The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals

Chris Cannam, Christian Landone, Mark Sandler, Juan Pablo Bello
Centre for Digital Music, Queen Mary University of London
Mile End Road, London, UK
chris.cannam@elec.qmul.ac.uk

Abstract

Sonic Visualiser is the name for an implementation of a system to assist study and comprehension of the contents of audio data, particularly of musical recordings.
It is a C++ application with a Qt4 GUI that runs on Windows, Mac, and Linux. It embodies a number of concepts which are intended to improve interaction with audio data and features, most notably with respect to the representation of time-synchronous information. The architecture of the application allows for easy integration of third party algorithms for the extraction of low and mid-level features from musical audio data. This paper describes some basic principles and functionalities of Sonic Visualiser.

Keywords: Visualisation, Musical Feature, Semantic Descriptor.

1. Introduction

The Sonic Visualiser project originated as an integrated framework to aid researchers at the Centre for Digital Music at Queen Mary University of London in the development of algorithms for the extraction of low and mid-level features from musical audio signals.

The internal requirements called for a cross-platform application that could enable the user to browse and edit time-synchronous musical features overlaid and aligned to waveform and spectrogram representations of the audio signal under analysis. A further requirement was to allow researchers to integrate feature extraction algorithms and use the application as a test bed to evaluate their performance.

Although tools for audio analysis and annotation are readily available in the open source space, these applications are mostly inherited from the linguistics and speech analysis communities and enhancements have been introduced in order to extend their usability to the music information retrieval community.

A modified version of WaveSurfer [1] has been adopted as the annotation client for the MUCOSA environment [2] and various automatic feature extraction plugins have been developed for the same application [3].

Also ad-hoc tools for the visualisation of musical features have emerged in recent times; the CLAM Annotator [4], for instance, natively allows the user to browse and edit automatically extracted descriptors, as well as to manually enter "ground truth" annotations.

Whilst both WaveSurfer and the CLAM Annotator associate each data set to a separate canvas, the Sonic Visualiser project was commenced in order to provide the users with an innovative approach to the visualisation of musical features, where each set of low and mid-level descriptors is presented as a logical layer, in a fashion similar to an image processing application.

The management of machine-extracted descriptors in the Sonic Visualiser differs substantially from that employed by the CLAM Annotator, that requires an external executable following specific command line and output conventions. The Sonic Visualiser, instead, supports a specialised type of plugin named "Vamp", that has been developed in parallel at QMUL's Centre for Digital Music.

This choice was driven by the need to introduce a common C/C++ API for the development of feature extraction algorithms in order to speed up development time, increase the potential for dissemination and harmonise legacy code.

2. The Application

The Sonic Visualiser has been developed in C++ using the open source distribution of QT4 [5] for the user interface and is licensed under the terms of the GNU General Public License, with the hope of promoting further and constant community-driven enhancements.

The application uses stackable panes that can be added or removed at the user's request and can show a number of different pieces of data overlaid on one another and aligned according to time. The overlay approach assists comparisons among extracted and annotated data, as well as aiding understanding of the original audio data by allowing it to be viewed in more than one form "at once".

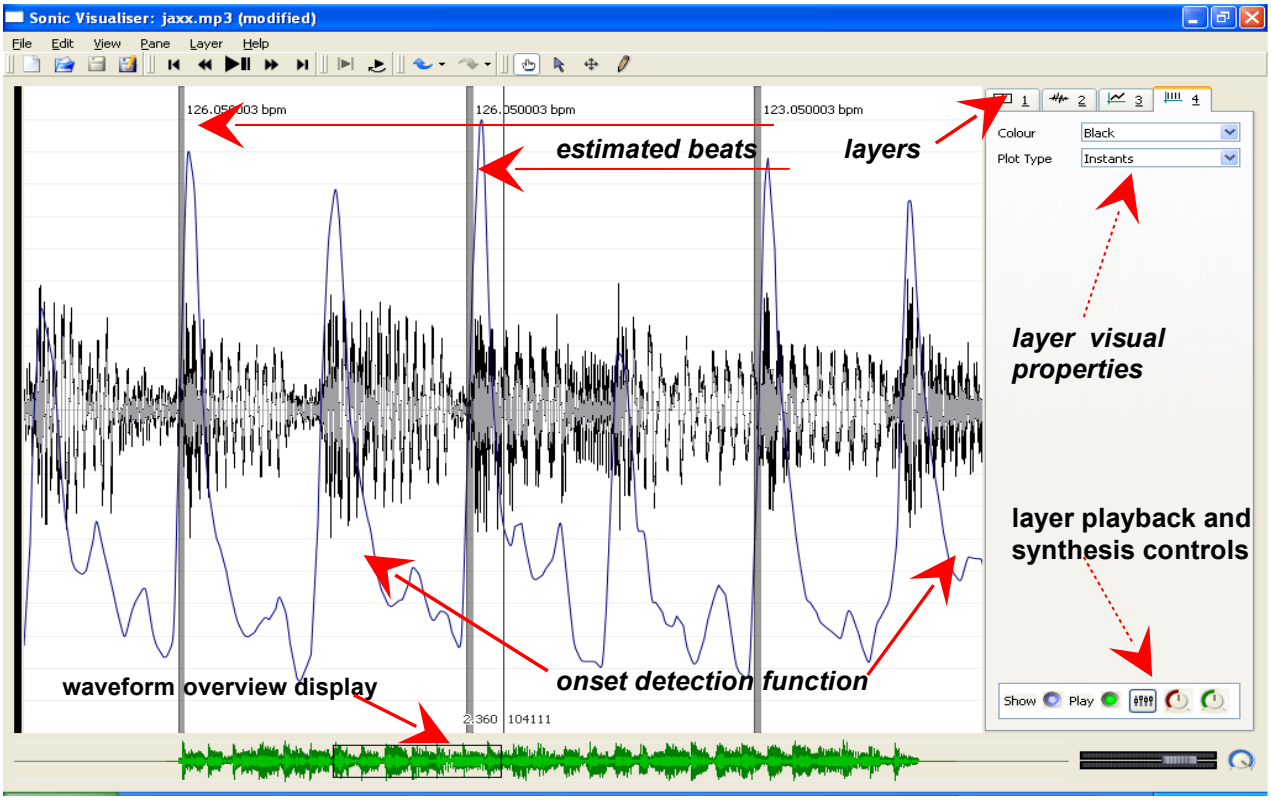

Figure 1. Waveform with beat detection function and estimated beats

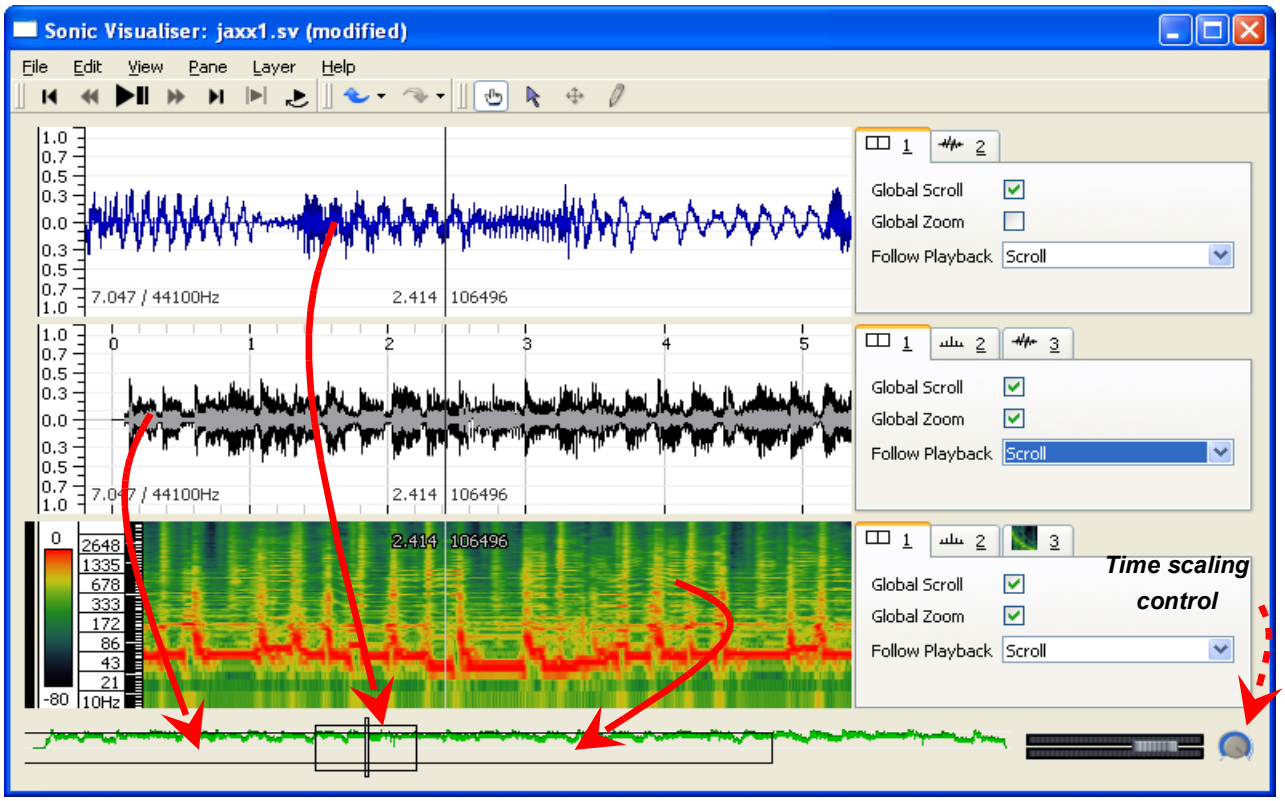

Figure 2. Waveforms and spectrograms at different zoom levels

This facility may be used with various sorts of data, including points in time, values on a discrete time graph, as well as waveforms and spectra. Figure 1 shows a detail from a pane with four layers:

1 - The pane scroll/zoom properties layer.
2 - A mono waveform.
3 - The output of a beat detection function: a series of user-editable values shown as a curve.
4 - Estimated beats: a series of instants in time shown by a vertical bar at each beat location

Each layer has a set of visual properties (for example colour or plot type in layer 4); these are available in a stack of property boxes shown to the right of figure 1. A layer may be visually raised above the other layers by selecting its property box from the stack.

The y axes of separate layers on the same pane do not necessarily conform with one another; for example, there is no meaningful common y axis between a waveform and spectrogram on the same pane.

In contrast to the y axes, x axes of separate layers on the same pane always align exactly by audio frame. For example, data in a particular spectrogram window covers the correct x extent corresponding to the source audio frames for that window. Similarly, beats detected using a function that operates on a particular window size are shown using vertical lines of width corresponding to that of the window.

In addition to the user-definable panes, an overview of the time domain waveform under analysis is shown in an area at the bottom of the application's user interface (figure 1); this is used also to display the position of the playback cursor (vertical line) as well as the extent of the current view (rectangular box).

2.1 Multi-resolution representation

When more than one pane is used in the Sonic Visualiser, these are usually aligned to the same sample frame in the middle of the pane. The user can scroll through the audio by dragging the pane left and right with the mouse.

However, the separate panes do not have to be shown at the same time resolution and different zoom levels can be applied to each pane independently.

Figure 2 shows the same source in two waveform panes and a spectrogram pane, with the top pane zoomed to a significantly higher resolution than the second and the third ones, this is also reflected in the waveform overview display, where multiple rectangular outlines show the different zoom extents (pointed out by the solid arrows in figure 2)

The top pane covers a period of approximately one second in time, while the second and third panes cover approximately six seconds. The top pane is therefore effectively a detailed view of the centre one-second of data in the bottom panes. If the user were to scroll through the file around this point, the two frames would scroll together, with the top pane appearing to scroll more rapidly than the bottom one, whilst maintaining the same sample frame at their centre.

2.2 Annotation Layers

In its current implementation, the Sonic Visualiser also provides the user with four different types of manual annotation layers that can be exported to a file or used to import and visualise comma and space separated, or MIDI data files.

The "Time Instants Layer", shown at the top of figure 3, allows the insertion of vertical lines at points in time that may represent onsets or beats, while the "Time Values Layer" (middle of fig. 3) is an editable sequence of points in time, where each point has a value that determines its position on the Y axis. Each point may also have a label.

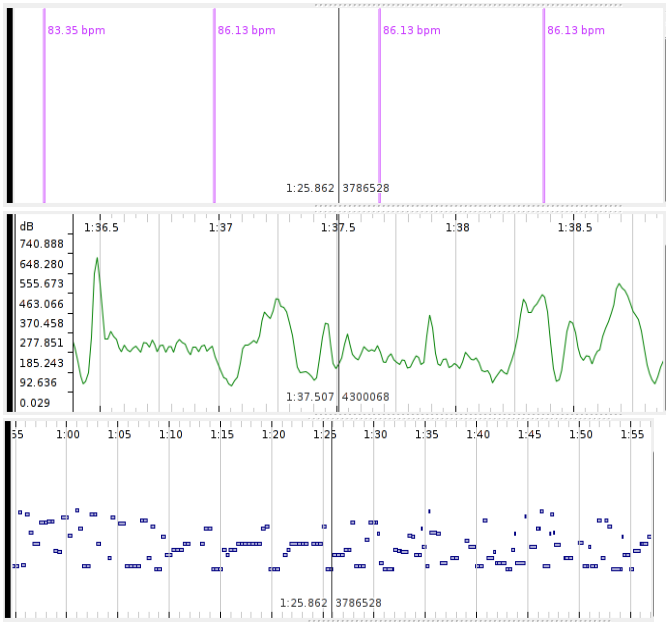

Figure 3:Editable layers

The notes layer (bottom of figure 3) consists of an editable sequence of notes with an associated label where each note has a start time, a duration and a value that determines its position on the Y axis.

A text layer is also available for placing freely around the pane informal or descriptive annotations that are not specifically attached to other features.

The Sonic Visualiser allows entire work sessions to be saved to file, enabling the instant recall of the audio file under analysis, visualisation styles and settings, annotations and data layers.

2.3 Playback of audio and features

Audio data can be played by the Sonic Visualiser and a full set of transport commands is provided.

To aid analysis and annotation, looping and time scaling during playback are provided. The amount of time scaling can be varied in real time using a rotary control in the user interface, shown in figure 2.

Features can also be "listened to" using the playback and synthesis controls shown in figure 1; each of the data layers displayed in the Visualiser has a set of performance characteristics associated with it, including auralisation method (e.g. play a particular sound each time a data point is hit during playback), output level, and stereo pan.

It is therefore possible to compare aurally a set of calculated or annotated features by panning them separately or playing them with different sounds.

3. Vamp plugins

An external API is used by the Sonic Visualiser for the automatic extraction of low and mid-level descriptors from musical audio data.

Vamp plugins are dynamic link libraries that process sampled audio data, returning complex multidimensional data with labels representing semantic observations that are automatically loaded by the Sonic Visualiser and shown on a new layer.

The plugins receive their data block-by-block but are not required to return output immediately on receiving the input, therefore a Vamp plugin may be non-causal, preferring to store up data based on its input until the end of a processing run and then return all results at once.

The feature extractor can indicate to the host the preferred processing block and step sizes, and may ask to receive data in the frequency domain instead of the time domain. The host takes the responsibility for converting the input data using an FFT of windowed frames or, alternatively, cache frequency-domain data when possible, allowing the plugins to do straightforward frequency-domain processing.

A Vamp plugin is configured once before each processing run, and receives no further parameter changes during use – unlike real time plugin APIs in which the input parameters may change at any time. This also means that fundamental properties such as the number of values per output or the preferred processing block size may depend on the input parameters.

The cross-platform plugin SDK is distributed under the terms of a BSD-style license, allowing for closed-source feature extraction modules to be used by the Sonic Visualiser and thus permitting the safe dissemination of musical analysis algorithms that are yet unpublished and/or not protected by patents.

Vamp plugins are already available: implementations of algorithms for onset detection [6] and beat tracking [7] are available from the Centre for Digital Music website [8] while other functionalities are available from the Mazurka project [9].

4. Conclusions

A framework for the automatic extraction, browsing and editing of musical feature descriptors has been presented in this paper. The Sonic Visualiser runs on Linux, Windows and MacOS and proposes a novel "layered" approach to musical features visualisation. The binaries and source code, along with the Vamp plugin SDK are available for download from the project's web site: **www.sonicvisualiser.org**.

5. Acknowledgments

Part of this work has been done in the context of the SIMAC IST-507142 and EASAIER EU-FP6-IST-033902 projects. The authors would like to thank Daniel Leech-Wilkinson and Craig Sapp for their valuable suggestions.

References

[1] K. Sjölander, J. Beskow, "WaveSurfer - An Open Source Speech Tool", *in Proceedings of the International Conference on Spoken Language Processing, 2000.*

[2] P. Herrera *et al*, "MUCOSA: A Music Content Semantic Annotator", *in Proceedings of the 6th International Conference on Music Information Retrieval, London 2005.*

[3] F. Gouyon, N. Wack, S. Dixon, "An Open Source Tool for Semi-Automatic Rhythmic Annotation", *in Proceedings of the 7th International Conference on Digital Audio Effects, Naples 2004*

[4] X. Amatriain *et al*, "The CLAM annotator: A Cross-Platform Audio Descriptors Editing Tool", *in Proceedings of the 6th International Conference on Music Information Retrieval, London 2005.*

[5] QT4 open source edition: www.trolltech.com

[6] J.P. Bello, C. Duxbury, M.E. Davies, M.B. Sandler, "On the use of Phase and Energy for Musical Onset Detection in the Complex Domain", *IEEE Signal Processing Letters*, Vol 11, No. 6, pp 553-556, June 2004

[7] M.E. Davies, M.D. Plumbley, "Beat Tracking with a Two State Model", *in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2005*

[8] http://www.elec.qmul.ac.uk/digitalmusic/downloads/

[9] http://sv.mazurka.org.uk