Recent developments regarding the WaveSurfer speech tool

Sjölander, K.

journal: Proceedings of Fonetik, TMH-QPSR
volume: 44
number: 1
year: 2002
pages: 053-056

http://www.speech.kth.se/qpsr
Recent developments regarding the WaveSurfer speech tool

Kåre Sjölander
kare@speech.kth.se
Centre for Speech Technology, KTH

Abstract
The WaveSurfer speech tool and its associated toolkits have seen continued development and improvement since the initial release in January 2000. A major new feature is the inclusion of F0 and formant analysis functions from the ESPS suite of speech analysis tools. This was made possible because Microsoft and AT&T, current owners of the rights to this software package, generously made it available to the Centre of Speech Technology at KTH for inclusion in WaveSurfer. These ESPS functions are fully integrated in WaveSurfer and use the same open source license.

Other notable improvements is the inclusion of facilities for general curve plotting and editing, a built-in installation wizard for downloading extension plug-ins, and more flexible user interface when loading multiple sounds.

WaveSurfer is not only a tool in itself but also a platform for development of specialized tools that need sound signal visualization and analysis. It is seeing increased use in the speech research field and its libraries are being used in many other open source software projects. More information and downloads can be found at http://www.speech.kth.se/wavesurfer/.

Introduction
WaveSurfer is a tool to handle speech signal files and related data. It has been designed and developed at the Centre for Speech Technology at KTH (Sjölander & Beskow, 2000). The basic version can be used to play, record, view, edit, transform, and convert audio data in much the same way as can be done with many similar tools. WaveSurfer also serves as a base platform that has been adapted for use in speech synthesis, facial animation, automatic alignment, transcription, database validation, and different types of analysis. The program has been publicly available since January 2000 using an open source license and has seen widespread use since then. Some notable projects that have included WaveSurfer are the Linguistic Data Consortium’s Annotation Graph Toolkit (Maeda et al. 2002) and the Transcriber software (Barras et al. 2001). Also, many projects with a more narrow focus have successfully deployed WaveSurfer, for example OLP Tk (Öster et al. 2002). The use of the open source model (FSF, 2000) for software development and distribution has many benefits. More users mean more feedback and many of these help out in tracking down and fixing bugs and other problems. Software that is only used by a handful of people does not evolve and improve as quickly as software that is used by a larger group. The availability of the source code also makes it much easier for other parties to verify the correctness of computations and algorithms and improve on them if necessary.

Software architecture
WaveSurfer’s software architecture is very different from most contemporary software. It is common today to use compiled monolithic executables, sometimes with the added use of dynamic linking. WaveSurfer on the other hand has a relatively small compiled core, the Snack toolkit (Sjölander & Beskow, 2000), with most higher-level program-logic and parts of the user-interface existing in the form of a library, wsurf, written the interpreted scripting language Tcl/Tk (Ousterhout 1994). The WaveSurfer application itself is also created using the scripting language Tcl/Tk, see Fig. 1. Applications using the wsurf library can also be written in the scripting language Python (Lutz 1996). The use of scripting languages has several useful
implications. Programs are easy to update and change, even for novice users. It allows for flexible programming methods where data and code are treated in a uniform manner. There is no need to invent special file formats for program specific settings and similar support files. More importantly there is no need for specialized batch command languages, something more capable programs usually need. Instead the scripting language is used that invariably is more capable.

WaveSurfer features

WaveSurfer has certain features that make it different from other speech signal software. Each opened sound can have one or more time-aligned visualization panes attached to it. Most of WaveSurfer’s functionality is centred on these panes.

WaveSurfer’s configurations

The concept of a configuration is central in the use of WaveSurfer. A set of customized panes can be saved together for later use in a configuration. These play a central part in the use of WaveSurfer. For example, when browsing and editing sound files a simple waveform display with a time axis might be what is most practical to use. For simple phonetic transcription a spectrogram and a transcription pane might be enough. For detailed speech analysis it might be necessary to use waveform, spectrogram with overlaid formant plots, and pitch and energy contours. Using saved configurations designed specifically for each task is time saving when working with many files. It is also practical to set up a certain configuration for use by for example students or other novice users for a certain task.

Plug-ins

Usually plug-ins adds optional functionality to a program. In WaveSurfer they have a much more central part. Most of WaveSurfer’s functionality is located in its plug-ins. The wsurf library, for example, only provides empty time aligned panes, no content. It is up to the plug-ins to fill these with content. A basic set of plug-ins is included as default giving analysis, transcription, and plotting capabilities. New optional plug-ins can be found, downloaded, and installed from within WaveSurfer itself using the plug-ins dialog.

The difference between configurations and plug-ins is only conceptual, they both consist of program code, which is, executed either during application initialization or when the user requests a certain configuration. The fact that program code is used also makes it possible to reuse it in other applications.

Embedding

Not only does WaveSurfer consist of several components it is also easy to use it as a component in other applications as well. For some tasks, for which the WaveSurfer application might be too general and cumbersome to use, it is more convenient to create specialized applications where only the specific functionality needed is provided. One example is a validation and mark-up tool developed for the SPEECON speech database project, http://www.speecon.com/. In the Swedish part of this project about 600 speakers will be recorded under different conditions with several different microphone placements. The speech material is intended to be used to train speech recognizers for use in controlling, for example, home electronics products. The aim in creating this tool was to minimize time, effort, and error during manual validation and transcription. The material is transcribed orthographically and a few extra tags are used, for example, to mark disturbances. Most of the material is read speech and as long as the sound files contain the expected speech only one keyboard click is needed per sound file. This single click accepts the current transcription, loads the next set of files and start playback. The tool uses a wsurf widget to present the sound files. This can be used by the operator to select specific portions of a signal for detailed study. Also, it is possible to use different
configurations and analysis types. A screen-shot of the tool is shown in Fig 2.

Figure 2. The SPEECON validation tool that uses the wsurf widget library.

ESPS functions

The ESPS tools get_f0, for automatic pitch extraction, and formant, for automatic formant tracking, were chosen for the inclusion in WaveSurfer initially. They are both stand-alone command line tools. Also, they store their output (and intermediary results) in files using proprietary file formats. In order to make them fit into the WaveSurfer framework they were both modified and included in the Snack toolkit as library functions. In this way they could benefit from this toolkit’s internal data handling facilities. The new commands now process Snack sound objects instead of ESPS signal files; also they return lists instead of creating files on disk. If needed data files have to be created in a separate step. WaveSurfer simply uses the list representation internally as with most other Snack commands. This is more efficient and no temporary files are generated in the process. See Fig. 3 for a screen-shot of WaveSurfer with ESPS formant and F0 functionality.

Future work

WaveSurfer’s capabilities are quite mature in many areas, but there is always room for improvement. There are still many interesting functions remaining in the ESPS libraries that could be integrated into WaveSurfer. Hopefully there should be several interested parties that might help out in this process. Also, WaveSurfer might be extended to include modules that so far has been used in-house only. An improvement of a totally different nature is the planned creation of discussion forum that would help spread information about WaveSurfer and also allow users to get in contact with each others to get help and share ideas. It is our hope that WaveSurfer will continue to change and improve with the needs and requirements of the speech research community.

Acknowledgements

This work was carried out at the Centre for Speech Technology, supported by Vinnova, KTH and participating Swedish companies and organizations. We also would like to thank all users of WaveSurfer who have contributed improvement suggestions, bug reports, actual code, or encouragement.

References

Figure 3. Screen-shot showing ESPS functionality in WaveSurfer. In the middle a spectrogram with overlayed formant tracks and at the bottom an F0 plot. It is possible to manually edit the plots and to save the data values to disk.