SYNTHESIZING SPEECH FROM SPEECH RECOGNITION PARAMETERS

Kris Demuynck, Oscar Garcia, Dirk Van Compernolle

Katholieke Universiteit Leuven – Dept. ESAT
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
{Kris.Demuynck,Oscar.Garcia,Dirk.VanCompernolle}@esat.kuleuven.ac.be

Abstract

The merits of different signal preprocessing schemes for speech recognizers are usually assessed purely on the basis of the resulting recognition accuracy. Such benchmarks give a good indication as to whether one preprocessing is better than another, but little knowledge is acquired about why it is better or how it could be further improved. In order to gain more insight in the preprocessing, we seek to re-synthesize speech from speech recognition features. This way, we are able to pinpoint some deficiencies in our current preprocessing scheme. Additional analysis of successful new preprocessing schemes may allow us one day to identify precisely those properties that are desirable in a feature set. Next to these purely scientific aims, the re-synthesis of speech from recognition features is of interest to thin-client speech applications, and as an alternative to the classical LPC source-filter model for speech manipulation.

1. Introduction

In this paper we seek to reverse the signal processing done in current automatic speech recognition (ASR) systems, i.e. we want to re-synthesize speech from the feature vectors used internally by the speech recognizer.

Our interest in this operation is fourfold. First, there is the pure scientific question of determining whether all necessary information to understand the speech is still present in the feature vector. Any information lost in this very first step of the speech recognition process will inherently cripple the final recognition accuracy. Secondly, we want to investigate whether certain operations, e.g. vocal tract length normalization (VTLN), really have the effect attributed to them. A third application of this research is more practical. In thin clients [1], the mobile appliance converts the speech only into a compact (low bandwidth) set of parameters, while a dedicated (non mobile) server does the actual recognition of the spoken queries and then sends back the requested information. Being able to re-synthesize the speech in case a human operator is needed to complete the query would be very convenient. Finally, our approach may provide an alternative to the classical LPC source-filter model for speech manipulation. The signal processing in speech recognizers more or less extracts the following three independent components of a speech sound: the excitation signal (voiced or unvoiced + pitch), the energy and the vocal tract configuration. Given this decomposition and some re-synthesis algorithm, it is easy to reconstruct the speech signal with one or more properties changed.

Although the proposed work is related to HMM-based speech synthesis [2] and cepstral-vocoders, there are some essential differences. In (HMM-based) speech synthesis and vocoders the main goal is high quality speech, and as such, features are chosen that are well suited for resynthesis. Our aim, on the other hand, is to analyze existing preprocessing schemes for ASR with respect to information loss. This means we want to be able to inverse most, if not all, prevalent preprocessing schemes. This results in different techniques being used for the resynthesis. Furthermore, for our work, the resynthesis techniques are only the means to an end. The focus is on the information content in ASR features.

Figure 1 gives a high-level overview of the operations involved in converting the speech signal into features suitable for speech recognition. In each step, part of the information present in the original signal is lost. When reversing the process, the following difficulties have to be overcome: 1) estimation of good phase information based on the magnitude of the spectrum only, 2) handling of pitch and voicing information, and 3) reversing the smoothing operation. These steps will be separately described in the next three sections. In section 5 we combine the partial results into the final re-synthesis system. Section 6 investigates the information loss at each point in the preprocessing, and section 7 looks at the effects of some standard channel and speaker normalization operations. Finally, in section 8 we check to what extent the aims put forth in the introduction could be met, we summarize what we learned about the signal preprocessing used in speech recognizers, and we have a short look at future research.
2. RECONSTRUCTING THE PHASE INFORMATION

Figure 2 outlines a typical set of transformations modern speech recognizers perform on the incoming speech signal to convert it into a compact feature vector, which –hopefully– still contains all relevant information. For the explanation of the different steps in the resynthesis process we will closely follow this preprocessing scheme. However, the combined set of algorithms can cope with most prevalent alternatives to this scheme as well.

The first step that needs to be reversed is the time-frequency analysis, which results in a (magnitude) spectrogram. The only information lost in this process is the phase information.

To reconstruct a time signal from a real or modified magnitude spectrogram, we use the algorithm Griffin and Lim proposed in [3]. This algorithm iteratively decreases the squared error between the target magnitude spectrum $|Y_{wt}(kT, \omega)|$ and the magnitude spectrogram $|\hat{X}_{wt}(kT, \omega)|$ of the re-synthesized signal $x(t)$ ($t$ is the iteration index, $k$ the frame index, $T$ the frame shift, $\omega$ the frequency tab, $t$ the sample index, and $w$ the windowing function). At each iteration, the next estimate of the time signal $x^{i+1}(t)$ is constructed from the target magnitude spectrum $|Y_{wt}(kT, \omega)|$ combined with the phase spectrum of the previous estimate of the time signal $x_i(t)$. The reconstruction is done with a "least-squares" overlap-add technique:

$$w'(t) = \frac{w(t)}{\sum_{k=-\infty}^{\infty} w^2(t + kT)}$$

$$\hat{X}_{wt}(kT, \omega) = |Y_{wt}(kT, \omega)| \frac{X_{wt}(kT, \omega)}{|\hat{X}_{wt}(kT, \omega)|}$$

$$x^{i+1}(t) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} w'(kT - t) \int_{-\pi}^{\pi} \hat{X}_{wt}(kT, \omega) e^{j\omega t} d\omega$$

Note that regeneration of phase information is only possible due to the oversampling of the magnitude spectrum. See [3] for the convergence properties of the algorithm and for more details on the least-squares aspect of the overlap-add technique. When computational resources or latency are important, other algorithms such as the one proposed by Kang and Kim in [4] may be better suited. However, for our primary research interest –gaining insight in the speech recognition features– Griffin’s algorithm works excellent.

When reversing the speech preprocessing depicted in figure 2, we observed that the emphasis put on the high frequency part of the spectrum by the pre-emphasis step, is unwanted during the re-synthesis. Since Griffin’s algorithm minimizes the squared error between target and estimated spectrum, unnecessary effort will be put into minimizing the mismatch in the high frequency regions after pre-emphasis. We therefore lift the amplitude spectrum in advance to remove the spectral tilt introduced by the pre-emphasis.

3. HANDLING PITCH INFORMATION

Ignoring the vocal tract length normalization for the time being, the next steps in the preprocessing scheme (see figure 2) result in pitch removal and a coarse (smoothed) description of the shape of the spectral envelope. Although pitch removal and smoothing are not separated well defined blocks in the preprocessing, we will handle them as such for the re-synthesis.

A standard speech recognizer completely ignores the pitch frequency. However, for high-quality re-synthesis, this information is needed. Hence we extract two extra features from the speech signal: the pitch frequency $f_0(k)$ and the percentage of voicing $v(k)$ for each frame (see the central part of figure 2).

Based on these two values for each frame, an artificial excitation signal can be constructed as follows:

1. Do a linear interpolation to obtain a pitch frequency $f_0(t)$ and voicing fraction $v(t)$ at each sample point.
2. Make a pulse-train according to figure 3, i.e. the spacing between the pulses equals $1/f_0(t)$ and the amplitude equals $\sqrt{1/v(t)}$. This results in an average energy per sample point of 1.0.
3. Filter the pulse-train with a second-order low-pass filter and add white noise filtered with a complementary high-pass filter. The LP-filter can be seen as a simple model of the glottal pulse, while the HP-filter injects a small amount of (high frequency) noise in the pitch generation process. These alterations largely remove the metallic sound associated with a pure pulse-train as excitation signal. For a 16kHz sampling frequency, following filter coefficients are used:

$$LP(z) = \frac{0.3793 + 0.4765z^{-1} + 0.1753z^{-2}}{1 - 0.1776z^{-1} + 0.2093z^{-2}}$$
$$HP(z) = \frac{0.3456 - 0.6244z^{-1} + 0.3454z^{-2}}{1 - 0.1063z^{-1} + 0.2118z^{-2}}$$

4. Make a linear combination of the voiced excitation signal (step 2 & 3) and white noise (the unvoiced excitation signal) using the voicing fraction $v(t)$ as weighting coefficients.
improves the clearness of the reconstructed speech markedly (monics.

4. REVERSING THE SPECTRAL SMOOTHING

After applying the MEL-scaled triangular filterbanks, logarithm, and the truncated inverse discrete cosine transform (cepstral transformation), a coarse (mainly smoothed) description of the shape of the spectral envelope is obtained. In order to transform the resulting 13 MEL-scaled cepstral coefficients back to a complete spectral envelope for each frame, an inverse transform was estimated on the Resource Management (RM) database (3h 20' of speech).

In a first attempt, we calculated the linear transformation that reconstructs best –in least squares sense– the log amplitude spectrum. By working in the log domain, we ensure that the transformation only has to reverse the MEL-filterbanks and the cepstral transformation (both linear operations), and not the log compression. Note that the target spectrum still contains the pitch information. Figure 4 shows the original spectrogram and the spectrogram reconstructed from the cepstral coefficients for a female voice. As can be observed, the reconstructed spectrogram contains the first two pitch harmonics. This shows that, contrary to general belief, MEL-cepstral coefficients still contain some information concerning the pitch.

Since, during re-synthesis, the pitch information recreated by the above described transformation interferes with the artificial pitch signal constructed in section 3, an alternative inverse transform that does not recreate any pitch information is needed. The new transform is again estimated in least squares sense on the RM-database, but now we first remove all pitch information from the target spectrum by means of a cepstral transformation on the amplitude spectrum. We prefer working on the amplitude spectrum instead of on the (usual) log amplitude spectrum, since now the truncation operation removes a certain amount of spectral energy from the signal which is all the more natural than removing a certain amount of log energy. Instead of using a hard truncation, we also relied on a tapered window. This way, the pitch could be efficiently suppressed without introducing too much smoothing in the spectral envelope (see e.g. the central part of figure 1).

The re-synthesized speech based on the spectral envelope constructed as described above, sounds a bit duller than the original. This is due to the less pronounced (smared) formants in the reconstructed spectral envelope. Enhancing the formant structures by replacing each value of the spectral envelope \(|E_w(kT, \omega)|\) with

\[
|E_w(kT, \omega)| \left|\frac{E_w(kT, \omega)}{\max(|E_w(kT, \omega - \nu)||E_w(kT, \omega + \nu)|)}\right|
\]

improves the clearness of the reconstructed speech markedly (\(\nu\) set to 5 for 16kHz speech).

5. COMBINING PITCH AND SPECTRAL ENVELOPE

During the re-synthesis, the artificial pitch signal \(p(t)\) and the reconstructed spectral envelope \(|E_w(kT, \omega)|\) are combined to create an initial guess for the time signal \(x^0(t)\), and to create the target magnitude spectrogram \(|Y_w(kT, \omega)|\) for Griffin’s algorithm.

As target magnitude spectrogram, we use the dot product (element wise multiplication) of the reconstructed spectral envelope \(|E_w(kT, \omega)|\) with the magnitude spectrum of the artificial pitch signal \(|P_w(kT, \omega)|\). To create the initial time signal \(x^0(t)\), we rely on the source-filter model for speech generation: the artificial pitch signal \(p(t)\) is used as excitation signal for a set of time varying FIR-filters obtained from the spectral envelope \(|E_w(kT, \omega)|\) (which in fact more or less quantify the vocal tract). As FIR-filters, we use the minimum phase time signals corresponding to each spectral slice. As shown in figure 5, the minimum phase signals show the typical exponential decay in amplitude seen in most impulse response functions. So they are good candidates for the FIR-filtering.

Figure 6 shows the convergence rate of Griffin’s algorithm when starting with the above constructed initial signal and when starting with white noise as initial signal. Note that Griffin’s algorithm only uses the phase info, so using white noise as initial signal equals to starting with a random phase. Having a good initial guess for the phase speeds up the convergence of Griffin’s algorithm considerably.

6. BASIC EXPERIMENTS

All original and re-synthesized speech samples used in our experiments, with corresponding spectrograms, can be found at: http://www.esat.kuleuven.ac.be/~spch/research/Resynth/
In a first set of experiments, we investigated whether and where vital information concerning the intelligibility of the speech is lost in the preprocessing scheme depicted in figure 2. We therefore re-synthesized speech from 1) the amplitude spectrum, 2) the spectrum after removing the pitch using the technique described in section 4, 3) the MEL-filterbank outputs, and 4) the cepstral coefficients. After applying Griffin’s algorithm, the speech sounds a bit harsher. The degradation in quality when using the artificial pitch is negligible. Applying the MEL-filterbanks, and to a larger extent the cepstral transformation, degrades the clearness of the speech somewhat, but everything remains very intelligible. When using white noise as excitation signal (no pitch information), everything sounds as whispered speech, but remains intelligible.

The preprocessing used in modern speech recognizers thus succeeds in concentrating most, if not all, relevant information in a compact set of features. Augmenting the feature set with phase information [5], seems unnecessary for clean speech. However, we expect that the reference preprocessing will prove to be less optimal for noisy speech. Yet, the handling of noisy speech falls outside the scope of this paper.

7. SPEAKER NORMALIZATION

In a second set of experiments, we looked at the effect of vocal tract length normalization (VTLN). The frequency axis is scaled linearly. The gap in the spectral envelope at the high frequencies, which occurs when handling female voices, is filled with the average of the magnitude spectrum pattern as seen in the four highest frequency tabs available. To map male or female voices to neutral voices, the frequency axis is expanded or compressed 7% respectively, and the pitch frequency $f_0$ is scaled in order to obtain an average $f_0$ of 160Hz. We also tried changing the gender by overcompensating, i.e. applying a 14% expansion or compression of the frequency axis and 210Hz or 110Hz as average $f_0$.

When listening to a gender neutral variant or to an artificially created male or female voice, it is still possible to detect the gender on other characteristics of the speaker such as prosody and rhythm. However, when removing the pitch in the gender neutral voices, it becomes very difficult to discern between males and females.

These experiments show that the proposed re-synthesis technique provides the same flexibility in modifying voices as the LPC source-filter model. The effect of VTLN was also clearly demonstrated.

8. CONCLUSIONS & FUTURE RESEARCH

Recapitulating the aims and applications put forth in the introduction, we see that it is possible to re-synthesize very intelligible speech from speech recognition features. The re-synthesized speech is not studio-quality, but is of sufficient quality for low- and medium-end applications. For high-quality voice modifications, algorithms based on time-domain signal processing such as WSOLA [6] still provide a better quality but with less flexibility.

Concerning the MEL-cepstra features, we observed little information loss when handling noise free speech. However, some interventions were needed to produce optimal speech quality when re-synthesizing. First of all, the pitch sensitivity of cepstral coefficients—especially with female voices—had to be dealt with. Hence, understanding how to cope with pitch and deriving new preprocessing schemes which suppress pitch more efficiently, may prove beneficial [7, 8, 9]. This does not mean that pitch provides no information at all. The re-synthesized speech without pitch is, similar to whispered speech, harder to understand. But instead of having remnants of the pitch “contaminating” the pure acoustic feature set, handling the pitch as a separate information stream (e.g. as proposed in [10]), is most likely a better approach. As to the loss of phase information and the (hyper) smoothing provided by the cepstra, both effects were to a large extent recoverable; so implicitly the necessary information is still present in the features.

Future research is still needed for noisy speech and for concurrent speakers. Also the analysis of new promising preprocessing schemes, e.g. [9], may provide additional insights, allowing us to better identify which properties are desirable in a feature set for speech recognizers. And finally, one may also take the next step and extract intelligible speech from the acoustic models now used for large vocabulary recognition (cf. [2]), trying to pin-point deficiencies there as well.

9. ACKNOWLEDGMENTS

The research reported in this paper was funded by IWT in the GBOU program, project FLaVoR (Project nr. 020192). http://www.esat.kuleuven.ac.be/~spch/projects/FLaVoR.

10. References


