Guide to DVD Chapter 19 Examples: *John Glover*

Sound Manipulation With Spectral Modeling Synthesis

These are the examples discussed in the text. To build them requires the C or C++ compiler and several additional software packages.

Package Managers

Some operating systems have *package managers*, which make finding and installing the dependencies listed below a lot easier. For example Ubuntu Linux comes with *apt-get,* so from the command-line you can install *scons* by running:

```
$ sudo apt-get install scons
```

The package manager will then search an online repository to find, download and install the software for you. Most of the major Linux distributions have a similar package management system; Fedora, for example, has *yum* and *rpm*. Some of these systems also have graphical front-ends so you do not need to do everything from the command-line (Synaptic Package Manager on Ubuntu Linux for example). If you are unfamiliar with Linux package managers, more information should be available on the homepage of your distribution.

For Mac OS there is *MacPorts* (http://www.macports.org/). Once installed, you can install additional packages in a similar manner from the command-line. To install libsndfile for example, run:

```
$ sudo port install libsndfile
```

It is not necessary to use these package management systems as all of the libraries below can be downloaded from their respective web sites; but they do generally make life easier though.

Dependencies

The following libraries are required to build the example programs. All are freely available and open source. The examples are known to work with the listed versions of these libraries, although they may also work with subsequent versions.

C Math Library

This should be distributed with the *gcc* compiler, but it must be linked to explicitly when building the examples. See below for examples. For other compilers, consult your compiler documentation.

Libsndfile (version 1.0.17)

Instructions for downloading and installing libsndfile can be found at http://www.mega-nerd.com/libsndfile/

GNU Scientific Library (version 1.11)

Instructions for downloading and installing the GNU scientific library can be found at http://www.gnu.org/software/gsl/. To build the examples you must link to both the *gsl* library and the *gslcblas* library, both of which are part of the main GNU scientific library distribution. Examples are given below.

Python (version 2.5 or later recommended)

Python is only needed in order to use *SCons*, and so any version after 1.5.2 should work. However, version 2.5 or later is recommended by *SCons* in order to improve performance. Instructions for downloading and installing Python can be found at http://www.python.org

SCons (version 1.2.0)

Instructions for downloading and installing *SCons* (version 1.2.0) can be found at http://www.scons.org/

A Note for Mac OS Users: If you are installing *SCons* from source code on the command-line, you may need to include the `--standard-lib` option. So, the full command would be:

```
$ sudo python setup.py install --standard-lib
```

libsms (version 1.1)

Instructions for downloading and installing *libsms* (version 1.1) can be found at http://mtg.upf.edu/static/libsms/. In these examples, we do not need to install several of the additional libsms software tools that are built by default, so we will build libsms by calling

`scons` with the `tools=no` argument. For example, on Mac OS X or Linux, at the command prompt go into the libsms directory and run the following commands to build and install it:

```
$ scons tools=no
$ sudo scons install
```

Building The Examples

To build the examples call gcc and link to all of the above libraries, as demonstrated below.

```
$ gcc –o analysis_synthesis analysis_synthesis.c -lsms -lm -lsndfile -lgsl -lgslcblas

$ gcc –o transpose transpose.c -lsms -lm -lsndfile -lgsl -lgslcblas
$ gcc –o transpose_with_env transpose_with_env.c -lsms -lm -lsndfile -lgsl -lgslcblas

$ gcc –o timestretch timestretch.c -lsms -lm -lsndfile -lgsl -lgslcblas

$ gcc –o change_env change_env.c -lsms -lm -lsndfile -lgsl -lgslcblas
```

Running The Examples

All of the examples can be run from the command line, but the arguments that they require may differ slightly depending on the example. A list of arguments for each example is given below, along with an example of running it from the command line.

analysis_synthesis

Argument 1: path to the input sound to analyze.
Argument 2: path to the output sound that will be generated.

Example: `$ ./analysis_synthesis flute.wav flute_synth.wav`

transpose

Argument 1: path to the input sound to analyze.
Argument 2: path to the output sound that will be generated.
Argument 3: transposition amount, which is a number of semi-tones. 2 will transpose up by 2 semi-tones, -3 will be 3 semi-tones lower, etc.

Example: `$ ./transpose flute.wav flute_transposed.wav 4`

transpose_with_env

Argument 1: path to the input sound to analyze.
Argument 2: path to the output sound that will be generated.
Argument 3: transposition amount, which is a number of semi-tones. 2 will transpose up by 2 semi-tones, -3 will be 3 semi-tones lower, etc.

Example: `$ ./transpose_with_env voice.wav voice_tranposed.wav 4`

timestretch

Argument 1: path to the input sound to analyze.
Argument 2: path to the output sound that will be generated.
Argument 3: the amount of time stretching. A value of 1 will leave the duration unchanged, 2 will make the synthesized file twice as long, 0.5 will make it half the original duration, etc.

Example: `$ ./timestretch flute.wav flute_stretched.wav 2.0`

change_env

Argument 1: path to the source input file. The spectral envelope from this file will be used to shape the harmonic partials of the target input file.
Argument 2: path to the target input file.
Argument 3: path to the output sound that will be generated.

Example: `$ ./change_env voice.wav flute.wav new_flute.wav`

Input Sounds

Several sound files have been included in the *examples* folder for this chapter; but it should be possible to use any sound file that is supported by libsndfile. More details can be found on the libsndfile web site (http://www.mega-nerd.com/libsndfile/). However, when using the default SMS parameters (as the examples in this text do), we recommend keeping to monophonic sounds stored in 16-bit mono audio files with a sampling rate of 44.1 KHz in order to get the best results.

The specific examples used in the text and in this guide are two recordings taken from the **freesound** project, which can be found at http://www.freesound.org.

The *flute.wav* sound is available at http://www.freesound.org/samplesViewSingle.php?id=21673

The voice.wav sound is available at
http://www.freesound.org/samplesViewSingle.php?id=4228

Both sounds were converted to 16-bit wav files at 44.1 KHz.