DSP Project

Pitch shifting of audio signals

This document provides an outline for a Matlab project to be completed by the end of the course. You are expected to investigate, in detail, methods related to solving the problem. There is a design element to the project, and a quantitative evaluation of the performance of the proposed methods must be performed and presented. You are to write up a comprehensive report (of no more than 8 pages) describing your method and results. You should work in groups of two, although you may work alone if you really want to.

If you wish to propose a project of your own, then please come and talk to me.

The task: Pitch shifting of audio signals

The aim of this project is to investigate methods of shifting the pitch of audio signals. For example, given a soundtrack of a person singing, you might want to process the data to make them sing one octave lower.

A simple way to change the pitch is to compress or expand the signal in time. Expanding the signal will shift all the frequencies present to lower frequencies. Unfortunately, however, this will also make the signal longer and it will take longer to play it back — so the audio will effectively be slowed down.

A fairly simple method of altering pitch without changing the length is presented by Keith Lent, "An efficient method for pitch shifting digitally sampled sounds," Computer Music Journal, vol. 13, no. 4, pp.

65–71. This can be accessed online from on the UCT campus at `http://www.jstor.org/stable/3679554`. I haven't implemented it myself, but the algorithm seems quite straightforward and the paper is explicit on programming details. Alternatively you may choose to implement a different method — the internet seems to have plenty of resources relating to this problem.

The task is to implement a pitch shifting method and analyse its performance. If you have experience with embedded hardware you may choose to implement a real-time variant, or alternatively it can be done offline using the programming language of your choice. Note that the implementation probably won't be trivial, so if you have absolutely no idea of how to program then you should perhaps choose another topic.

As always, you will have to formulate experiments and generate a comprehensive set of results to assess the performance of the algorithm. This is always difficult: you need to develop performance measures that are effective at quantifying how well the system is working. Just saying "It sounds right" does not suffice. Exploring around the different choices of parameter settings is also usually appropriate.

The report should not exceed 8 pages. While it can discuss the software structure or design it is not appropriate to include source code: the project is to assess the *method*, not your specific implementation of it (although that might deserve a mention).