An Open-Source Octave Toolbox for VTLN-Based Voice Conversion

Denis Stadniczuk [#*1], Gregor Bauckmann [+*2], David Suendermann-Oeft [*3]

[#] *Robert Bosch GmbH*
Stuttgart, Germany
[1] denisstadniczuk@gmail.com

[+] *Daimler AG*
Stuttgart, Germany
[2] gregor.bauckmann@daimler.com

[*] *Baden-Wuerttemberg Cooperative State University (DHBW)*
Stuttgart, Germany
[3] suendermann@dhbw-stuttgart.de

***Abstract*—This paper introduces the Voice Conversion Octave Toolbox made available to the public as open source. The first version of the toolbox features tools for VTLN-based voice conversion supporting a variety of warping functions. The authors describe the implemented functionality and how to configure the included tools.**

I. Introduction

Voice conversion is the task of converting the speech of a source voice into a target voice [1]. Prior work of the authors [2], [1] discussed a number of techniques suitable for this task causing many academic and commercial institutions to express their desire to obtain access to the described technology. Inspired by this interest, we started an endeavor to devise an open-source toolkit implementing the prior art in this field as described in the aforementioned publications.

The two most popular voice conversion paradigms are (by number of publications in major conferences and journals):

- Gaussian mixture models (GMMs) [3] and
- frequency warping [4].

In the past decade, a number of improvements to these techniques were proposed mainly addressing the rather poor speech quality original approaches produced. These techniques include

- residual prediction [5],
- global variances [6],
- weighted frequency warping [7].

The present first version of the Voice Conversion Octave Toolbox is limited to frequency warping based on the well-studied technique of vocal tract length normalization (VTLN) [8]. We decided to use the programming language Octave due to its strength of expressing complex signal processing and numerical data manipulation operations with little realty space. Also, due to its compatibility to the widely adopted industry standard Matlab, we expect broad applicability of the code in the community.

As shown in Figure 1, from a high level, the current version of the toolbox consists of the following three modules:

- a pitch tracker,
- VTLN,
- pitch-synchronous overlap and add (PSOLA).

These modules will be explained in more detail in Section III.

The toolbox is available for download as a GIT repository at

https://github.com/DenisStad/Voice-Conversion

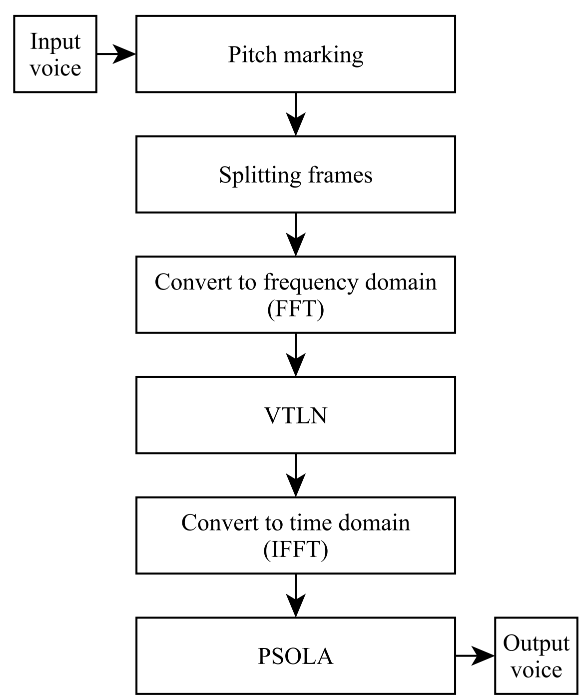

Fig. 1. Block diagram for VTLN-based voice conversion

II. System Requirements

The toolbox was tested on Octave Version 3.6.4 on a Mac OSX platform as well as on a Windows 7 platform. In addition to the Octave functionality of the toolbox, there are a few algorithms based on other environments or programs including

- Perl Version 5.8,
- Praat[1] [9] Version 5.3.31,
- Cygwin Version 1.7.5.

The toolbox is also executable under Matlab (tested with Version 2012b).

III. VTLN-Based Voice Conversion

According to [2], the following steps are required to convert a voice using the VTLN technique:

- split the speech utterances into pitch-synchronous frames,
- convert frames into frequency domain,
- convert frames using VTLN,
- convert frames back into time domain,
- concatenate frames using PSOLA.

A. Fundamentals

Human speech production is often considered to be the result of a source-filter model [10]. In this model, the source represents the vocal cords producing a voiced excitation. The fundamental frequency corresponds to the rate with which the vocal cords vibrate. The virtually flat spectrum of the excitation is shaped by the vocal tract depending on which specific sound has to be generated. The vocal tract is represented by the filter.

A consonant is a non-periodic signal, whereas a vowel is a nearly periodic signal. Vowels can be represented by their frequency content or, more specifically, by the spectral peaks of the sound spectrum which are called formants. The formants of a speech signal vary for different voices: Female voices generally have formants with higher frequencies.

B. Pitch Marking

As motivated above, speech is a series of pseudo-periodic or non-periodic signals, varying over time. Figure 2 shows a sample of a voiced region of speech. Most speech processing applications segment speech signals into frames of 10 to 30ms duration to account for temporal changes of sound characteristics. Frames are then processed individually. In speech synthesis, the duration of individual frames is very important, because it affects the speech quality when subject to concatenation during PSOLA. In particular, it was found that the splitting of a speech signal into frames that match the pseudo-periodicity of voiced sounds as determined by the fundamental frequency of the voice (pitch) produces the best speech quality. The process of cutting speech into pitch-synchronous frames is called pitch marking.

The authors decided to use Praat for pitch marking, since it produces reliable pitch marks (cf. pitch tracker evaluation presented in [11]), but also to provide an implementation

[1]`http://www.fon.hum.uva.nl/praat`

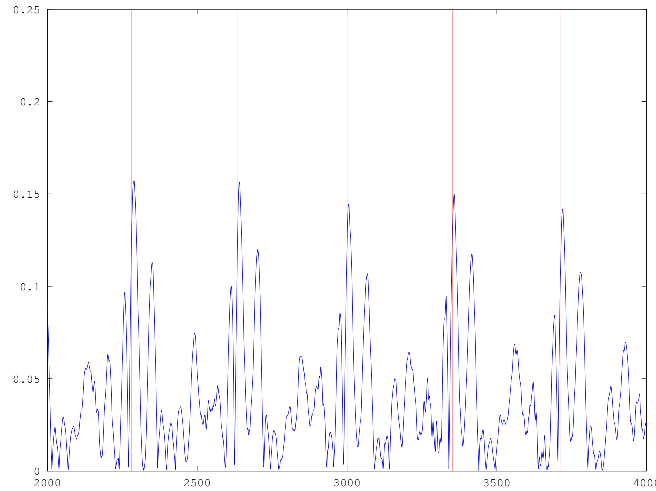

Fig. 2. Result of the pitch marking algorithm

written in Octave . Both Praat and the Octave implementation find the pitch marks as follows: The first one is the maximum of the interval $t_0 - \frac{T}{2}$ to $t_0 + \frac{T}{2}$, where t_0 is the middle of the speech and T is the period length at t_0, calculated by using the fundamental frequency f_0. The next pitch mark to the left of t_0 (t_{-1}) is between $t_0 - 1.2T$ and $t_0 - 0.8T$. The exact location is the maximum in this interval. This step is repeated to the left as well as to the right until the beginning of the speech is reached.

Praat's pitch marking algorithm is based on the auto-correlation method to perform acoustic periodicity detection [12]. As compared to other auto-correlation methods, Praat applies $\frac{sinx}{x}$ interpolation in the lag domain. Additionally, Praat uses Gaussian instead of Hamming windows resulting in a decreased pitch determination error.

The following describes how to use Praat to extract pitch marks for the Voice Conversion Octave Toolbox:

In the directory of the toolbox where we expect all Octave and other commands to be executed, there is a folder `data` containing one example speech file `sample.wav` [2] .

Praat uses an own scripting language to record macros and enable batch processing which are used for pitch marking in this toolbox. For this purpose, the toolbox contains the script `PraatToPitchMarks.praat`. Praat provides a GUI as well as a command line interface. If you want to run the aforementioned script from the GUI simply open it from Praat and enter the path to your wav file and specify where the pitch marks should be saved. Alternatively, you can use the script from the command line as follows:

```
Praat PraatToPitchMarks.praat
sample.wav pitchMarks.pm
```

where `sample.wav` is your source speech file and `pitchMarks.pm` is the file where pitch marks are saved.

[2]Taken from `NATOPhoneticAlphabet.ogg, Michael R. Irwin, http://commons.wikimedia.org/wiki/File%3ANATOPhoneticAlphabet.ogg`

The next step is to convert the Praat format of storing pitch marks into an Octave data object. To do this, run the Perl script `convertPraatToMatlab.pl` from the toolbox:

```
perl convertPraatToMatlab.pl
pitchMarks.pm pitchMarksMat.txt
```

where `pitchMarks.pm` is the file obtained from Praat and `pitchMarksMat.txt` is the destination of the Octave data object.

C. Vocal tract length normalization (VTLN)

In order to change the voice into another, the spectrum of a frame has to be transformed. Vocal tract length normalization is used to warp the spectrum of a frame, i.e., stretch or compress the spectrum with respect to the frequency axis which represents normalized frequencies in the range $0 \leq \omega \leq \pi$. Frequencies are altered according to a *warping function* $g(\omega)$ which needs to be a monotonous function returning values between 0 and π. g returns the warped position of the original frequency. Commonly, g depends on a warping parameter α affecting the shape of the function. In the present version of the Voice Conversion Octave Toolbox, the warping function is one of five predefined functions as shown in Table I. Figure 4 shows the corresponding illustration of these functions.

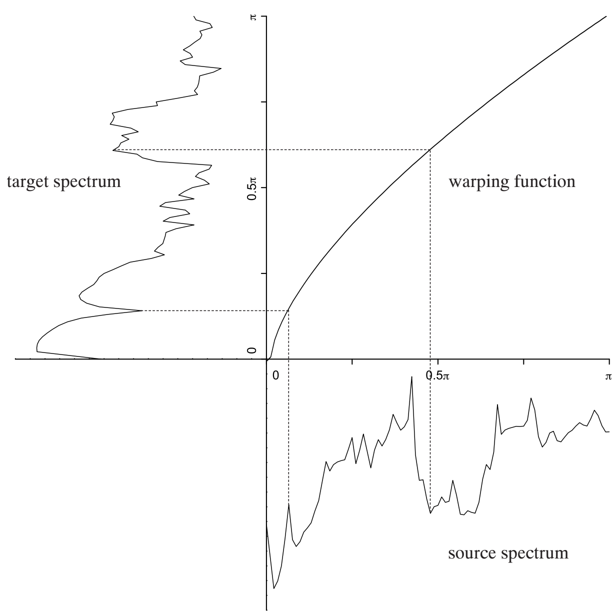

Fig. 3. Example of a spectrum warped by a bilinear warping function [2]

The values of the spectrum are then interpolated depending on the warped locations of the frequencies. This results in a warped spectrum, with some regions compressed and other regions stretched as diplayed in Figure 3 [2]. This operation is performed for every frame of the input data. After that, each frame is transformed back to time domain by using inverse FFT.

TABLE I
WARPING FUNCTIONS

Type	**Formula**
Symmetric	$g(\omega,\alpha) = \begin{cases} \alpha\omega, & \omega \leq \omega_0 \\ \alpha\omega_0 + \frac{\pi-\alpha\omega_0}{\pi-\omega_0}(\omega-\omega_0), & \omega > \omega_0 \end{cases}$ $\omega_0 = \begin{cases} \frac{7\pi}{8}, & \alpha \leq 1 \\ \frac{7\pi}{8\alpha}, & \alpha > 1 \end{cases}$
Asymmetric	$g(\omega,\alpha) = \begin{cases} \alpha\omega, & \omega \leq \omega_0 \\ \alpha\omega_0 + \frac{\pi-\alpha\omega_0}{\pi-\omega_0}(\omega-\omega_0), & \omega > \omega_0 \end{cases}$ $\omega_0 = \frac{7\pi}{8}$
Quadratic	$g(\omega,\alpha) = \omega + \alpha\left(\left(\frac{\omega}{\pi}\right) - \left(\frac{\omega}{\pi}\right)^2\right)$
Power	$g(\omega,\alpha) = \pi\left(\frac{\omega}{\pi}\right)^\alpha$
Bilinear	$g(\omega,\alpha) = \left\lvert -i\frac{z-\alpha}{1-\alpha z}\right\rvert$ $z = e^{i\omega}$

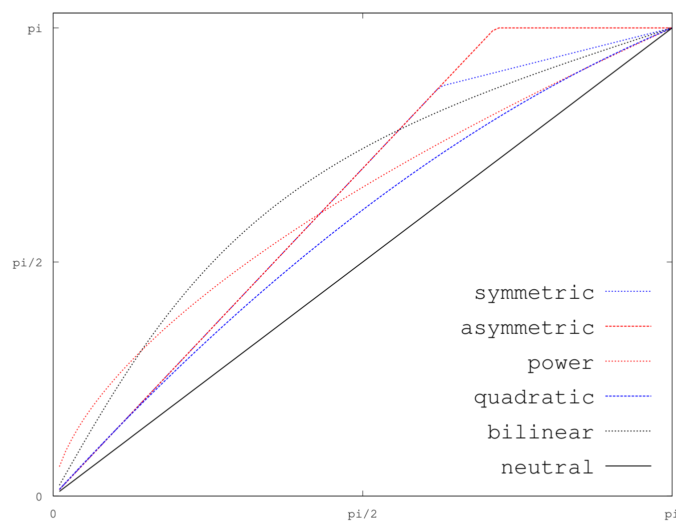

Fig. 4. Warping functions. The warping factor α is 0.4 for the bilinear and 0.6 for the power function and 1.4 in all other cases.

D. PSOLA

The warped frames have to be concatenated to create an output file. Simply concatenating the frames often results in sudden signal jumps causing poor speech quality. Therefore, pitch-synchronous overlap and add (PSOLA) [13] is used. The frames are multiplied by a Hamming window [14] before concatenating. Then, frames are overlapped and added as displayed in Figure 5. As the amount of overlap influences the periodicity of the output signal, PSOLA also serves as means of modifying the signal's fundamental frequency. Also the duration of the signal can be altered by repeating or skipping frames.

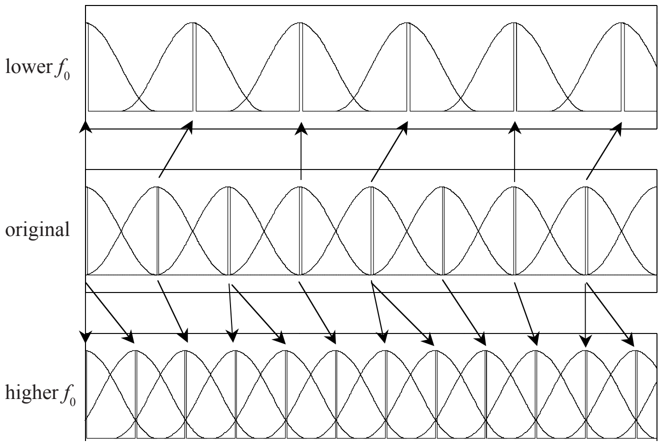

Fig. 5. Hamming windows over frames: Frames are inserted or skipped in order to increase or decrease the pitch [2]

To perform VTLN-based voice conversion including PSOLA using the Voice Conversion Octave Toolbox and pitch marks generated by the Praat program, type:

```
praatcon.exe PraatToPitchMarks.praat
data/sample.wav data/pitchMarks.pm
perl convertPraatToMatlab.pl
data/pitchMarks.pm
data/pitchMarksMat.txt  alpha=1.2;
fRatio=1.2; vc('data/sample.wav',
alpha, fRatio, 'data/output.wav',
'data/PitchMarksMat.txt')
```

If you want to use the integrated pitch tracker, simply use the following command:

```
alpha=1.2; fRatio=1.2;
vc('data/sample.wav', alpha, fRatio,
'data/output.wav')
```

Here, `sample.wav` is the input speech file to be converted. `PitchMarksMat.txt` contains the pitch marks obtained after running Praat and the Perl script as discussed in Section III-B. `alpha` is the warping factor and `fRatio` the fundamental frequency ratio used by PSOLA. The converted speech is stored in `output.wav`. By default, the command uses the symmetric warping function. To choose another warping function, change the corresponding parameter in `vc.m`.

IV. Conclusion

We presented the first version of the open-source Voice Conversion Octave Toolbox. The toolbox's functionality is currently limited to the popular frequency warping technique (aka VTLN-based voice conversion). Five different warping functions are available. This toolbox is the first step of providing an open-source, easy-to-use voice conversion framework to the community. We highly encourage fellow researchers to support the Voice Conversion Octave Toolbox by improving the existing algorithms or adding new modules to the open-source platform. In particular, we plan to implement the following techniques in future versions:

- GMM-based voice conversion (with global variance),
- residual prediction,
- support for jitter and shimmer manipulation.

Additionally, the authors plan to conduct a subjective evaluation of the toolbox using the evaluation metrics described in [2] to compare the toolbox's speech quality and conversion performance to the state of the art in the field.

References

[1] D. Sündermann, "Voice conversion: state-of-the-art and future work," in *Proc. of the DAGA*, 2005.

[2] ——, "Text-independent voice conversion," Ph.D. dissertation, Universität der Bundeswehr München, 2008.

[3] Y. Stylianou and O. Cappé, "A system for voice conversion based on probabilistic classification and a harmonic plus noise model," in *Proc. of the ICASSP*, 1998.

[4] D. Sündermann, A. Bonafonte, H. Ney, and H. Höge, "Time domain vocal tract length normalization," in *Proc. of the ISSPIT*, 2004.

[5] D. Sündermann, A. Bonafonte, and H. Ney, "A study on residual prediction techniques for voice conversion," in *Proc. of the ICASSP*, 2005.

[6] H. Benisty, D. Malah, and K. Crammer, "Modular global variance enhancement for voice conversion systems," in *Proc. of the EUSIPCO*, 2012.

[7] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, 2010.

[8] D. Sündermann and H. Ney, "VTLN-based voice conversion," in *Proc. of the ISSPIT*, 2003.

[9] P. Boersma, *Praat, a System for Doing Phonetics by Computer*. Glot International, 2001, vol. 5, no. 9/10.

[10] L. Docio-Fernandez and C. Garcia-Mateo, *Encyclopedia of Biometrics*. Springer, 2009.

[11] B. Kotnik, H. Hoege, and Z. Kacic, "Evaluation of pitch detection algorithms in adverse conditions," in *Proc. of the Speech Prosody*, 2006.

[12] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sample sound," *Proceedings of the Institute of Phonetic Sciences, Amsterdam*, 1993.

[13] H. Valbret, E. Moulines, and J. Tubach, "Voice transformation using PSOLA technique," in *Proc. of the ICASSP*, 1992.

[14] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, 1978.