WebVoice: A Toolkit for Perceptual Insights into Speech Processing

Thilo Stadelmann, Steffen Heinzl, Markus Unterberger, and Bernd Freisleben
Department of Mathematics & Computer Science, University of Marburg
Hans-Meerwein-Str. 3, D-35032 Marburg, Germany
e-mail: {stadelmann, heinzl, unterberger, freisleb}@informatik.uni-marburg.de

Abstract

Feature extraction and modeling techniques for speech processing are often complex. Understanding a new technique theoretically can be difficult for a novice, just as it is difficult for a practitioner to find the best parameter settings and/or combination of methods for a new task or data. In this paper, a novel approach and a corresponding software toolkit for facilitating both education and experimentation in speech processing is presented: listening to the results of feature extraction and modeling is made possible via resynthesis of intermediate pattern recognition results. The software is made publicly available as a web service called `WebVoice` *with accompanying user interfaces for ease of use.*

1. Introduction

The automatic processing of speech is a classical pattern recognition task: to detect speech, infer voicing, identify a speaker or recognize what has been said, the audio waveform is first converted into a sequence of feature vectors. Then, statistical models using these feature vectors are built for recognition. Typical features are Mel Frequency Cepstral Coefficients (MFCC), linear prediction-based features and pitch, modeled by Gaussian Mixture Models (GMM) or Hidden Markov Models (HMM) [11].

The techniques involved in extracting these features and building models are quite complex: the inner mechanisms are sophisticated and typically require a large number of parameters. These have to be tuned to the application's needs and to the data at hand. Additionally, several alternative techniques compete for each stage. When the method itself and its parameterization is fixed, the choice of the concrete implementation may offer variability, too [5][17]. This makes it difficult to successfully apply common speech features and models in contexts where there is no prior experience on how to make these choices. Experience may be lacking in several ways:

Experimentation: new areas of application and new data in new conditions render commonly used best practice rules useless. Nothing works out of the box. Where is the problem with the current setting and what is a promising way to solve it? Researchers and developers need tools showing them what area is affected.

Education: novices are overwhelmed by the many degrees of freedom introduced by the methods. The meaning of each possibility is not easily comprehensible. What do the various options represent and how do they interact? Students need tools making the abstract methods vivid and their exploration exciting.

In this paper, the practical benefit *resynthesis* might bring to research, development, and teaching in the complete process of speech processing is emphasized: hearing what is going on in the pattern recognition chain is likely to give insights into speech processing. For this purpose, existing approaches are extended to cover the complete speech processing chain with a broadened scope, and a corresponding software toolkit is offered. In particular, we present the `WebVoice` web service, a publicly available software tool accessible with a web browser. Two clients are offered to guarantee ease-of-use and ready availability without performing complex installations or writing additional software.

This paper is organized as follows: Section 2 motivates our approach and puts it in the context of related work. In Section 3 and Section 4, it is shown how the signal is resynthesized and the software is deployed, respectively. Section 5 illustrates the components of the software with some use cases. Section 6 concludes the paper and outlines areas of future research.

2. Analysis by Perception

To get a quick and intuitive insight into speech and speaker related features and models, one often wishes to be able to virtually *hear* what is grasped by the features and models—after all they represent *acoustic*

978-1-4244-4131-0/09/$25.00 ©2009 IEEE

events. This can be achieved by *re*-synthesizing the intermediate results of the speech processing system to give a direct, sensible feedback on how specific choices affect the final outcome. If, for example, the difference between subtle variations of the MFCC filter bank can hardly be heard, it is probably not the first area to address in searching for improved speaker recognition results. The human auditory system aids thorough technical experimentation and evaluation of methods and settings. The auditory feedback can further be enhanced visually by looking at graphical representations (e.g., spectrograms or speech flakes [13]) of a resynthesized signal, together yielding a toolbox of multi-sensory perceptual analysis instruments. *Analysis by perception* thus works as a guide in the large, unstructured hypothesis space of speech processing methods.

Resynthesis based on voice features aims at restoring formerly uttered speech. The used techniques are those from speech synthesis, but the aim is to revert a previous analysis process rather than synthesizing something purely artificial. This is reflected several times in the literature: Milner and Shao [12] introduce a system designed for distributed automatic speech recognition (ASR) over mobile networks that outputs the restored speech. Demuynck et al. [2] also work in the ASR domain, but focus on the aspect of analyzing what preprocessed MFCCs do and do not represent in order to improve feature extraction. In this case, truthful exhibition of the information loss in each step is important. Ellis [3] provides Matlab routines for MFCC- and perceptual linear prediction (PLP) feature inversion to advocate playful preoccupation with speech recognition techniques. Aucouturier [1] uses resynthesis to gain insights into the differences of human- and machine perception of music.

To be useful for a wide range of users - whether they are researchers, practitioners or (lifelong) learners - a toolkit based on resynthesis techniques is needed that is accessible, easy to use and delivers quick results as well as useful insights. Setting up the tool and learning to operate it must not introduce an additional barrier. Furthermore, the scope of such a tool should be broader than just ASR or speech coding [10] in order to comprise all the facets of speech processing. We offer both within the `WebVoice` web service and its underlying technologies.

3. Resynthesis

The aim of resynthesis within this paper is to exhibit what each voice processing stage does with the signal. The important stages are feature extraction (including signal preprocessing) and modeling. We focus on the most widespread techniques, namely MFCC, Linear Prediction Coefficients (LPC) and pitch as features, and GMM and HMM as models. The primary requirement for a resynthesized signal is to make audible what is contained in information models and feature vectors. Thus, instead of making the result more intelligible or natural, we even omit to reverse the effect of some intermediate steps such as a preemphasis filter.

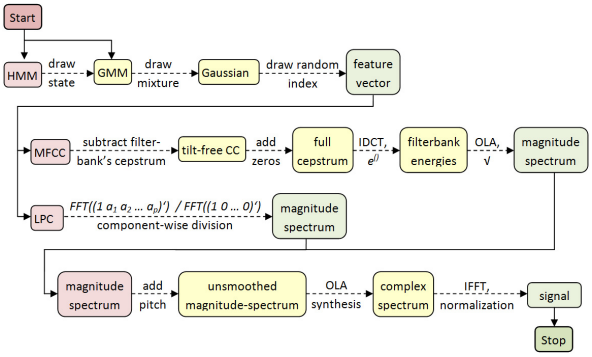


Figure 1. Flow diagram of the model inversion process.

The inversion process from a voice model back to audio is depicted in Figure 1. A GMM represents a probability distribution. Feature vectors following the distribution can be obtained via sampling from the model. This is a two-stage process: First, a mixture component is chosen at random according to the distribution determined by the mixture weights. Second, a normal deviate is drawn from this mixture component via, e.g., the polar (Box-Muller) method. If the GMM is just a state model inside a HMM, a zeroth stage has to be introduced that determines the GMM from which to sample according to the HMM's current state and state transition matrix. The state transition is randomized the same way a mixture component is chosen.

Depending on the actual feature type(s), the inversion process continues: Converting an MFCC vector back to a waveform means to first cancel out the spectral tilt introduced by the (mel) filter bank. This is done by subtracting this filters' cepstrum in the cepstral domain. The circumsized cepstrum is then filled up with zeros and transformed back to the log filter bank domain by the Inverse Discrete Cosine Transform (IDCT), where the $\log()$ operation is reversed. This yields a vector of the size of the filterbank, which is expanded to spectrum size via an overlap-and-add method. Taking the square root of each resulting component yields a standard magnitude spectrum. It lacks most of the pitch information that is removed by the heavy cepstral smoothing during feature extraction. More details on these steps can be found in the works of Ellis [3] and Milner et al. [12].

A LPC vector is converted to a magnitude spectrum by first prefixing it with the zeroth coefficient (1.0) that has been discarded during feature extraction. The new vector is then regarded as some signal and Fourier transformed to yield a complex spectrum via the Fast Fourier Transform (FFT). Dividing the complex spectrum of an impulse [15] by this spectrum yields the frequency response of the LPC filter [14].

Superimposing pitch (extracted via the RAPT algorithm [16] in our system) on these MFCC or LPC spectra is done via the following scheme: the spectral envelope is amplified/de-amplified up to 25% according to the distance of each frequency bin to the next harmonic of F_0:

$$\hat{s}_i = s_i \cdot \left(1.25 - \left|\left\lfloor \frac{f_i}{F_0} + 0.5 \right\rfloor - \frac{f_i}{F_0}\right|\right) \qquad (1)$$

where s_i is the ith component of a magnitude spectrum vector, f_i is the corresponding center frequency and F_0 is the extracted pitch in Hz, respectively. This introduces a repeated pattern of rise and descent and evokes a sensation of pitch. It works fine for MFCCs, but introduces considerable musical noise upon LPC spectra after phase reconstruction. How to deal with this has been exemplified by Goh et al. [6], but is not further considered here.

The missing phase spectrum has to be estimated from the information present in the overlapping of frames. For this purpose, the iterative method introduced by Griffin and Lim [7] is used. The process is stopped when the average (across frequency bins) absolute difference (error) between two successive iterations of the magnitude spectrum is less than 4% of the average magnitude in the current spectrum (or 100 iterations are reached, whatever happens first). The final signal is obtained by applying the Inverse FFT (IFFT) to the complete sequence of complex spectra. It is normalized to have 30% of the maximum amplitude at its biggest peak. If pitch is the only extracted feature, the resynthesized signal is directly assembled in the time domain by a smooth composition of sinusoids at the frequency of F_0.

The presented approach differs in several aspects from existing approaches: by including the modeling stage and shifting the focus away from pure ASR methods, a more complete solution is offered. Our novel way of reintroducing pitch into smoothed spectra is both methodically and computationally simple and effective.

4. Implementation

The software was implemented as a C++ library for Windows and Unix-based systems. A library has several drawbacks with respect to deployment, such as the need of a potential user to work with the source code, to adopt it to a specific platform, and to build a complex software package from scratch. This is certainly not beneficial for the application of a tool that is intended to *ease* work.

We therefore offer our resynthesis toolkit as a web service called `WebVoice`, accompanied by an automatic user interface (UI) generator that works as a plug-in for the Firefox web browser: the Web Service Browser [9]. This makes using the tool as easy as browsing to the service's URI after installing the browser plug-in with a single click. The service-oriented approach offers several advantages:

- Invocation is platform independent and does not need any installation.
- Software runs on the server side, using the computational power of the remote machine (possibly a cluster).
- Source code does not need to be released (might be prohibited by organizational policies and licenses).
- Updates are directly available to everybody.

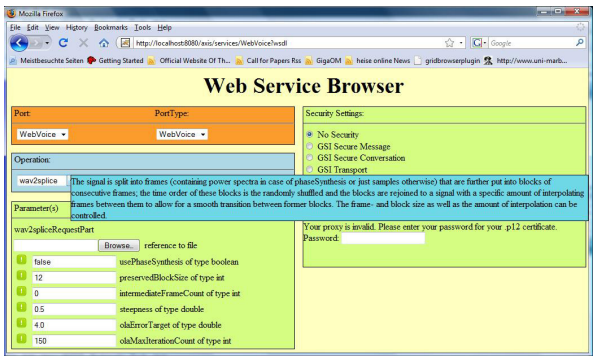


Figure 2. `WebVoice` in the Web Service Browser.

Figure 2 shows the UI for one of `WebVoice`'s operations. It is automatically generated from the web service's WSDL description. It offers drop-down boxes to select among the available methods and helps to input different data types for the parameters, e.g., to perform a file upload in the service invocation step. Data transfer is handled efficiently by the Flex-SwA framework [8] using a communication policy that allows to describe bulk data transfer protocols. After selecting an operation, loading up a (16KHz, 16bit, mono) wav file and possibly tailoring other parameters to a user's needs, the web service is invoked by simply clicking a button. When the computation on the server has finished, the result page opens with a MIME type representation of the result (a media player plays back the audio file in our case), offering to download it and also showing a textual representation and the SOAP message.

As an alternative to the Firefox Add-on, we also provide a Flash-based Rich Internet Application that

works with a larger set of browsers, namely the Service-enabled Mashup Editor able to invoke the `WebVoice` service. Figure 3 presents a screenshot of the mashup editor showing the UI of the `WebVoice` service.

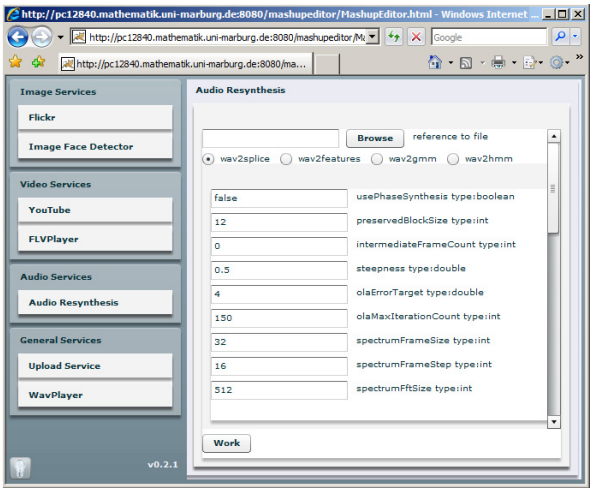


Figure 3. `WebVoice` in the mashup editor.

The `WebVoice` endpoint (URI), the Web Service Browser Firefox plug-in, and the mashup editor are publicly available under http://mage.uni-marburg.de/. Updates and further developments will be announced there, too.

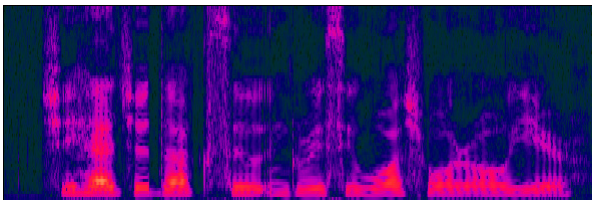
5. WebVoice Usage

Four methods as part of our web service are offered to accomplish the goal of making the functionality and content of features and models audible. Each method's detailed description is shipped with `WebVoice` and contains explanations and sound default values for each parameter. We give an overview here:

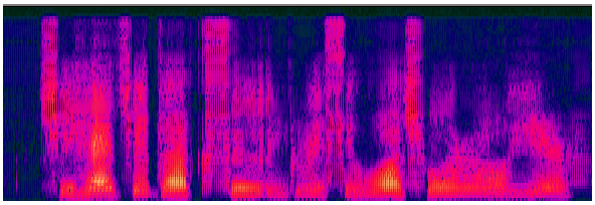
`wav2splice` offers to listen to a spliced version of the original signal. Splicing is the operation of randomizing the time order of subsequent non-overlapping blocks of the signal, where the block size can be specified. Listening to a spliced signal offers insights into the way pattern recognition systems handle data as a bag of frames. Sigmoidal interpolation between successive blocks can be switched on and controlled via some parameters to allow for smooth transitions. Additionally, the user can choose to hear or not hear the effect of phase spectrum reestimation in order to discern the influence it has on the results of the following methods.

`wav2features` goes one step further in the pattern recognition chain and offers to analyze various feature extraction methods: MFCCs and LPCs, both possibly accompanied by pitch, or pitch alone. The effects of virtually all possible parameters and the difference between the (combination of) techniques can be observed.

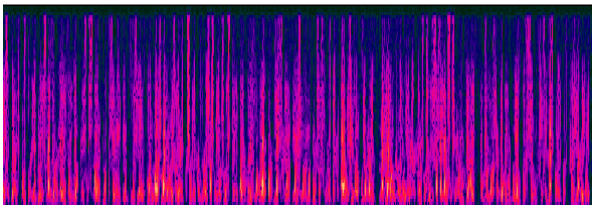
`wav2gmm` provides GMM inversion using all the features (and respective parameter settings) known from above. The user has control over all modeling parameters. The same is true for `wav2hmm` that contributes the same service for continuous density HMMs of any internal structure.



(a) Spectrogram of the original signal.



(b) Spectrogram of the resynthesized features.



(c) Spectrogram of the resynthesized GMM.

Figure 4. Spectrogram comparison.

Figure 4(a) shows the spectrogram of a 3 seconds long sentence from the TIMIT database [4] recorded under noise-free studio conditions and sampled at 16 kHz. When resynthesized using `wav2features` and a standard parameterization (i.e. MFCCs and pitch), the speech is still intelligible and the voice discernible from other resynthesized voices, as confirmed in listening experiments. It sounds, however, like a "robot voice", having lost much of its natural appearance. The corresponding spectrogram is depicted in Figure 4(b). This is how MFCCs sound like.

The same signal resynthesized from a GMM (again using standard parameters and MFCC+pitch features) sounds like "boiling water". This is due to the fact that the individual frames — although strictly obeying the original speaker's frequency distribution — are completely independent of each other. This results in a sound that is not perceived as a voice by human listeners (although it still contains features for voice comparison even for humans). Apart from listening to this signal, its long term spectral analysis can reveal interesting

details about the model at hand — i.e. what frequency characteristics are captured by the distribution. The spectrogram for the resynthesized model is shown in Figure 4(c).

6. Conclusions

In this paper, a novel method and a corresponding software toolkit has been presented to quickly understand what effect choice and parameter setting of various feature extraction and -modeling techniques in speech processing has on the intended outcome. This both guides the search for optimized parameters in the presence of new data and tasks, and it eases the first contact with these methods, i.e. educational learning. The ease of use of the web service `WebVoice` and its playful character invites a user to a deeper engagement with the important topic of parameter tuning and method selection. This will likely shorten the time needed to become familiar with the discussed pattern recognition techniques, and it will probably enhance the range of tasks to which they can readily be applied.

An area of future work is the flexible composition of techniques for resynthesis. Our current implementation allows fast and easy testing of what the majority of users may need. But to offer even more flexibility in the light of tasks yet to be developed, it is beneficial to be able to plug modules together in the way a software library could be used — but with the usability of a graphical user interface and with the benefits of web service technology. For this purpose, the mashup editor is currently being extended.

Acknowledgements

The authors thank C. Gregor van den Boogaart for an encouraging discussion. This work is supported by the Deutsche Forschungsgemeinschaft (SFB/FK 615).

References

[1] J.-J. Aucouturier. A Day in the Life of a Gaussian Mixture Model: Informing Music Pattern Recognition with Psychological Experiments. *Journal of New Music Research*, submitted, 2009.

[2] K. Demuynck, O. Garcia, and D. Van Compernolle. Synthesizing Speech from Speech Recognition Parameters. In *Proc. International Conference on Spoken Language Processing, Jeju Island, Korea*, volume II, pages 945–948, 2004.

[3] D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/, 2005. online web resource.

[4] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. The DARPA Speech Recognition Research Database: Specification and Status. In *Proceedings of the DARPA Speech Recognition Workshop, Report No. SAIC-86/1546, February 1986, Palo-Alto*, 1986.

[5] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task. In *Proc. of the ICTAI2007*, pages 510–513, 2005.

[6] Z. Goh, K.-C. Tan, , and B. T. G. Tan. Postprocessing Method for Suppressing Musical Noise Generated by Spectral Subtraction. *IEEE Transactions on Speech and Audio Processing*, 6:287–292, 1998.

[7] D. W. Griffin and J. S. Lim. Signal Estimation from Modified Short-Time Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32:236–243, 1984.

[8] S. Heinzl, M. Mathes, T. Friese, M. Smith, and B. Freisleben. Flex-SwA: Flexible Exchange of Binary Data Based on SOAP Messages with Attachments. In *Proc. of the IEEE International Conference on Web Services, Chicago, USA*, pages 3–10. IEEE Press, 2006.

[9] S. Heinzl, M. Mathes, T. Stadelmann, D. Seiler, M. Diegelmann, H. Dohmann, and B. Freisleben. The Web Service Browser: Automatic Client Generation and Efficient Data Transfer for Web Services. In *Proc. of the 7th IEEE International Conference on Web Services (ICWS 2009)*, page (accepted for publication). IEEE Press, 2009.

[10] W. Kleijn and K. Paliwal. *Speech coding and synthesis*. Elsevier Science Inc. New York, NY, USA, 1995.

[11] M. Kotti, V. Moschou, and C. Kotropoulos. Speaker Segmentation and Clustering. *Signal Processing*, 88:1091–1124, 2008.

[12] B. Milner and X. Shao. Clean Speech Reconstruction from MFCC Vectors and Fundamental Frequency using an Integrated Front-End. *Speech Communication*, 48:697–715, 2006.

[13] C. A. Pickover. On the Use of Symmetrized Dot Patterns for the Visual Characterization of Speech Waveforms and Other Sampled Data. *Journal of the Acoustic Society of America*, 80:955–960, 1986.

[14] L. R. Rabiner and B.-S. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, USA, 1993.

[15] S. W. Smith. *Digital Signal Processing - A Practical Guide for Engineers and Scientists*. Newnes, USA, 2003.

[16] D. Talkin. A Robust Algorithm for Pitch Tracking (RAPT). In W. B. Klejin and K. K. Paliwal, editors, *Speech Coding and Synthesis*, chapter 3, pages 495–518. Elsevier Science, Amsterdam, NL, 1995.

[17] F. Zheng, G. Zhang, and Z. Song. Comparison of Different Implementations of MFCC. *Journal of Computer Science and Technology*, 16:582–589, 2001.