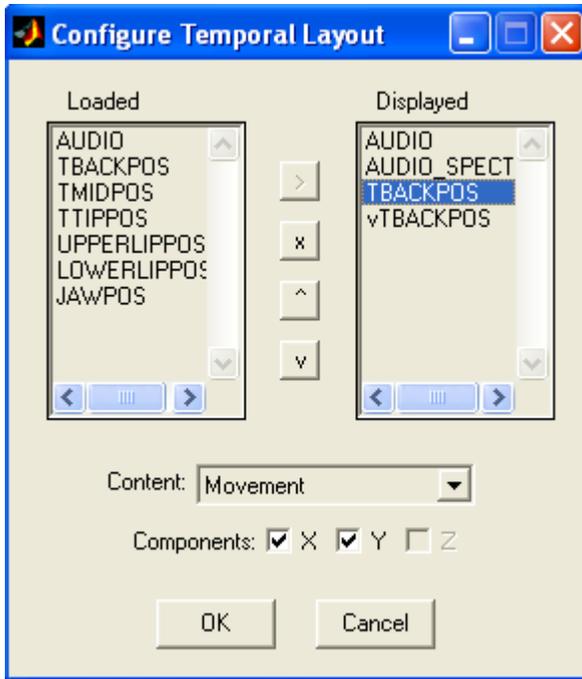
The first velocity peak corresponds primarily to motion in the y- dimension towards the soft palate, (a), for the [k] constriction. The third velocity peak corresponds primarily to backward movement in the x- dimension for the [u]. The second velocity peak, however, corresponds to both the vertical motion of the tongue back lowering, (b), and the horizontal motion associated with the retraction of the tongue body, (c). As Figure 1 shows, action events (b) and (c) overlap in time. Both events imply higher velocities in the x-, for (c), and y-, for (b), components of the positional signal. Therefore, each event contributes to a single broad tangential velocity peak spanning the 1275-1325 ms window.

Let us consider now how `mview` performs when asked to parse out the gestures for [k] and [u]. Direct matlab to the location containing MB's data. Then, type the following command in the matlab command window:

```
mview('kulha_05_0103','dproc',{'mdp_PosOnly2D'},'lproc','lp_findgest')
```

This will open the token displayed in Figure 1 (MB kulha_05_0103). Isolate the positional and velocity signals of the TONGUE BACK by selecting 'temporal layout' from the 'configure' menu within `mview`. This will open a window, shown in Figure 2, which will allow you to configure the signals displayed by `mview`. The default `mview` setting displays position signals for each receiver. For the purposes of this example we are only interested in the TONGUE BACK. Other positional signals may be removed from the display by highlighting the receiver label (e.g. TMIDPOS for the tongue mid positional receiver) and clicking the 'x' button. Remove all positional signals except the TONGUE BACK (labeled TBACKPOS). Next add the TONGUE BACK velocity signal. To do this, highlight TBACKPOS in the 'Loaded' box in the left side of the window and click on the left-pointing arrow, '>'. This will display two panels containing the TONGUE BACK signals in `mview`. To make one of these panels display the velocity signal, highlight TBACKPOS from 'DISPLAYED' box in the right side of the window and select 'velocity' from the pull-down menu labeled 'content'. If your 'Configure Temporal Layout' window matches that of Figure 2, then `mview` will display the positional and velocity signals of the TONGUE BACK as shown in Figure 1.

Figure 2: Configuration of temporal layout window which instructs `mview` to display the positional and velocity signal associated with the tongue back with the acoustic waveform and spectrogram.



After configuring *mview*'s temporal layout, zoom in on the portion of the file corresponding to [kul] and use *lp_findgest* to locate the [k] and [u] gestures. Although the lowering of the tongue back for [k] and the retraction of the tongue back for [u] are distinct articulatory events, *mview lp_findgest* uses the same tangential velocity peak to delineate both [k] and [u]. This yields an artificial correspondence between the landmarks of the two gestures. This correspondence is demonstrated in Table 1. The GONS landmark of [u] is synchronous with the NOFFS landmark of [k], and the GOFFS of [k] is synchronous with the NONS landmark of [u]. This is a consequence of defining different gestures with the same velocity peak.

Table 1: (MB's kulha_05_0103) Lumped landmarks in the [k] and [u] of [ku] due to gesture parse based on a shared tangential velocity peak.

	<u>NOFFS(ms)</u>	<u>PVEL_fro(ms)</u>	<u>GOFFS(ms)</u>
[k]:	1245	1315	1360
	<u>GONS(ms)</u>	<u>PVEL_to(ms)</u>	<u>NONS(ms)</u>
[u]:	1245	1315	1360

The lumping problem can be avoided in this case by directing *lp_findgest* to the individual dimensions of movement and their corresponding component velocity signals. To see how this can be done, let us consider the data by first looking at individual dimensions. To do this, reopen the 'Configure Temporal Layout' window. The *mview* command given above loads only the x- and y- components of the positional signal. *Mview*'s default setting displays each of the loaded component signals. To deselect one of the loaded components, untick the box associated with that component. To select just the positional signal in the y- dimension, highlight TBACKPOS in the right-hand side of the window and untick the box associated components boxed labeled 'x'. Do the same with

the `vTBACKPOS` component to isolate the y- component of the velocity signal. This will yield the display shown below in Figure 3.

The problem with the second velocity peak in Figure 1 is that it is defined by motion in two dimensions from two different gestures. Figures 3 and 4 isolate the y- and x- dimensions, respectively. In Figure 3, we can see that action events (a) and (b), movement to and away from [k] closure, each correspond to unique velocity peaks in the y- dimension. In Figure 4, we can see that events (c) and (d), movement to and away from [u], correspond to unique velocity peaks in the x- dimension (Figure 4).

Figure 3: (MB's kulha_05_0103) Positional signal for tongue back movement in the y- dimension (middle panel); Component velocity signal for tongue back movement in the y- dimension (bottom panel).

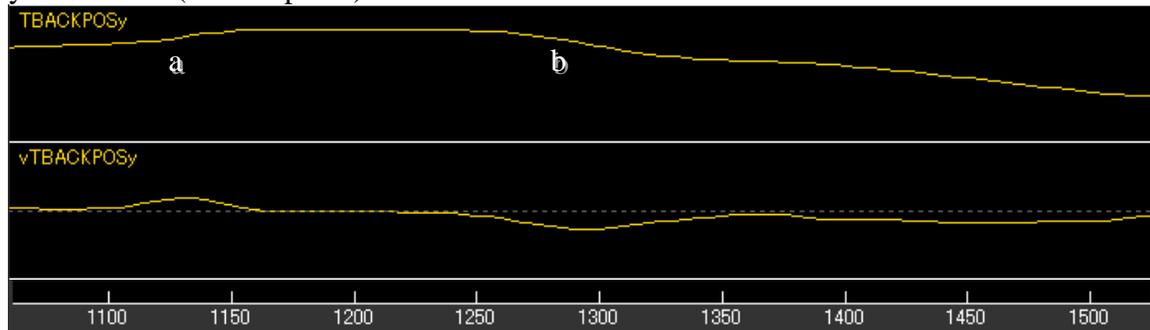
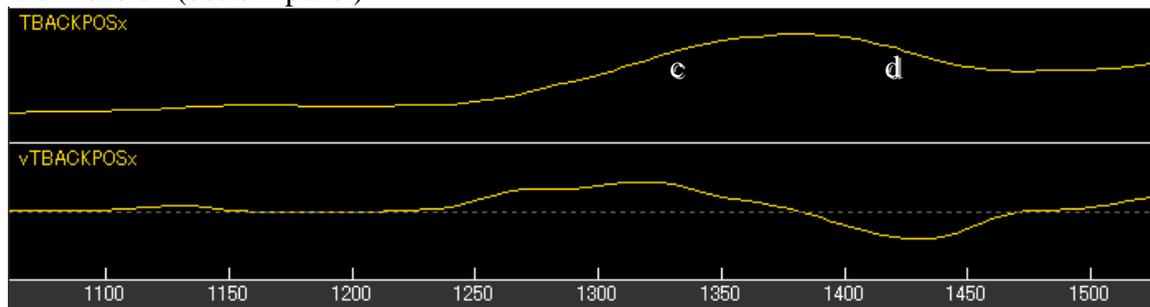


Figure 4: (MB's kulha_05_0103) Positional signal for tongue back movement in the x- dimension (middle panel); Component velocity signal for tongue back movement in the x- dimension (bottom panel).



By decomposing movement into separate x- and y- components, we can guide `lp_findgest` to identify separate velocity peaks for (b) and (c). This is demonstrated in Table 2. Now, the peak velocity point of the vowel [u] follows the peak velocity point of the consonantal release [k] by 25 ms. Thus, although the [k], [u] gestures are temporally overlapping, we can tease apart distinct movements corresponding to the movement away from target for the [k] and movement toward target for the [u] by referring to different component velocities.

Table 2: (MB's kulha_05_0103) landmarks in the [k] and [u] of [ku] now teased apart due to gesture parses based on a different component velocities.

	<u>NOFFS(ms)</u>	<u>PVEL_fro(ms)</u>	<u>GOFFS(ms)</u>
[k] y-dimension	1250	1295	1345
	<u>GONS(ms)</u>	<u>PVEL_to(ms)</u>	<u>NONS(ms)</u>
[u] x-dimension	1245	1320	1365

As an exercise, the reader can attempt to determine whether the velocity peaks in [ku] sequences are merged or not in the following datafiles. These datafiles are all from subject AB and should be found in the folder MBABdatafiles, part of the mview software packet.

kulha_06
kulha_07
kulha_09
kulha_08
skulha_06
skulha_09
skulha_08
skulha_10
meskulha_06
meskulha_07
meskulha_08
meskulha_09

References

- Gafos, A., Hoole, P., Roon, K. and Zeroual, C. In press. Variation in timing and phonological grammar in Moroccan Arabic clusters. *Laboratory Phonology 10: Variation, Detail and Representation*, Ed. Cécile Fougeron (Mouton de Gruyter: Berlin/New York).
- Saltzman, E. L., & Munhall, K. G. (1989). A dynamical approach to gestural patterning in speech production. *Ecological Psychology*, 1, 333–382.