GlottDNN

A full-band glottal vocoder for
statistical parametric speech synthesis

USER MANUAL
Version 0.1

January 18, 2017
Manu Airaksinen & Lauri Juvela

Contents

1 Introduction

1.1 Background

1.2 Licence and distribution

1.3 Authors & contact

1.4 Acknowledgements

2 Quick-start guide

2.1 Installation

This distribution intended for use in Unix-like systems that support the GNU autotools build-chain. Standard ./configure script and Makefile are provided for easy compilation. The vocoder C++ code has the following library dependencies:

- `GSL` (GNU scientific library), for basic linear algebra and FFT etc.
- `libsndfile`, for reading and writing audio files
- `libconfig`, for reading structured configuration files

Usually the best way to install the dependencies is with the system package manager. For example, in Ubuntu 14.04, install the packages `libgsl0-dev`, `libsndfile1-dev`, `libconfig++-dev`

Alternatively, you may download the source code for the libraries and compile them yourself. Make sure to add the built libraries into the `LD_LIBRARY_PATH`, so that configure can find them. The libraries (and their documentation) are available at

https://www.gnu.org/software/gsl/

http://www.mega-nerd.com/libsndfile/

http://www.hyperrealm.com/libconfig/

Additionally, this package uses the C++ wrappers for GSL provided at http://gslwrap.sourceforge.net. The code is included in this distribution, but is set to build as a separate library.

We recommend to use the GitHub distributed version to get the latest updates, and for the ease of other people contributing to the development. First, clone the code to a suitable location with

```
git clone https://github.com/ljuvela/GlottDNN.git
```

Then, if everything goes well, you should be able to compile with the basic

```
./configure
```

```
make
```

If something went wrong, it might help to use autotools to regenerate the configure and make files.

```
autoreconf
```

```
automake --add-missing
```

Python

SAMPLING_FREQUENCY : Sampling frequency should match that of the wav file

FRAME_LENGTH : Analysis frame length (in ms)

UNVOICED_FRAME_LENGTH : Analysis frame length in unvoiced frames. Shorter frames can better capture plosives and other impulse-like unvoiced events.

F0_FRAME_LENGTH : Frame length used for fundamental frequency analysis.

FRAME_SHIFT : Frame rate (in ms)

LPC_ORDER_VT : LPC order for the vocal tract filter

LPC_ORDER_GLOT : LPC order for the glottal source

HNR_ORDER : Number of ERB bands for Harmonic-to-noise ratio

DATA_TYPE : Data type for saving and reading parameters. Valid types are "ASCII" / "DOUBLE" / "FLOAT"

6.2 Pulse extraction related parameters

MAX_PULSE_LEN_DIFF : Percentage of how much pulse length can differ from F0. Pulses are searched iteratively until the nearest pulse fulfilling the length condition is found.

PAF_PULSE_LENGTH : Pulses-as-features length in samples. If interpolation is not used, this should be large enough to fit two pitch periods at the lowest F0.

USE_PULSE_INTERPOLATION : If true, two pitch-period pulses are interpolated to fill the feature vector. If false, the pulse is only centered at GCI.

USE_WAVEFORMS_DIRECTLY : If true, the speech waveform is used directly instead of the inverse filtered waveform.

USE_FOUR_PERIOD_PULSES : If true, Four pitch-periods are used instead of two.

PAF_WINDOW : Select the windowing function applied to the pulse at analysis. Valid options are "NONE"/"HANN"/"COSINE"/"KBD"

USE_PAF_ENERGY_NORM : Normalize the pulse to unit energy.

Appendices