Seminar Wissenschaftliches Rechnen

Alexander Gustafson
ZHAW Zurich

Project Proposal : Audio FIle Analysis

The goal of this project is the implmentation of a C++ cross platform application that will analyse audio files. The application will utilize the MPICH MPI Library in order to effitiently utilize multi-core or multi-computer environments. The primary analysis tool will be a fourier-based frequency anaylsis of audio data.

16.03.2013

Audio Analysis

Frequency Analysis

The user should be able to select an audio file, or folder containing audio files. The application will then begin to divide the audio file(s) into smaller overlapping slices. Each slice will be treated with a windowing function that will fade-out the edges. Windowing reduces false spectral artifacts that would result from discontinuities resulting from the slicing. Windowing also helps to "reduce the problem of spectral leakage"[1], but I do not really understand this yet.

The result of the Fourier Transform will be presented to the user in graphic form, as a spectrograph showing the magnitude of the frequency components over time. The application should preserve the numeric results of the fourier transformation, including the imaginary component, so that the original signal might be "reconstructed".

Further Analysis

The application will also calculate the enery level or loudness of each slice in order to gain information about the dynamics of the audio signal. By analysing how often and how quickly the amplitude or energy of a signal changes it should be possible to make estimations about the tempo or the type of content contained in the audio signal.

The application will display the results in graphical form to the user.

The Application

The application will developed in C++. It will be cross-platform compilable. It will have a graphical interface that will alow the user to select one or multiple audio files to analyse. It will alow the user to initiate the analysis, diplay the results in a managed and graphical form that make sense in the context of the data to be displayed. It will also record information about the number of threads/cores used and the time needed to finish processing. Results of the analysis should be exportable, at least as a csv, so that the data can be imported into other applications for testing and comparison.

Tools, Libraries, and Frameworks

Audio File Processing

JUCE [2] is a cross platofrm C++ framework that includes many classes for importing and handling of most common audio file formats (WAV, AIFF, Flac and Ogg-Vorbis). JUCE is released under the GNU Public License, commercial licenses are also available.

Application and GUI Framework

I would also like to use JUCE as the application and GUI framework. Juce offers a large set of cross platform components as well as basic image support that will be usefull for displaying the results of the audio analysis. I have been able to compile the JUCE demo applications on both my Mac and Linux computers with little effort. Although JUCE can be compiled with the BOOST library JUCE already offers many of the advanced C++ feautures that BOOST has such as smart pointers and modern string manipulation functions for example.

Project Repository

I have created a github repository for this project. It will be used to host all code and documentation as well as sample files for anaysis. The url is : https://github.com/alexgustafson/WR-Project

Parallel Processing

The C++ MPI Library MPICH will be used to distribute processing among available cores or processors.

Information Resources

Fourier and Math Background:

Parallel Scientific Computing in C++ and MPI, George Em Karniadakis and Robert M. Kirby II

Signal Processing for Commnications, Paolo Prandoni and Martin Vetterli, EPFL Press, 2008

http://www.katjaas.nl/

Stephan Bernsee , http://www.dspdimension.com/admin/dft-a-pied/

References

1 http://www.katjaas.nl/FFTwindow/FFTwindow.html

2 http://www.rawmaterialsoftware.com/juce.php