Stereo Wavesurfer: A Tool for Dialog Analysis

Ernesto Medina and Thamar Solorio

Department of Computer Science
University of Texas at El Paso
500 West University Avenue
El Paso, TX 79968-0518

Email: {equintana3, tsolorio}@utep.edu

Technical Report UTEP-CS-06-40, revised version

October 17, 2006

Abstract

Researchers in the Interactive Systems Group at UTEP have been using a research tool called Didi for some time now. It was originally designed to be easily adaptable. This tool has proven to be adaptable as it has been changed by different researchers to suit particular needs. As a result, multiple versions of the program exist. In addition to this, the tool only works in Linux and has grown quite a bit. To solve these problems, the different versions could be consolidated into one program and modified to produce a version that worked on other platforms, or another program could be customized instead. In order to make an informed decision about which alternative was more feasible, research was conducted on a potential replacement candidate called Wavesurfer. The result of our research was a modified version of Wavesurfer called Stereo Wavesurfer. This new version extends the functionality of Wavesurfer to stereo files. Specifically, Stereo Wavesurfer allows for the display, analysis, transcription, and pitch calculation of individual tracks. This paper briefly describes Didi, Wavesurfer, the initial modifications to the original Wavesurfer, and possible future modifications to Stereo Wavesurfer.

1 Introduction

As part of their research effort, members of the Interactive Systems Group (ISG) at UTEP use a tool called Didi (Dialog Displayer) to analyze sound files. As shown by [4], its features include a simple display of sound, playback of sound, and labeling of conversation and speech signals. It is used by ISG in "looking for prosodic and voicing correlates of conversations phenomena" and in "labeling the words in a sample of speech to prepare the data for training word-based speech recognizers, labeling phrases from a dialog in order to use them as voice prompts, etc.,…" [4]. Figure 1 shows a graphic display of Didi's features. A more detailed description of this program is available at www.cs.utep.edu/nigel/didi.

Many of the members of ISG modified the original version of the program to suit a specific need. As a result, multiple versions of the program were created. To solve this particular problem, ISG could either merge the different versions of Didi or customize another program to do what Didi does. Part of our effort was to research a program called Wavesurfer, figure out what it had that was useful to ISG, and determine how difficult it would be to modify. The overall goal was to determine the feasibility of replacing Didi with a modified version of Wavesurfer.

The next section presents the basic features of Wavesurfer that are of interest for our research in spoken dialog systems. Section 3 describes how we modified Wavesurfer for supporting stereo files, which was the first most important feature to import from Didi. This section provides enough detail to allow other programmers to start customizing Wavesurfer. Section 4 discusses future modifications for Stereo Wavesurfer, and finally, Section 5 presents concluding remarks.

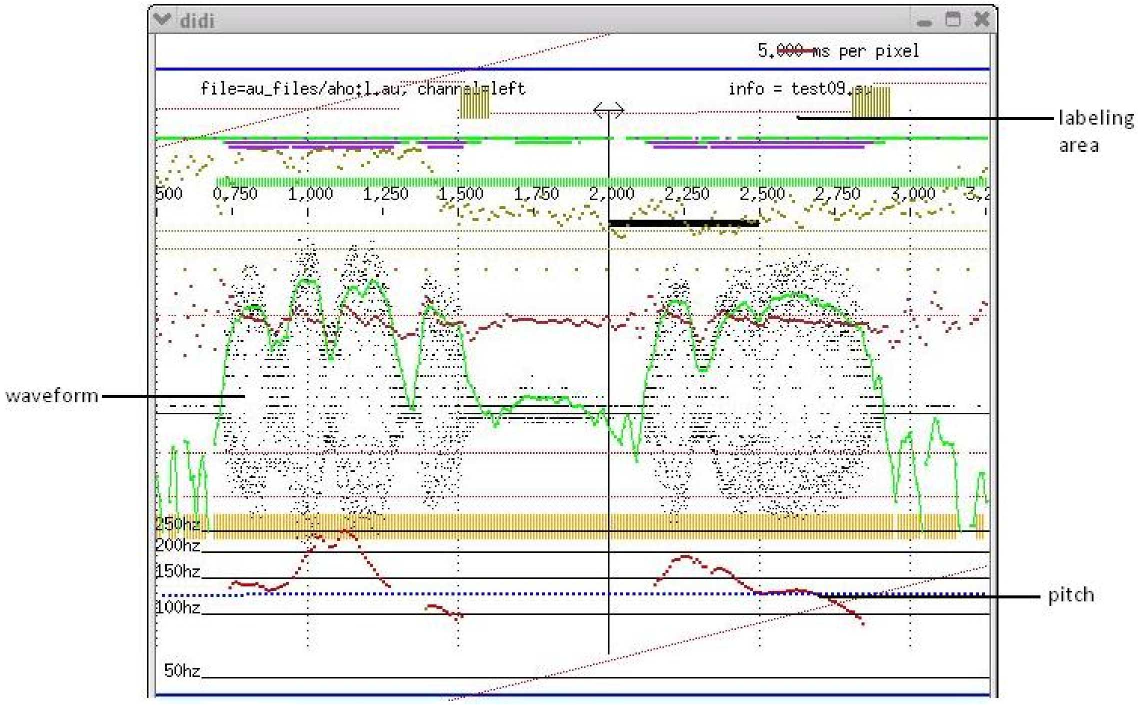

Figure 1: Snapshot of Didi

2 Wavesurfer

Wavesurfer is a sound analysis tool that can be used to visualize and manipulate sound files. It is freeware that can be modified and redistributed as wanted. One of the main reasons it was chosen is because it already supported most of the functions needed by ISG. Moreover, the documentation provided by [2] suggests that the program could be modified easily. Specific information can be found at the Wavesurfer website at http://www.speech.kth.se/wavesurfer/.

2.1 Wavesurfer functionality

Wavesurfer offers a variety of sound processing features. For brevity we would only like to mention a few. For more information about specific features or for other features not mentioned, please refer to the program's website at www.speech.kth.se/wavesurfer/documentation.html.

One of the features of Wavesurfer is the graphical display of the waveform created by a person's voice. When a person speaks, sound is created by a vibration of the vocal cords. As Carmell describes, "sound takes on the form of radiating waves of variation in air pressure around an average resting value at sea level of about 100,000 Pascals (Pa)" [1]. These radiating waves also have a frequency, or the number of repetitions of the wave. Carmell also mentions that microphones capture these changes in air pressure and transforms them into variations in electrical voltage. As seen in Figure 2, the waveform is present at the top of the interface. The line running down the middle of the waveform is the resting value. The black shaded regions represent the radiating waves of a voice. A time axis specifies the progression of time.

Another key feature is the ability for a researcher to listen to a sound file and transcribe a part of the file. In ISG's case, labels can be added to point out specific features of the sound file such as silence and response signals. Figure 2 points out these features.

Finally, Wavesurfer has the ability to plot a pitch contour for the sound file. As mentioned above, waveforms in general have a frequency. Pitch can be described as the perception of that frequency. A wave with a higher frequency has a higher pitch. Wavesurfer, as well as Didi, has the ability to measure and plot pitch values for a sound wave. This can also be seen in Figure 2.

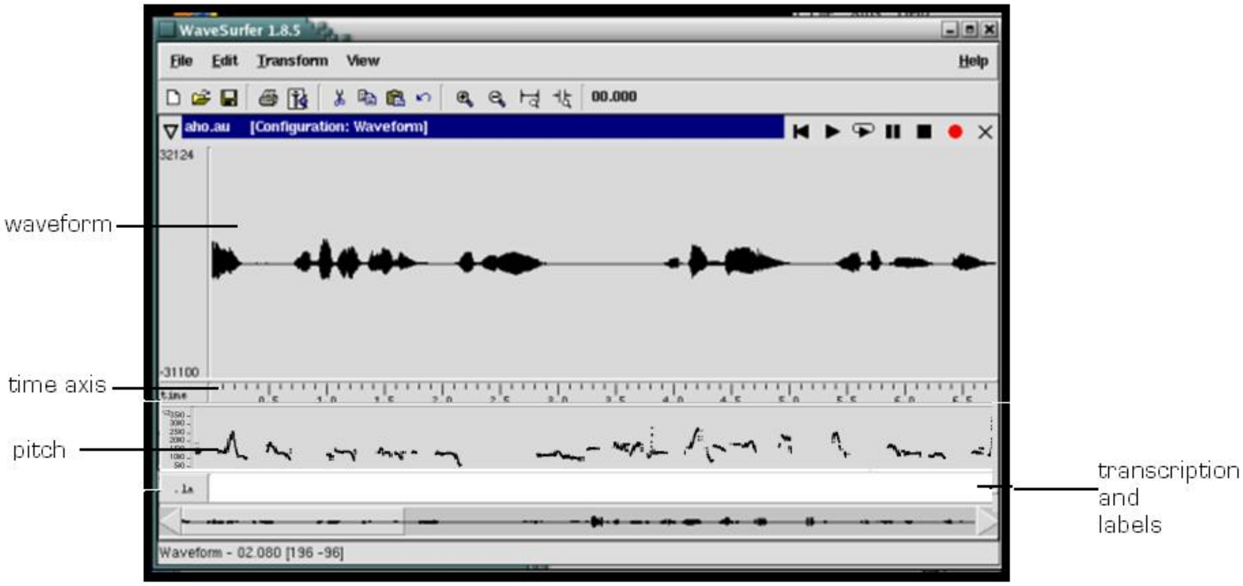

Figure 2: Snapshot of Wavesurfer

2.2 Implementation of original version

The original version of Wavesurfer was implemented in two steps. First the original creators developed the Snack Sound Toolkit. This toolkit is a library of functions written in the C programming language that allows direct manipulation of sound files. A few examples of this library include the ability to play a determined region of a sound file and the copying of a region of a sound file. The Snack Sound Toolkit provides all the functionality for the Wavesurfer program. The second step dealt with the actual interface of the program. The interface was designed using a scripting language called TCL. This scripting language was used to produce different widgets for each part of the interface such as the menus and the display panels. Separate configuration files were designed to control the layout of the interface. For example, there are certain configurations that only display the waveform. Last of all, plug-in files were also developed to enhance the functionality of Wavesurfer.

3 Modifications

Before Wavesurfer could be evaluated as a replacement candidate, it had to be modified to accommodate the tasks performed by researchers in ISG. In this section we describe what modifications we were able to perform and also how these improvements to Wavesurfer were achieved.

3.1 Motivation

On our initial investigation, we realized that Wavesurfer already provided much of the functionality needed by ISG. The basics such as waveforms, transcription, labeling, and pitch were already present. The problem was that Wavesurfer was not designed to support transcription, labeling, and automatic track detection. In order to determine if Wavesurfer was an ideal replacement candidate, we had to determine if stereo support along with all the missing functionality provided by Didi could be added. Moreover, we also had to determine how complex it would be to accomplish such a task.

3.2 Implementation

As mentioned before, the original version of Wavesurfer already provided most of the functionality that ISG needed. Despite this fact, Wavesurfer still needed a major modification. Certain features, such as transcriptions, labeling, and pitch contours cannot be applied to stereo files. The files that ISG works with are stereo files. These files contain recorded conversations between two people. Each voice is held on an individual track or an individual stream of data. Therefore it is important to be able to apply these features to both tracks of the file.

To modify the pitch, we integrated another program called Sox into Wavesurfer. As shown by [3], Sox is a versatile sound processing program. In our case, we used Sox to split the original stereo file into two mono files containing one track. In other words, each voice of the recorded conversations was now in a separate file. Then we used the original method created in Wavesurfer to compute the pitch of each mono file. More information on Sox can be found at their website http: //sox.sourceforge.net/.

Modification of the labeling and transcription only required one step. When a sound file is first opened, Wavesurfer searches for a file with the extension of .la. These files contain information about specific regions of the sound files that contain labels or transcriptions. If a label or transcription has never been created for the particular sound file, then of course a .la file will not exist. Now, the problem with the original transcription and labeling method was that it only created a main .la file for the original file. So for stereo files, only one track could be labeled. In order to be able to change this, we had to create two individual .la files, one for the left track and one for the right. Figure 3 shows the new version of Wavesurfer.

Stereo Wavesurfer can be found on the Interactive Systems Group website at the following URL: http://www.cs.utep.edu/isg/downloads.html.

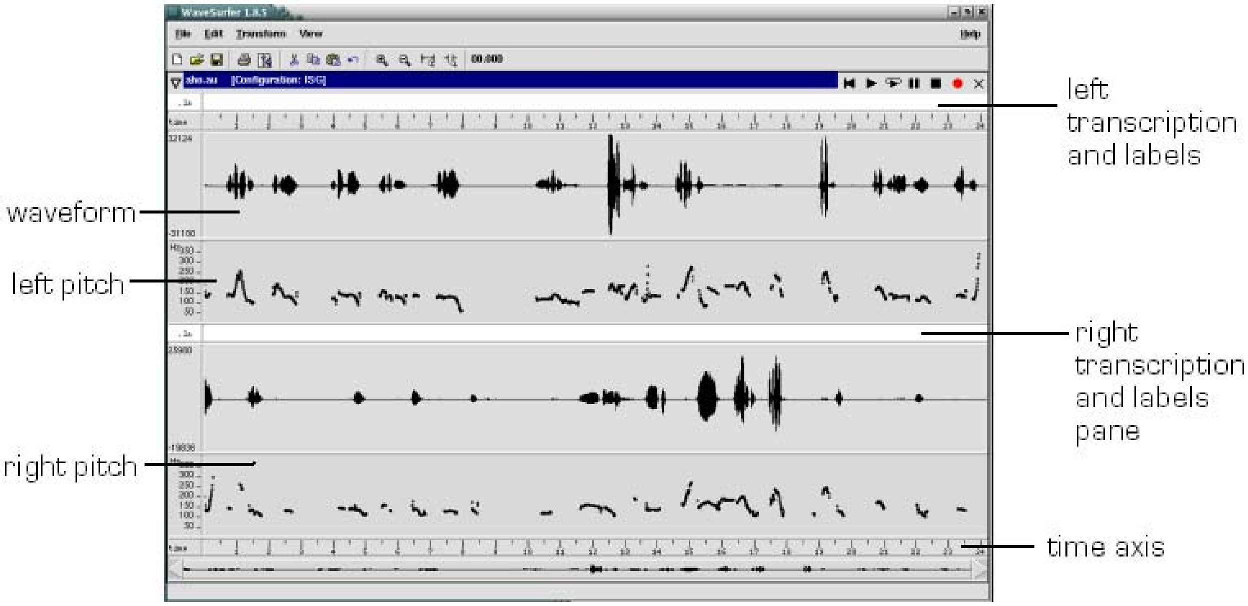

Figure 3: Snapshot of the modified version of Wavesurfer

4 Future Modifications

Another important determining factor in deciding if Wavesurfer would make a good candidate was whether or not Wavesurfer could be easily extended. The initial modification process provided us with enough details to determine that Wavesurfer could in fact be extended in a manner that was simple enough.

There are two aspects of Wavesurfer that would be of interest to modify, the interface and the functionality. The interface, or look and feel, of Wavesurfer can be changed by modifying the TCL files included in the program. Each part of the Wavesurfer interface is generated by different component files. These component files can be modified to change anything from the color of the interface to shape and size. For example, a certain part of the interface can be highlighted in green if desired. Basically, Wavesurfer's interface can be customized to the preferences of the user.

The functionality of Wavesurfer can be modified by changing the Snack Sound Toolkit. As mentioned previously, the Snack Sound Toolkit is the driving force of the functionality in Wavesurfer. Anything that cannot be done by the toolkit would need to be added. This would involve writing C files that incorporate C programming constructs and TCL programming constructs. This is basically a C file that imports the TCL library provided in C. So if you know how to code in C and TCL, you can add functionality.

After writing your new C files, a shared object file (.so) would need to be created for the C files and loaded into Wavesurfer. The loading process can be achieved by adding a line that reads load filename.so to the TCL files. Then the methods declared in the C/TCL files can be invoked. More information on writing these files and loading them into Wavesurfer is described in detail by [2].

In the process of making these modifications, we noticed one bug. The left edge of the first label or transcription is not preserved when the labels or transcriptions are saved. The left edge is moved to the zero mark. This bug was present in both the original and modified version of the program.

5 Summary

We learned several things as a result of our research and experimentation with the new version of Wavesurfer. First, we learned that Wavesurfer already provided a lot of the functionality found in Didi. Second, we learned that all aspects of the original Wavesurfer could be modified to suit a specific need. Last of all, we learned that Stereo Wavesurfer could be modified in a relatively efficient manner. Although in its current state Stereo Wavesurfer is not ready to take over the duties as ISG's research tool, these advantages make Stereo Wavesurfer a good replacement candidate.

To build on what we have learned, we have several interesting features in mind that we wish to add in the future to Stereo Wavesurfer. One thing would be to indicate with a label on the interface which part represents the left track of the sound file and which represents the right. For now, it is described only in the manual for the new version. Also, there is the need to have Stereo Wavesurfer compute the error between the pitch plotted by Stereo Wavesurfer and a fitted pitch curve.

Acknowledgements

This work was funded in part by The University of Texas System Louis Stokes Alliance for Minority Participation with help from the National Science Foundation through project HRD-0217691, by NSF Grant No. 0415150, and by DARPA. We are especially thankful to Dr. Ben Flores, Ariana V. Arciero, and The University of Texas at El Paso for their support.

References

[1] Carmell, T (1997). Spectrogram Reading.
http://cslu.cse.ogi.edu/tutordemos/SpectrogramReading/waveform.html

[2] Sjolander, K and Beskow, J (2006). Wavesurfer Documentation. Speech, Music and Hearing part of School of Computer Science and Communication.
http://www.speech.kth.se/wavesurfer/documentation.html.

[3] SourceForge.net (2006). Sox-Sound exchange. http://sox.sourceforge.net/

[4] Ward, N. G. (1998). Didi User's Manual. University of Tokyo.
http://www.cs.utep.edu/nigel/didi/didiman.pdf