User Manual for Ground Truthing Tool

1. Overview

Hand labeling the whistles in the large recordings is a near-impossible task as it consumes hours of tedious work. Due to the impracticality of manual transcription we created a graphical user interface tool (GUI) tool for tracing tonals. The user can either mark all tonals manually or use the automated detections as a starting point. The tool takes audio file (*.wav) as an input and display's the spectrogram. The analyst can then trace tonals over the spectrogram using different controls displayed alongside the spectrogram.

When starting with the automatically detected tonals to generate ground tonals, the tool takes the audio file and detected tonals as input. It allows an analyst to add, merge, trim, delete and smooth the tonals to generate ground tonals. Edit operation that is performed is displayed on the screen so that user has immediate feedback. User can discard the last known operation by using undo operation. Various edit boxes are present to change the default settings. Tool provides easy way for user to move to different sections of recording by using start, previous, next and end buttons. Furthermore, user can turn the threshold on/off for spectrogram plotting. Once ground truthing is done user can save the ground truthed tonals for later reference.

This document can also be read to learn how to use the automatic tonal detector (*Silbido*) on its own. Advanced topics section of this document describes the process of getting time and frequency of individual tonal and plotting them in Matlab.

2. Requirements

Matlab 2008b or later with signal processing and image processing toolboxes.

3. Setup

Copy the Triton folder (directory) to your machine and start Matlab. Set Matlab's path (File→Set Path…) to include the Triton folder. If the path is saved, this only need be done once.

While the tools are not integrated into Triton yet, they rely on functionality provided by Triton subroutines. Consequently, Triton must be initialized before using *Silbido* or the ground truth annotation tool. This need be done only once per Matlab session. To initialize triton, type:

```
triton; close all
```

4. Sample Run

4.1 Manual Detector

Trace tonals from scratch within a file.

```
>> dtPlotUIGroundtruth ({'filename.wav'}, [], Start_s, Stop_s,
   'Framing', [2 8], 'Nosie', 'median');
```

Example:

```
>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', [], 0, Inf,
   'Framing', [2 8], 'Nosie', 'median');

>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', [], 50, 54,
   'Framing', [2 8], 'Nosie', 'median');

>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', [], 0, Inf);
```

- More help on ground truth tool can be obtained by running *help dtPlotUIGroundtruth* command on Matlab command.
- If Start_s and Stop_s are set to 0 and Inf, then whole file can be traversed. User can also work on portion of the recording by specifying appropriate start and stop time in seconds as shown in one of the example above.
- If 'Framing' is not specified in the command, then the default values of [2 8] are used which indicate that each successive frame is 2 ms apart and 8 ms in duration. 'Framing' values may be changed depending on what works best for the file.
- If type of noise filter is not specified then the "median' filter is used. 'Noise' is a filter set at either 'meansub', 'median', 'MA' or 'wiener'. Using 'median' is best. Median filtering is a combination of running a median filter followed by spectral means subtraction over a 3 s fixed window. When noise is very variable, the moving average (MA) process may be effective.
  ***NOTE: If using the detected tonals for generating ground tonals it is recommended that the same noise compensation as used for creating the tonal set be used. Generating ground tonals from detected tonals is described in automatic detection section.

4.2 Using *Silbido* to provide a starting point for annotations

STEP 1. Initiate the detector to trace tonals

```
>> [tonals graph] = dtTonalsTracking ({'filename.wav'}, Start_s,
   Stop_s, 'Framing', [2 8], 'Noise', 'median', 'DetectorRange',
   [low_freq_hz high_freq_hz]);
```

Example:

```
>> [tonals graph] = dtTonalsTracking ({'palmyra092007FS192-
   071011.wav'}, 0, Inf, 'Framing', [2 8], 'Noise', 'median',
   'DetectorRange', [5000 50000]);

>> [tonals graph] = dtTonalsTracking ({'palmyra092007FS192-
071011.wav'}, 0, Inf);
```

- More help on the dtTonalsTracking can be obtained by running *help dtTonalsTracking* command on Matlab command prompt.
- If 'Framing' is not specified in the command, then the default values of [2 8] are used which indicate that each successive frame is 2 ms apart and 8 ms in duration. 'Framing' values may be changed depending on what works best for the file.

- 'Noise' is a filter set at either 'meansub', 'median', 'MA', or 'wiener'. Using 'median' typically performs best. Median filtering is a combination of running a median filter followed by spectral means subtraction over a 3 s fixed window. When noise is very variable, the moving average (MA) process may be effective. If the type of noise filter is not specified then the "median' filter is used.
- 'DetectorRange' can be set to any range of frequencies. Useful when avoiding certain anomalies.

STEP 2. Use the detected tonals from above step as input to the ground truth tool.
***NOTE: Use the detected tonals (**tonals**) instead of empty brackets [].

```
>> dtPlotUIGroundtruth ('filename.wav', tonals, Start_s, Stop_s,
   'Framing', [2 8], 'Noise',  'median');
```

Example:

```
>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', tonals, 0,
   Inf,  'Framing', [2 8], 'Noise', 'median');

>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', tonals, 0,
   Inf);
```

***NOTE: More information on dtPlotUIGroundtruth tool is described in manual detector section.

4.3 SAVE the file repeatedly throughout the session.

4.4 Retrieve saved files

The current file can be saved in the middle, closed, and later retrieved to finish tracing tonals.

STEP 1. Load the saved tonals

```
>> [traced_tonals] = dtTonalsLoad('filename.bin');
```

Example:

```
>> [traced_tonals] = dtTonalsLoad('palmyra092007FS192-071011.bin');

>> [traced_tonals] = dtTonalsLoad([], true); % Prompt user for filename
```

STEP 2. Run the ground truth tool with tonals loaded in STEP 1 as input.

```
>> dtPlotUIGroundtruth ('palmyra092007FS192-071011.wav', traced_tonals,
   0, Inf, 'Framing', [2 8]);
```

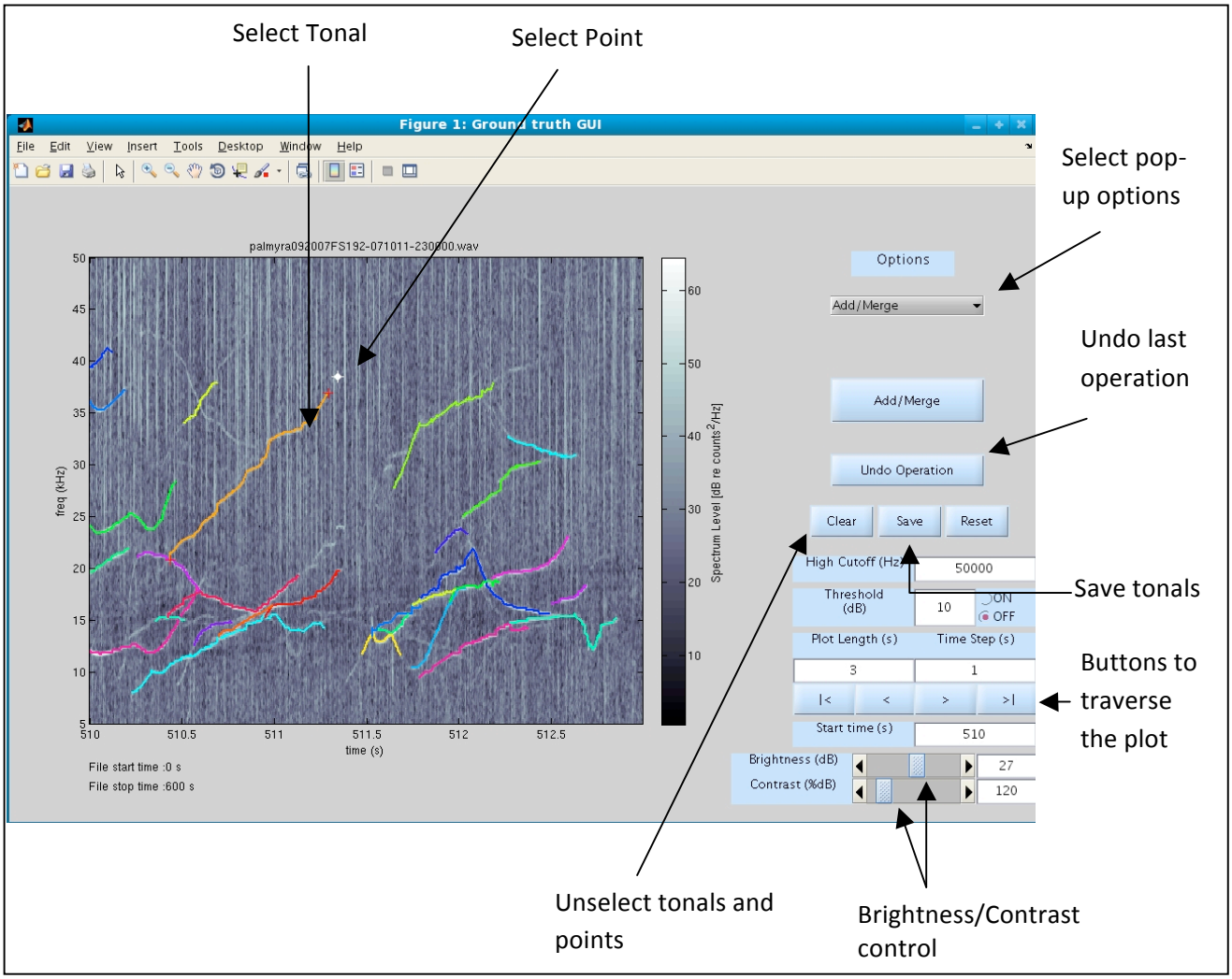
5. Description

Plot: Left side of the figure is a plot with detected tonals plotted over the spectrogram. Clicking on tonal selects the tonal and clicking on the plot area selects a point.

Pop-up: This widget has five different options to choose from: Add/Merge, Delete Tonals, Trim Tonal, Smooth and Spline Fit. Each option is described later in the manual.

Buttons: Different buttons has different functionality. Their functionality is described later in the manual.

Edit boxes: Default values can be edited with the help of edit box. For example, default values for plot length and time step can be modified using corresponding edit box.



5.1 Options

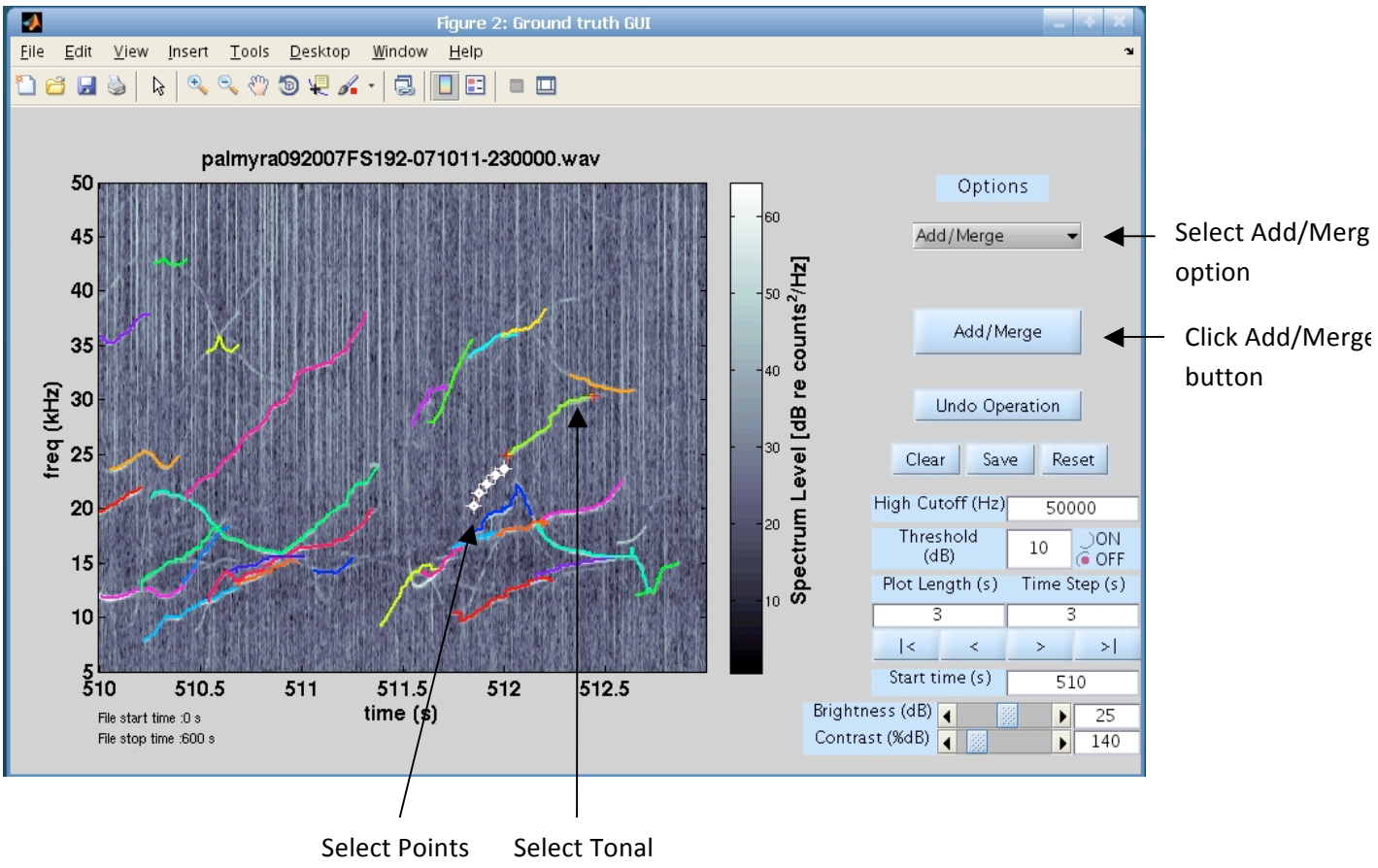
Add/Merge: Selecting *Add/Merge* option from pop-up menu enables user to extend the existing tonals by adding points or by merging tonals or by both adding points and merging tonals. Clicking on a plot selects a point and display's it. Point has drag gable feature so that user can move the existing point to appropriate place on the plot. Multiple points can be selected on the plot. Points can be unselected by "SHIFT-CLICK". Tonals are selected by clicking on tonals that are plotted. Like points, multiple tonals can be selected.

To create a new tonal from scratch user must select points on plot and click *Add/Merge* button. *Add/Merge* button appears after the *Add/Merge* option is selected from pop-up menu.

To extend a tonal user must select a tonal and point, and later click *Add/Merge* button. There are some pre-defined rules for adding points to tonals like points can be added only at the endpoints of tonals. If the selection doesn't follow the pre-defined rules of *Add/Merge* option error dialog box is displayed.

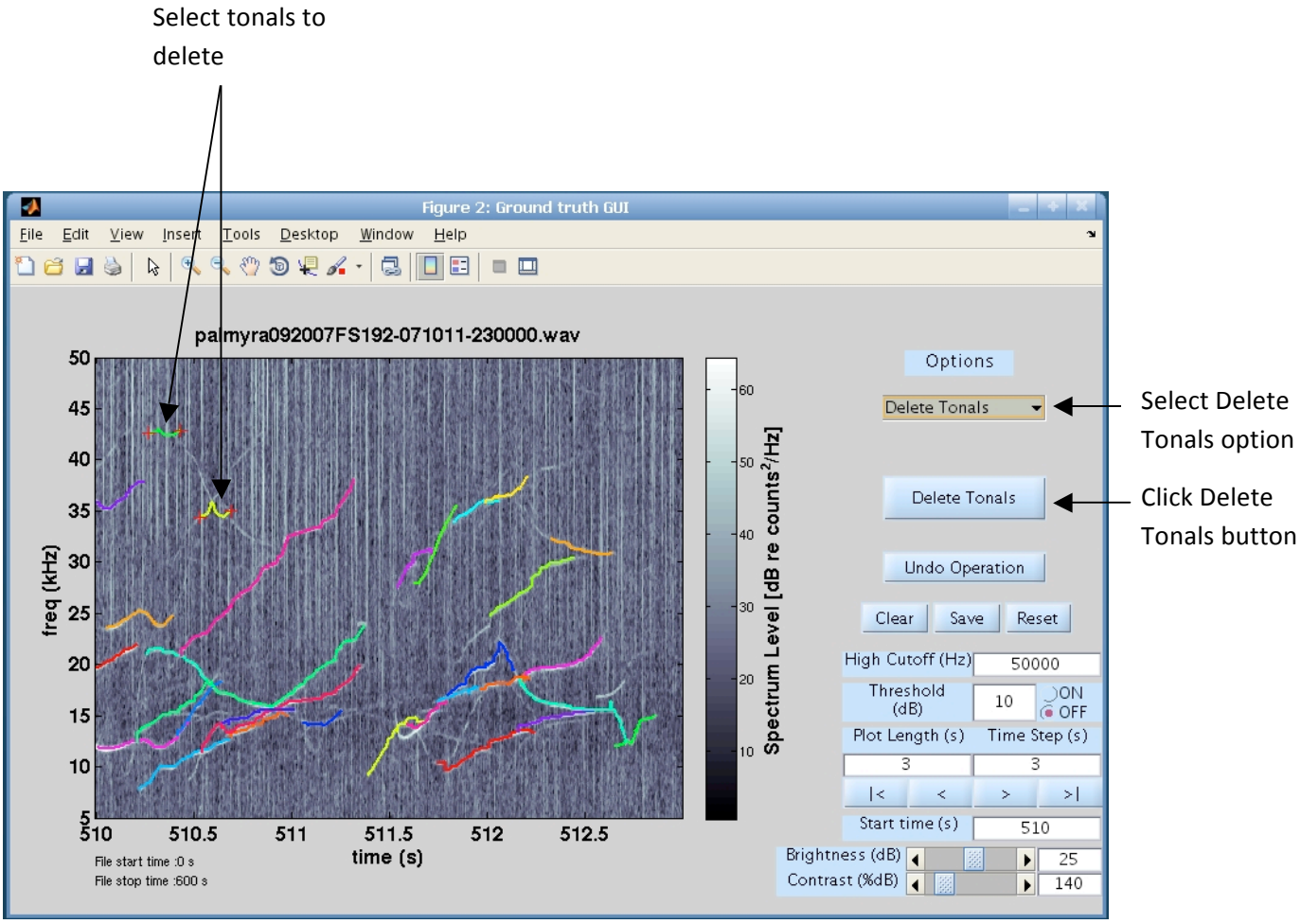
More than one tonals can be merged to create a single tonal. There are some pre-defined rules like the tonals that overlap in time frame cannot be merged. Furthermore, merging of tonals and adding of points to tonals can be done at same time if selection is in accordance to pre-defined rules.

Screenshot: Add/Merge option



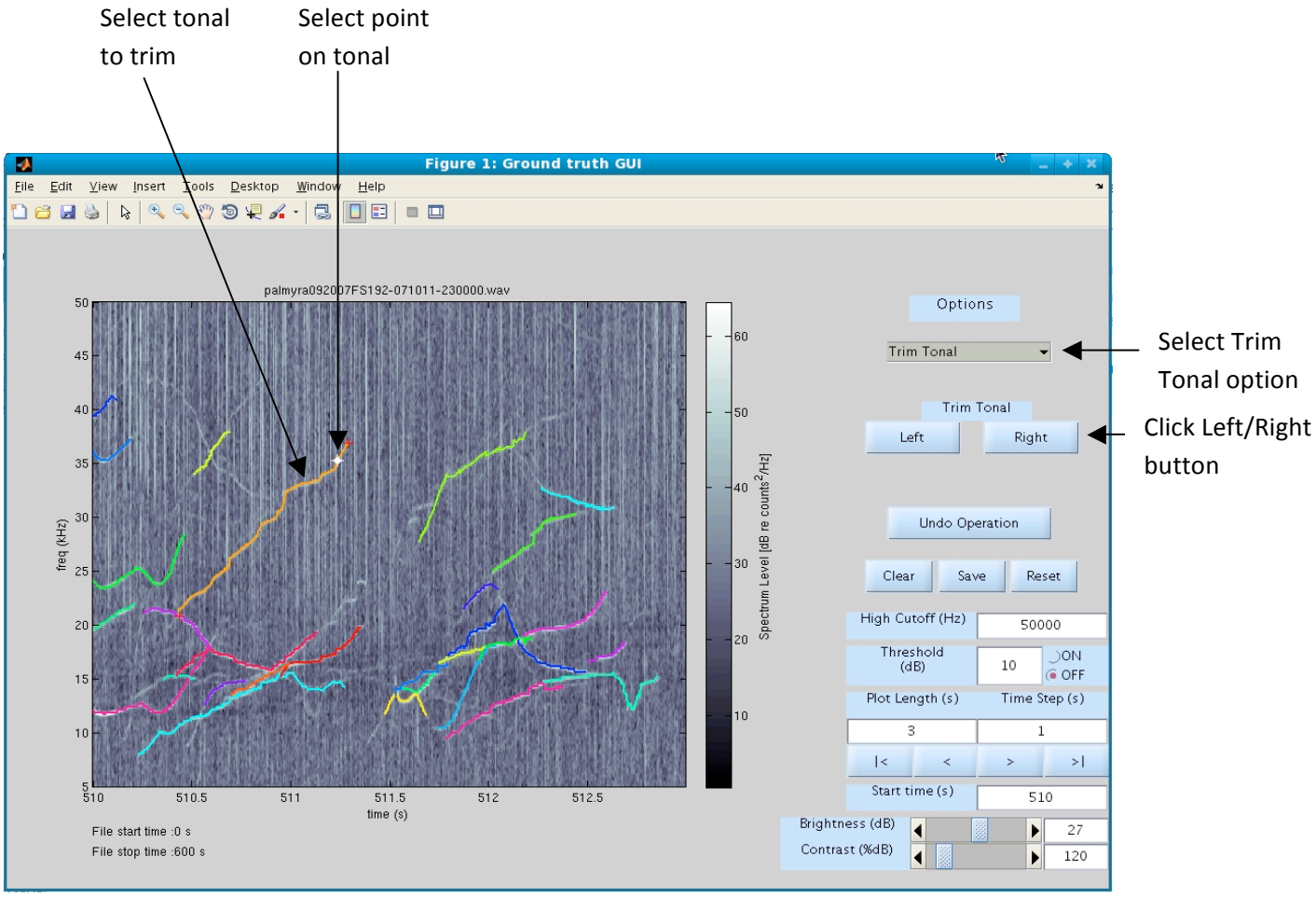
Delete Tonals: Selecting *Delete Tonals* option from pop-up menu allows user to delete multiple tonals. Multiple tonals can be deleted by selecting them and clicking *Delete Tonals* button that appears after the *Delete Tonals* option is selected from pop-up menu.

Screenshot: Delete Tonals option



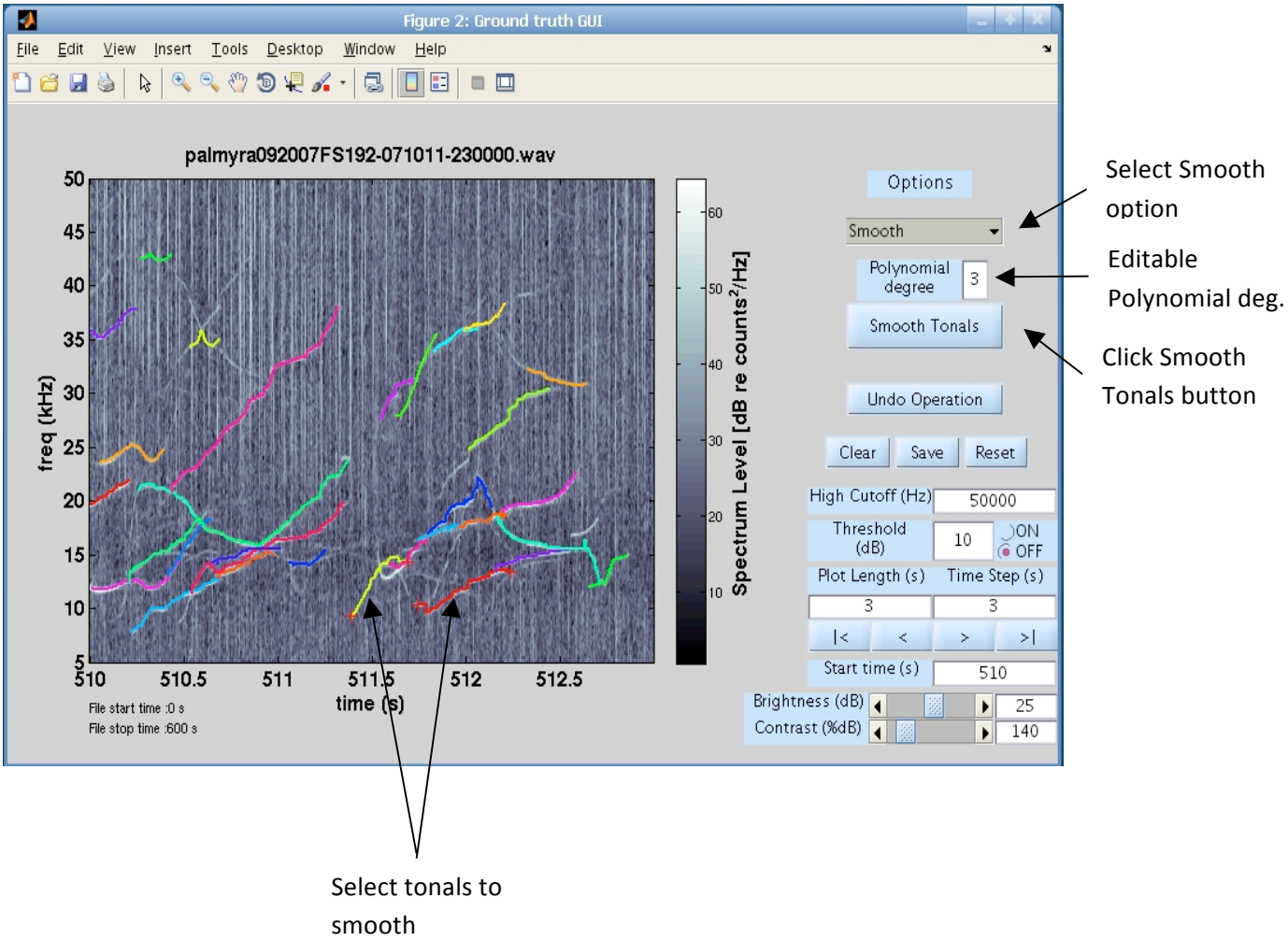
Trim tonal: Selecting *Trim Tonal* option from pop-up menu allows user to trim a tonal. Trimming operation is carried only on a single tonal. Trimming is done in three steps. In first step user has to select a tonal by clicking on it. In second step user again has to click on the tonal to select a point so that tonal can be trimmed towards left or right from the selected point on tonal. After selecting a tonal and a point on it user has to click *Left* or *Right* button to trim it. *Left* and *Right* button appears after *Trim Tonal* option from pop-up menu is selected. There are some pre-defined rules for trim operation like multiple tonals cannot be trimmed simultaneously. Error dialog box will be displayed if these rules are not followed.

Screenshot: Trim Tonal option



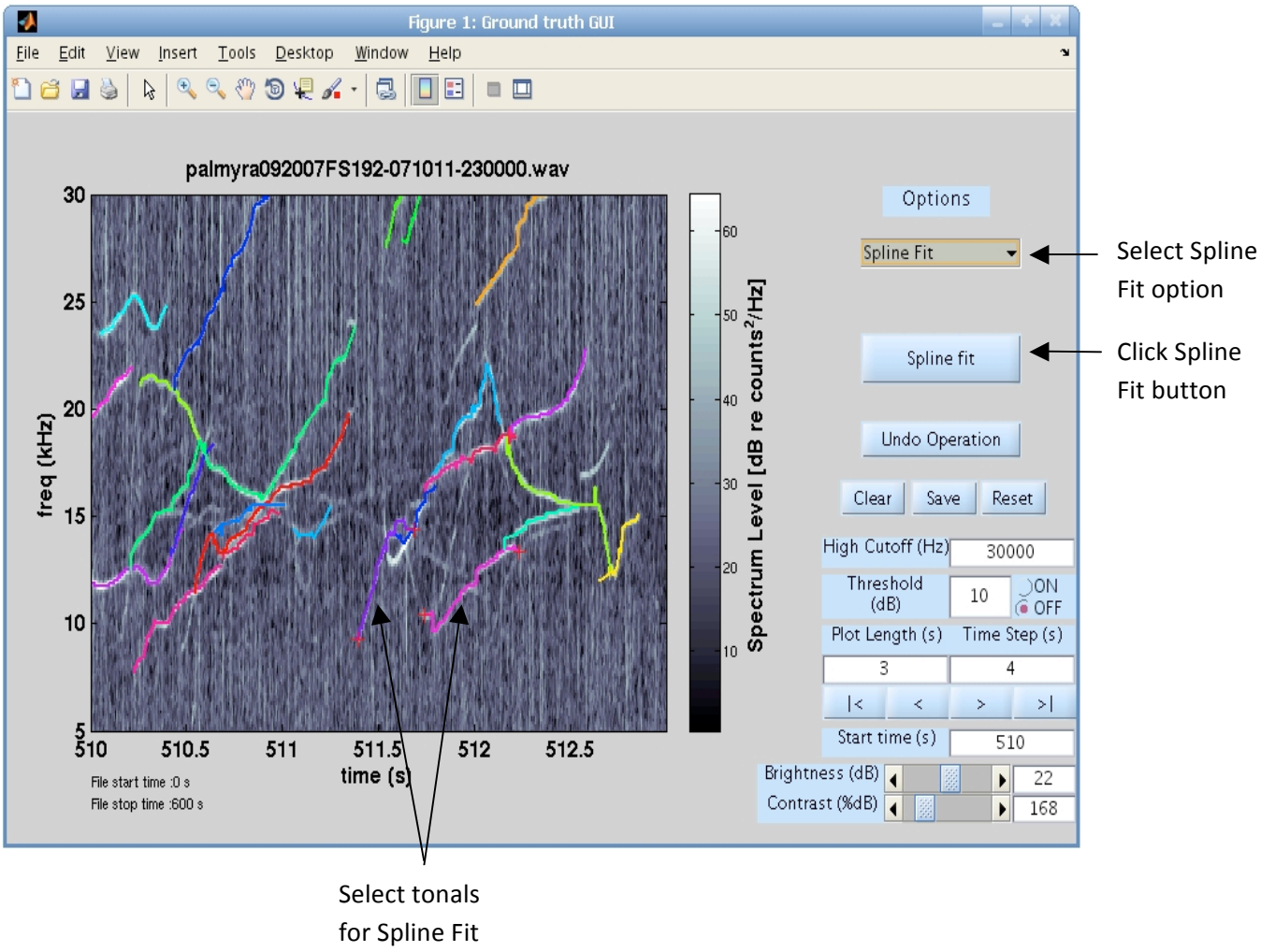
Smooth: Selecting *Smooth* option from pop-up menu allows user to perform smooth operation on multiple tonals. Polynomial fit is used to smooth tonals. User has to select tonals from the plot and click *Smooth Tonals* button in order to smooth tonals. User can modify the polynomial degree for polynomial fit from the edit box. Both edit box and *Smooth Tonals* button appears after selecting *Smooth* option from pop-up menu.

Screenshot: Smooth option



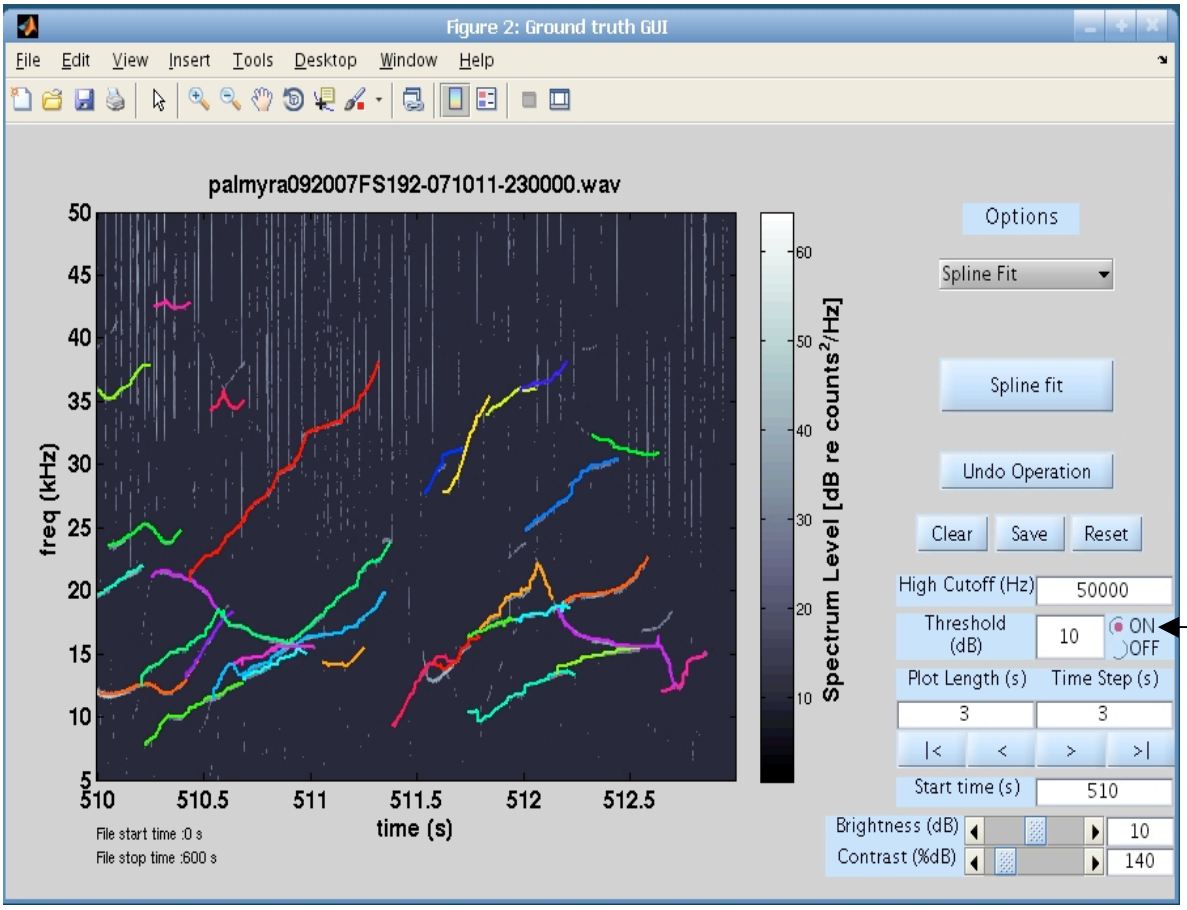
Spline Fit: Selecting *Spline Fit* option from pop-up menu allows user to perform spline fit on multiple tonals. User has to select tonals and click *Spline Fit* button to perform the fit. *Spline Fit* button appears after *Spline Fit* option is selected from pop-up menu.

Screenshot: Spline Fit option



Threshold ON/OFF: User can turn ON/OFF the threshold plotting. Threshold can be specified through threshold edit box. If threshold is on the threshold level is taken into account while plotting the spectrogram. Energy below threshold is indexed to 0 (black) and energy above threshold is indexed to spectral power. If threshold is turned off grey scale image of spectrogram is displayed.

Screenshot: Threshold is turned on



5.2 Controls

Undo operation: Most recent edit operation (Add/Merge, Delete, Trim, Smooth and Spline Fit) can be discarded by clicking *Undo operation* button. Not all the previous edit operations are discarded but only one that was most recently done is discarded.

Clear: *Clear* button clears the selection of objects (tonals or points) that may exists.

Save: *Save* button saves all tonals (edited and unedited).

Reset: *Reset* buttons allows user to discard all the edits.

Edit box:

High Cutoff (Hz): High cutoff frequency.

Threshold (dB): Energy Threshold. Energy below the threshold is indexed to 0 (black) and energy above threshold is indexed to spectral power.

Plot Length(s): Length of the spectrogram plot in seconds.

Time Step(s): Traverse forward or backward by time specified in seconds.

Start time(s): Spectrogram starts at a time specified in this edit box.

Brightness/Contrast (dB): User can enter value for brightness and contrast.

Scroll bar:

Brightness/Contrast (dB): Controls the brightness and contrast of the spectrogram.

Traverse Buttons:

|< (Start): This button allows user to go to the start of the recording section.

< (Previous): This button allows user to go back by certain time period in the recording. This time period can be specified by user in time step edit box.

> (Next): This button allows user to move forward by certain time period in the recording. This time period can be specified by user in time step edit box.

>| (End): This button allows user to go to the end of the recording section.

6. Advanced Topics

6.1 List of tonals:

Tonals extracted from automatic tonal detector (*Silbido*) and the ground tonals are java linked list objects. Each tonal itself is a list of nodes along the tonal contour. These nodes are represented by spectrum attributes time, frequency, signal to noise ratio (SNR) and phase angle. Both SNR and phase angle are set to 0.00 if their respective information is not present.

Example:

```
 >> [tonals graph] = dtTonalsTracking ({' palmyra092007FS192-071011.wav
   '}, 0, Inf);

>> tonals     %Displays the list of tonals

tonals =

[[(0.001 s, 8.50 kHz, 2.45 rad)(0.002 s, 8.50 kHz, 0.15 rad)…. (0.360
s, 13.50 kHz, 0.04 rad)],
.
.
[(122.428 s, 5.50 kHz, -2.67 rad)(122.431 s, 6.00 kHz, -1.20
rad)...(122.620 s, 8.50 kHz, -0.67 rad)]]

>> [traced_tonals] = dtTonalsLoad('palmyra092007FS192-071011-
   232000.bin');

>>  traced_tonals % Displays the list of tonals

traced_tonals =

[[(0.001 s, 8.50 kHz, 0.00 rad)(0.002 s, 8.50 kHz, 0.00 rad)...(0.360
s, 13.50 kHz, 0.00 rad)],
.
.
[(122.428 s, 5.50 kHz, 0.00 rad)(122.431 s, 6.00 kHz, 0.00
rad)...(122.620 s, 8.50 kHz, 0.00 rad)]]
```

6.2 Iterating over tonal list

Sample code for iterating over a list of tonals in Matlab is shown below.

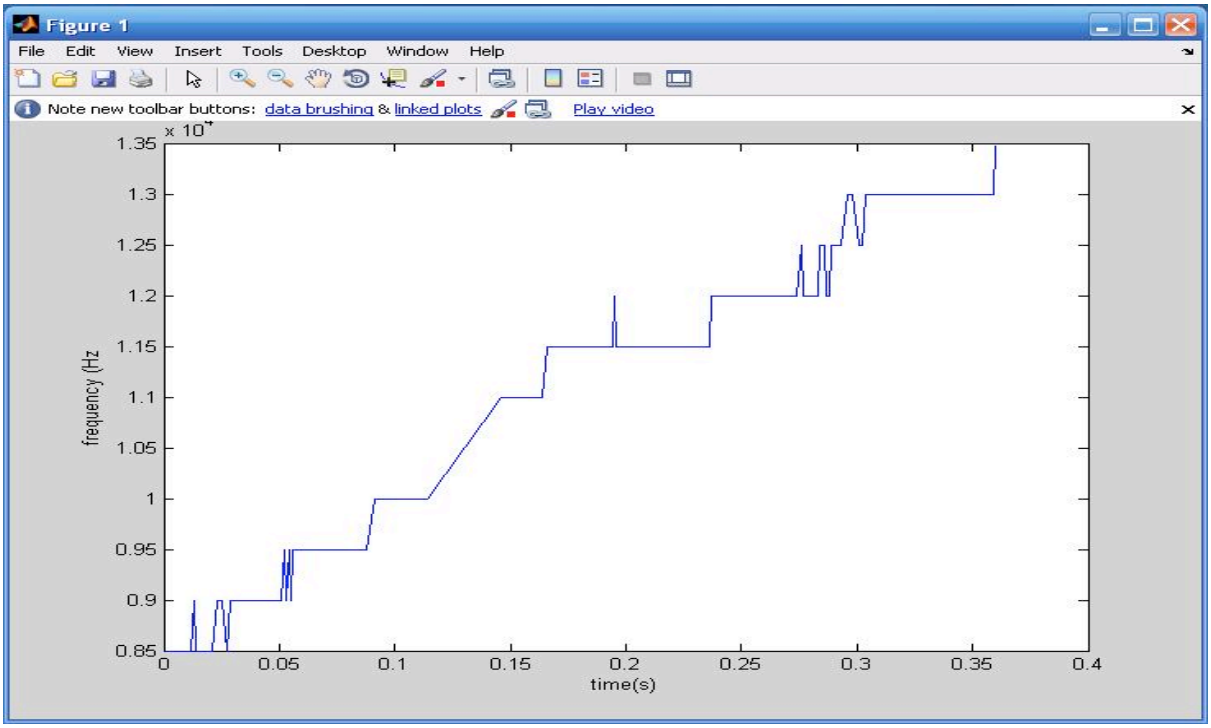
Assume `tonals` variable contains list of tonals. Tonals can either be extracted tonals detected by tonal detector *Silbido* or tonals that are retrieved after saving the tonals.

```
% Iterate over tonals
tonalsN = tonals.size();
for tidx = 0 : tonalsN -1
    tonal = tonals.get(tidx);

    % Get the time and frequency value of a tonal using java
    % method get_time() and get_freq()
    time = tonal.get_time(); % time is a double array
    freq = tonal.get_freq(); % freq is a double array

% One can plot the time and freq array to get better picture
% of a tonal shape by using plot command plot(time, freq, '-b');
end
```

Plot of a single tonal contour is shown below.



6.3 Constructing a tonal and list of tonals

Constructing a tonal from time and frequency arrays

To create a tonal in our format there is a tonal constructor that takes in two arrays time and frequency each of type double. Sample run in Matlab is shown below.

```
>> import tonals.*;      % Import Java's tonals package

>> new_tonal = tonal(time, frequency);
```

Creating list of tonals

```
>> tonals = java.util.LinkedList();     % Empty linked list created

>> tonals.add(new_tonal);      % Adds tonal to the list

>> new_tonal_1 = tonal(time, frequency);

>> tonals.add(new_tonals_1);  % Adds another tonal to the list
```

6.4 Java Resources

Java's executable file for whistle project is located at ../triton/java/whistle/whistle.jar
Java source files are located at ../triton/java/whistle/src
Java class files are located at ../triton/java/whistle/bin

6.5 Increasing Heap Space for JVM in Matlab

Create java.opts file:

For all users: Edit the java.opts file (which may not exist) by typing:

```
>> edit(fullfile(matlabroot, 'bin', computer('arch'), 'java.opts'))
```

For single user: Create the java.opts file in the Matlab startup directory.. If you wish to change the directory where Matlab starts in Windows, you can create a shortcut and edit the Start in property of the shortcut.

Value to put in java.opts file:

Matlab uses a set of default values for the initial and maximum heap size. Users can override these values by setting them manually in a java.opts file. For example, including the following line in a java.opts file sets the Max Heap Size value to 1024 MB:

For JVM of 1.2.2 and later:

```
-Xmx1024m
```

For JVM of 1.1.8:

```
-mx1024m
```

***NOTE: Increasing Java heap space decreases Matlab's as they run in the same process space.