A TIME STRETCHING AND PITCH SHIFTING WINDOWS PHONE APPLICATION

Richard Guerci, Joseph Timoney and Tom Lysaght

National University of Ireland, Maynooth

richard.guerci@telecomnancy.net; jtimoney@cs.nuim.ie;tlysaght@cs.nuim.ie

The development platforms for mobile applications have improved significantly in recent years to such an extent that it is possible to consider building applications that would have only been possible on a powerful personal computer beforehand. A number of competing platforms exist now for mobile application development: Apple's IOS, Google's Android and Microsoft's Window Phone 8. Although arriving a bit later than the other two Microsoft's Windows phone has been catching up quickly with its competitors. Windows Phone developers work in C#, which is comparable to Java. The User Interface is tile-based and offers multi-touch technology. A reasonably larger screen size also allows developers room to include interactivity in their applications This paper will detail how to create a Windows Phone application for the Time-stretching and Pitch shifting of audio files. Time stretching or time expansion/compression leaves the pitch of the signal intact while changing its tempo, whereas pitch shifting is the reciprocal process to Time Stretching: it is a way to change the pitch of a signal without changing its length. This manipulation of audio could be ideally suited to a phone application as the touchscreen could facilitate manipulation of the stretch and shift parameter values. This application has a number of uses including being used as a real-time computer music sound effect processing tool. Also, it could be a sound effect inside a more significant library of audio tools.

The main challenge of this work was to determine how a Windows Phone development environment could be harnessed properly to facilitate such an audio transformation algorithm. This algorithm is quite powerful and time consuming, requiring a number of Fast Fourier Transforms to be computed. This is especially burdensome on mobile devices as they have not been originally designed to do this kind of complicated and heavy processing. A second issue to be overcome was the user interface design that must be intuitive and also well integrated to the Windows Phone 8 design and philosophy. The answers to both questions will be addressed in this paper.