

Voice Modification for Applications in Speech Synthesis

Juergen Schroeter, AT&T Labs – Research

Abstract

A significant part of the work required to create a high quality speech synthesizer is the creation of “synthetic voices”. Reusing an existing voice database and making it sound like a different speaker, or like the same speaker in a different emotional state, or using a different speaking style, is obviously important for increasing the efficiency in creating voice options for a synthesizer. This article reviews techniques to change signal characteristics like pitch and durations, and also spectral modifications. We conclude by assessing the prospects of voice modification in speech synthesis in light of now-available advanced machine learning techniques.

1. Introduction

Over the last decade, speech synthesis [16] has come a long way towards producing higher quality, more natural-sounding output. The two most important reasons for this paradigm shift are: 1) computers are now powerful enough to employ sophisticated data-driven techniques such as unit selection (e.g., [9]), and 2) better software tools are now available that allow for faster and cost-effective creation of new TTS voices [18].

The advent of high quality text-to-speech (TTS) may have created the false notion of speech synthesis being a “solved problem”, that is, the idea that speech synthesis can replace a live human speaker (or a speaker’s recording) in any application, service, or product. This is definitely not the case, given the enormous richness and expressive capabilities of the human voice that are impossible, or at least impractical, to match with a speech synthesizer. What unit selection speech synthesis can do, however, is deliver surprisingly good quality speech for somewhat narrow applications, such as travel reservations, weather reports, etc. [24]. This high quality is achieved by recording special domain voice databases. For a given domain (e.g., “travel”), voice talents are being recorded while reading examples from that domain, such as “Your flight to <destination> has been confirmed.” The idea is to cover as much material as possible that is well suited for the given application. Also, the reading style used (e.g., friendly but affirmative) has to be appropriate for the application. What unit-selection TTS cannot do today (at least not in any practical way) is to turn an average voice, reading in a “newsreader” (i.e., reserved, toned-down) style, into a highly desirable “spokesperson” voice for marketing a new product (i.e., speaking in a highly expressive, enthusiastic style). The reason for this inability is simple: there is no way that all the necessary speech data (many hundreds of hours) can be recorded from one single speaker, given the time it would take and the fact that a speaker’s voice might change over time. Consequently, modifying recorded speech by signal processing is the only practical option.

This chapter is organized as follows. In section 2, we will review some of the algorithms available for modifying speech signals in pitch, durations, and other characteristics. Section 3 will highlight methods available for modifying voice characteristics with the goal of either changing speaking style or even speaker identity. Finally, in section 4, we conclude with a summary and an outlook at future developments.

2. Voice Signal Modification in Concatenative Speech Synthesis

Concatenative speech synthesis uses snippets (units) of recorded speech, usually cut from full sentences. Commonly employed units are diphones (bracketing exactly one phone-to-phone transition, starting from the spectrally stable middle region of one phone to the spectrally stable middle region of the next phone), or demisyllables (comprising consonants and vowels). At synthesis time, the unit inventory (voice database) is searched for the optimal sequence of units that make up the desired speech output.

Until about a decade ago, typical concatenative speech synthesizers for English used inventories of between 1,500 and 3,000 units total. Besides special cases such as /r/-colorization of neighboring vowels¹, mostly single prototypical instances for any given kind of unit were stored in the inventory, predominantly for reasons of storage restrictions and synthesis speed. Clearly, having just one or, at most, very few candidates to choose from for a specific time slot necessitates modification of the unit by means of signal processing (e.g., in terms of pitch, duration, and amplitude) in order to “fit” it in with its neighbors. Note also that the initial selection of the one (or the few) prototypical examples of each class of units to be included in the inventory is a critical process that has much influence on the quality of the synthesized speech (see, e.g., Section 7.3.3 in [17]).

More recently, Unit Selection Synthesis (USEL) [2][4][9][21] moved the selection process from the off-line (synthesizer design) phase to the on-line (actual time of) synthesis when the sentence to be synthesized is known. USEL takes advantage of the more powerful computers available to store and search hundreds of thousands of units in total, narrowing down to at least a dozen or so potential candidates for each time slot. While the availability of several choices (e.g., with different pitch values) reduces (and sometimes eliminates) the need for signal processing, we will see that it is still needed because recording an inventory that covers all potential needs is highly impractical.

2.1 Necessity of voice signal modifications

In a keynote paper at the 1997 Eurospeech conference [22], Jan van Santen presented far-reaching statistical results that support the importance of voice signal modifications. For example, to cover all combinations of prosodic units in a random test set of sentences with probability 0.75, the inventory size needed to be 150,000 diphone units. He also showed that the similarity of test (i.e., actual usage) and training (unit inventory) text corpora in terms of triphone and vocabulary distributions have a significant impact on

¹ Colorization is the influence of the sound of interest (here, /r/) on neighboring sounds. For instance, formant frequencies of vowels immediately preceding or following an /r/ are influenced by it.

this kind of inventory coverage. He concluded that learning algorithms for training TTS systems (such as used in training unit selection paradigms) need to employ solid generalization capabilities in order to make up for the rather poor coverage of even very large training corpora. This means that the system can cope robustly with unit combinations that are needed to synthesize specific output text but that have not been part of the training set. Note that so far, the argumentation used mainly the combinatorics of language. However, another way to approach the problem would be to estimate total inventory size based on assumptions of token multiplicity in terms of variations in pitch and durations (amplitude variations left aside for now). Assuming, for illustration purposes only, 2,000 unit types with voiced speech and a multiplicity of 20 for pitch variations and another factor of 10 for durational variations, we arrive at $2,000 \times 20 \times 10 = 400,000$ unit tokens that would be needed. This calculation assumes no signal processing is used to change pitch and durations. Traditional diphone-based synthesizers may be considered as existence proofs that signal processing is helpful (here: reducing the inventory back down to 2,000 unit types), although it seems that the reduced inventory size surely would come with the cost of lost naturalness when compared to large inventory unit selection synthesizers.

One way to reduce unit inventory size significantly is to use smaller base units. Conkie [4] proposed using half-phones (about $2 \times 50 = 100$ types for English). A half-phone cuts a diphone at the phone boundary that it contains. Consequently, two half-phones can form a diphone. For the scenario discussed above, the statistical advantage of using half-phones over a full set of diphones is based on the fact that the number of necessary unit tokens in the inventory without using any signal processing would be $100 \times 20 \times 10 = 20,000$, which is a much more manageable set.

From these considerations, it should be clear that signal modifications in terms of pitch and durations should be useful for reducing the inventory size down to practical (i.e., recordable with one voice talent) levels. However, it is also clear that too much signal processing tends to decrease the quality of synthetic speech [10]. For pitch, the limit might be 10-20%, for durations, perhaps 50%, given available signal processing methods. Generally, the objective should be to maximize quality while balancing inventory size, amount of signal processing used, and, last but not least, choosing the specific algorithms that allow for relatively large signal modifications with no perceivable quality degradation.

2.2 Methods for modifying voice signals

One of the key issues in concatenative speech synthesis is picking the specific speech signal representation (parameterization) for efficiently storing the inventory units, smoothing over concatenation points, and for prosody modification. The question of which speech representation to use is tightly related to the question of which algorithm to use for prosody modifications. Therefore, the ideal method for modifying voice signals for speech synthesis purposes meets the following objectives:

1. It employs a speech representation that enables transparent (inaudible distortions) encoding/decoding. When signal characteristics are being

modified, the result has to sound convincingly like it has been produced by the recorded voice talent using different characteristics (e.g., pitch, duration).

2. The speech representation needs to allow random access to any indexed inventory unit without large computational costs for switching from one unit to another. For speech representations with memory (e.g., filters), this encompasses also storing algorithm-internal state variables, plus appropriately smoothing over the transition.
3. Although not a strict requirement, choosing a speech representation that adds compression has the practical advantages of a smaller inventory file size (i.e., more recorded speech per megabyte of disk storage) that makes distribution of voice inventories and TTS server maintenance easier.

In the following, we will highlight the basics of the three most commonly used methods: 1) linear prediction coefficients (LPC) and their role in capturing spectral envelopes in close correspondence to vocal tract geometry, 2) time-domain overlap add (TDSOLA) as a low complexity method for pitch and durational changes, and 3) sinusoidal speech representations, enabling more sophisticated speech modifications.

2.2.1 Linear Prediction

A linear prediction filter of integer order p defines the output signal $y(n)$ (here: synthesized speech) as the result of an excitation signal $x(n)$ filtered by a recursive (autoregressive) filter $a(k)$:

$$y(n) = x(n) + \sum_{k=1}^p a(k)y(n-k) \quad (1)$$

It can be shown (e.g., [14], see Fig. 4.1) that the filter coefficients $a(k)$ represent a vocal tract model comprised of p sections of an acoustic tube of stepwise constant cross-sectional areas. Wave propagation in the tube is assumed to be without losses and one-dimensional (plane), and tube walls are assumed to be rigid. Any side branch (e.g., nasal tract) is not part of the model.

In the simplest version of LPC-synthesis, the excitation signal $x(n)$ is either a train of pulses or white Gaussian noise (see Fig. 1). The pulse train approximates the glottal excitation for voiced sounds, while the white Gaussian noise represents the turbulent noise that excites the vocal tract for unvoiced sounds. Note that both the noise and the (single) pulse excitation are assumed to have a flat (over frequency) spectrum. Note also that the LPC filter gives the synthetic speech the desired spectral envelope (approximating the spectral envelope of human speech and matching its formants), while the simplistic pulse/noise excitations clearly fail to match spectral details.

LPC synthesis as described so far violates property 1 stated above. LPC synthesis clearly sounds “buzzy” (due to the repetitive nature of the pulses that do not match the spectral details of human glottal excitation) and, hence, is not transparent as a speech

presentation. However, because of its relationship to vocal tract geometry, LPC lends itself to spectral modifications that are based on geometric differences of vocal tracts (e.g., male to female speech conversion, see section 3.3 below). Improvements to speech quality can be achieved by using more sophisticated models of the excitation, such as glottal pulse models (see [17]section 7.4.3; also see multi-pulse models [3]). In any case, it should be obvious that pitch can be modified by changing the pulse rate of the excitation and durations can be modified by changing the updated rate of the LPC filter coefficients during synthesis. Also, smoothing of the filter can be used to achieve better concatenation between units.

2.2.2 Time Domain Pitch Synchronous Overlap Add (TD-PSOLA)

An intriguingly low-complexity way of modifying pitch and duration of recorded speech was introduced in [15]. The basic idea is outlined in Fig. 2.

The left column of Fig. 2 shows time-domain speech waveforms and the right column shows spectra related to the waveforms on the left. For voiced speech as depicted in the top row, we recognize the quasi-periodic waveform that gives rise to the comb-filter like fine structure that is visible in the spectrum depicted on the top right.

In the second row of graphs in Fig. 2, four separate waveforms have been extracted. Each of them encompasses $L=$ two pitch periods and has been tapered with a Hanning (raised squared cosine) window so the waveforms build up and die down gradually. Note also that the individual waveforms have been repositioned in time relative to each other. The right panel in the second row shows the spectrum of one of the individual waveforms. Very little of the harmonic structure remains visible.

Finally, in the third row in Fig. 2, we see the result of adding the individual waveforms in their new positions in time. In this example, the outcome is a speech signal of lower pitch and longer duration. The related spectrum on the right shows the harmonic structure of that lower pitch.

Higher pitches can be achieved by staggering the individual waveforms closer together in time before adding them up. Shorter durations are achieved by dropping individual waveforms, while longer durations are achieved by repeating individual waveforms as needed. Finally, note that the Hanning window has the convenient property that reproduces the original speech signal when no modifications of pitch or durations are performed.

While the TD-PSOLA method may be applied to the speech signal itself, a possible variant would be to apply it to the so-called LPC-residual. The LPC residual is the specific excitation signal $x(t)$ in Eq. 1 above that would be needed to match the original speech exactly with the synthetic output $y(t)$. The LPC residual can be obtained by inverse filtering of the speech signal. Inverse filtering has the effect of “whitening” the individual waveforms used in TD-PSOLA. Combining the LPC and TD-PSOLA methods this way also allows for spectral envelope smoothing between inventory units.

For time-scale modifications without any pitch modification, the so-called WSOLA approach may be used [30]. (Sound Clip examples “prosody”)

2.2.3 Sinusoidal Representations

The main drawback of the simplest form of LPC synthesis is its crude representation of the glottal excitation signal. On a frame-by-frame basis, that is over 5-50ms intervals, the amplitude and phase relationships of the first few harmonics may contain crucial information on speaker identity (independent of the super-segmental prosodic and phonetic cues). Therefore, sinusoidal speech representations can be viewed as a more sophisticated means to meet the transparency requirement stated in the beginning of this section. As with TD-PSOLA, we have the choice of including LPC filtering to allow for vocal-tract geometry motivated modifications of the spectral envelope and for its smoothing at concatenation points. Different from TD-PSOLA, however, pitch modifications are done in the frequency domain, by compressing or stretching the spectrum. Slowing down and speeding up depends on the parameter update rate.

For the following, we chose to follow the pioneering work of [20]. Let $s(t)$ be a signal of interest, either speech or LPC residual, then

$$s(t) = \sum_{l=1}^{L(t)} A_l(t) \cos[\Omega_l(t)], \quad \text{with} \quad \Omega_l(t) = \int_{t_l}^t \omega_l(\sigma) d\sigma + \phi_l, \quad (2)$$

where $A_l(t)$ is the slowly time-varying amplitude of the l -th sine wave, $\Omega_l(t)$ its instantaneous phase, ω_l the radian frequency of the l -th sine wave, and ϕ_l its phase. The analysis task is comprised of estimating harmonic frequencies and phases. Potential problems are the “birth” and “death” of specific sinusoids and the representation of unvoiced speech.

Several variants of the sinusoidal approach exist. One is the HNM (Harmonic plus Noise Model) representation [26][12] that splits the speech spectrum into a low-frequency voiced part that is modeled by the sinusoidal approach, and a high-frequency unvoiced part that represents the more noise-like characteristic of speech spectra at high frequencies. The “maximum voiced frequency”, which is the cut-over frequency between the two representations, may be a fixed frequency or may be set once for each speaker individually or may be even updated over time or dependent on the specific speech sound. Another approach is STRAIGHT [13] where special care is given to estimating the harmonic amplitudes accurately (bias-free). Both, HNM and STRAIGHT, however, assume that all sinusoids are at frequencies that are integer multiples of the fundamental frequency of the signal. For practical applications in speech synthesis, the reader should be aware of the fact that specific speakers’ voices may be less suitable for sinusoidal speech representations than other speakers. For a “bad” speaker, the coded/decoded speech may sound somewhat busy or metallic, which is in clear violation of the transparency requirement. (Sound Clip examples “prosody”)

3. Modification of voice characteristics

Variability in speech has several dimensions, the most important ones for speech synthesis purposes being speaker variability, linguistic variability (inter and intra-speaker), and task variability. The acoustic correlates of these dimensions are variations in pitch, duration, amplitude, and spectrum. We already discussed how to change these low-level attributes with the exception of spectral characteristics.

In the following, we will focus on higher level applications of signal processing that are used to change voice characteristics on a more global scale. Among these applications are modifications to speaker identity (voice transformation, voice alteration targeted at changing gender, individual differences, or even dialect), and modifications to speaking style and/or emotions for a given speaker (changing emotions or adapting the speech to fit environmental needs such as Lombard speech [19] that is appropriate for noisy environments). Although the algorithms used may serve both classes of applications, there are important differences between the two. While changing the speaker identity can leverage the significant effort that went into creating one high quality TTS voice for the purpose of being able to offer several such voices, modifying intra-speaker characteristics such as speaking style or emotion avoids the efforts of having to re-record large parts of an existing voice database with the same speaker, each time using a different speaking style or emotion.

In order to mimic real-life variability of speech with a synthesizer, it is helpful to list different speech and speaker characteristics that might be changed by signal processing. Linguistic variability encompasses variations due to intonation plus the fact that phonemes may also have different spectral properties depending on context (coarticulation), speaking style/emotion, and speaker. Gender differences include the fact that average pitch and formant frequencies are higher for females than for males. Dialect differences consist of the use of different phonemes for a word and the use of different allophones for the same phoneme in a given context, plus differences in prosody. As an example, Fig. 3 shows data for male and female 2nd and 3rd formant frequencies depending on the dialect of US-English for the vowel /u/ [27]. Some of the variability of speech between individuals clearly has reasons that can be traced to differences in vocal tract geometry. Consequently, spectral (vocal-tract motivated) changes, as well as “individualized” prosody models that capture a specific speaker and/or emotion/speaking style, are being summarized next.

3.1 Spectral models for changing speaker and/or speaking style

Here we outline methods for transforming speech of one speaker either to sound like speech from another speaker, or to sound like speech of the same speaker, but in a different speaking style or with different emotions. We focus primarily on vocal-tract related characteristics of speech but note that the basic concepts also apply to representations of the glottal excitation.

Acoustic correlates of speaker individuality can be found in vocal-tract formants, spectral tilt, and glottal characteristics. One obvious way to try to map “source speech” (what we

have) with a certain set of characteristics (e.g., speaker identity, etc.) to “target speech” (what it should sound like) with another set of characteristics is to use two codebooks of vector-quantized spectral (e.g., LPC) vectors, each trained on one of the two sets, and find a mapping between corresponding codebook entries. This technique was pioneered by Abe at NTT in Japan [1] using parallel corpora, source set A and target set B , comprised of identical sentences. A challenge with this approach is the time-alignment of vectors from one set with the corresponding vectors from the other. Although ideally the same material is recorded in both corpora, it is not always straight-forward to map speech frames even for a given single word, in particular when two different speakers spoke the word, because of differences in vowel reductions, phone omissions or additions, or other individual pronunciation differences. Dynamic Time Warping (DTW) is used to establish this correspondence and histograms of these correspondences serve as weights for the mapping function:

$$\mathbf{S}_i^{(A \rightarrow B)} = \frac{\sum_{j=1}^J w_{ij} \mathbf{S}_j^{(B)}}{\sum_{j=1}^J w_{ij}}. \quad (3)$$

Here A and B represent source and target sets, respectively, $\mathbf{S}_i^{(A \rightarrow B)}$ is the i -th spectral vector of the mapped source codebook, and w_{ij} are the histogram counts of how often vector i of codebook A was found to correspond to vector j in codebook B , itself containing J vectors $\mathbf{S}_j^{(B)}$. Note that Eq. (3) can be viewed as an interpolation between different target spectral vectors. Therefore, it is important to choose a representation \mathbf{S} where linear interpolation makes sense and does not lead to “over-smoothing”.

Another way to map features from a source corpus to those of a target corpus is to estimate an appropriate transformation function. The most popular method for doing this is based on Gaussian Mixture Models (GMMs). One of the early adopters of GMM-based methods for voice conversion is Stylianou [26].

Let ω_m denote the m -th acoustic class out of a total of M classes. In a Gaussian mixture density model, the probability density functions of any of the K observation vectors \mathbf{x}_k ($k = 1, \dots, K$) is given by:

$$p(\mathbf{x}_k) = \sum_{m=1}^M P(\omega_m) p(\mathbf{x}_k | \omega_m) = \sum_{m=1}^M c_m N(\mathbf{x}_k, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \quad , \quad (4)$$

where \mathbf{x}_k is a L -dimensional random vector. The conditional probability density for each of the M classes ω_m is assumed to be a Gaussian component density $N(\mathbf{x}_k, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, and the c_m 's (the probabilities for classes ω_m) are the related normalized mixture weights ($\sum_{m=1}^M c_m = 1$). Each L -variate Gaussian component density is of the form:

$$N(\mathbf{x}_k, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(2\pi)^{L/2} |\boldsymbol{\Sigma}_m|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_m)' \boldsymbol{\Sigma}_m^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_m)\right] . \quad (5)$$

Here $\boldsymbol{\mu}_m$ is an L -dimensional mean vector for the m -th component, and $\boldsymbol{\Sigma}_m$ is its covariance matrix. Therefore, the m -th mixture component is characterized by its center (the mean) and the spread around that center, expressed by the covariance matrix. Note that Eq. (4) expresses arbitrary distributions as a linear combination of Gaussians. If we associate the m -th component with a specific acoustic class (e.g., nasals, vowels, etc.), it should be intuitively clear that mapping techniques based on Eq. (4) are more robust than those based on Eq. (3) above that do not explicitly distinguish between different acoustic classes while also lacking an inherent model (here Gaussian) for generalizing to unseen data.

In estimating a mapping $F_x(\mathbf{x})$, the goal is to minimize the mean-squared error between the transformed set $\tilde{\mathcal{Y}} = \{\tilde{\mathbf{y}}_k = F_x(\mathbf{x}_k), k = 1 \dots K\}$ of source data and the training set $\mathcal{Y} = \{\mathbf{y}_k, k = 1 \dots K\}$ of target data. Again, we assume that both sets of observation vectors are time-aligned using DTW.

Furthermore, let $h_m(\mathbf{x}) = P(\omega_m | \mathbf{x})$ be the conditional probability of \mathbf{x} belonging to class ω_m , and, using Eq. 4 and Bayes' rule, we find:

$$h_m(\mathbf{x}) = P(\omega_m | \mathbf{x}) = \frac{P(\omega_m) p(\mathbf{x} | \omega_m)}{p(\mathbf{x})} = \frac{c_m N(\mathbf{x}, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{i=1}^M c_i N(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} . \quad (6)$$

The conversion function $F_x(\mathbf{x})$ is chosen to be of the form

$$F_x(\mathbf{x}) = \sum_{m=1}^M h_m(\mathbf{x}) [\boldsymbol{v}_m + \boldsymbol{\Gamma}_m \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m)] . \quad (7)$$

The unknown L -dimensional vectors \boldsymbol{v}_m and the unknown $L \times L$ matrices $\boldsymbol{\Gamma}_m$ are obtained by solving normal equations for the least-squares problem between \mathcal{Y} and $\tilde{\mathcal{Y}}$. Details can be found in [26]. Note also that the method can be improved further by basing the conversion function on joint source and target densities [11] instead of basing them solely on source densities as outlined above.

Most recently, Toda [28] introduced a new statistical method based on maximizing the likelihood (ML) function $\log p(\mathbf{y}, \mathbf{x})$ for each class (mixture component). This leads to a generalization of the mean vectors $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$ in the simplified derivation given above. Toda also included dynamic aspects (time derivatives) of features to exploit their trajectories over time. Both measures led to less ‘‘erratic’’ feature

trajectories of the transformed speech when compared to the basic GMM method while keeping the global variance (observed over time) as high as in the recorded target speech (the GMM method usually reduces that variance considerably). The new method also avoids over-smoothing of GMM-converted spectra that sometimes leads to overly broad formants in the converted speech. (Sound clips “conv11” – “conv44”)

3.2 Prosody models and prosody changes

Having the ability to change the frame/segment-related signal correlates to prosody, such as pitch, durations, and amplitude, gives us the tool to approach changing prosody aspects of speech on a larger time scale. One purpose of super-segmental prosodic changes is to convey (different) emotions. Another is to indicate topic structure, or speech acts and other functional aspects of intonation [7]. However, there is evidence that a perfect “synthetic” copy (i.e., a TTS-based mimic) of a human speaker is only possible with a data-driven prosody model that is trained on the specific speaker. Here the idea is to enable powerful generalization from sparse training data by capturing the way a specific speaker uses prosody in a specific context (e.g., in a dialog scenario) and for a specific speech act (e.g., greeting) and make the TTS system predict the right prosody so its unit selection module can do a reasonable job of retrieving the correct speech segments, and/or help a prosody-modification enabled back-end to do the necessary signal modifications.

Recent developments in prosody models, much as in speech representations (i.e., “signal models”) for speech coding or speech modification purposes, followed a similar trend: from “knowledge-based” to “data-driven” paradigms [5]. Knowledge-based systems incorporate the skills and intuition of the researcher. Data-driven models learn from training data. As in many other cases, the “right” middle ground is to allow for data-driven adaptability/flexibility while basing the models on generally accepted principals for robustness and generalization [22]. Clearly, the future belongs to systems that use machine learning techniques that are seeded with generally accepted intonation rules.

Intonation, the way we assign fundamental frequencies to voiced signal segments of speech, is an important part of prosody generation. Theories of “tone sequences” pose hierarchical structure on the problem that may be expressed on a symbolic level in the form of Tone and Break Indices (“ToBI”, [25]). Such a symbolic linguistic representation then needs to be followed by a way of actually assigning fundamental frequency values. An example of an early model for generating actual fundamental frequency contours for synthesis purposes is the Fujisaki model [6]. The Fujisaki model basically applies LPC (see above) to fundamental frequency contours in that it tries to correlate major intonation events in timing, durations, and amplitudes with linguistically-motivated notions of phrase and accents “commands”. It models these commands as step functions and then filters these “excitation” signals by an LPC-like smoothing filter to arrive at a fundamental frequency contour for a whole utterance.

A second way to build an intonation model is to identify proper “building blocks” of intonation in the form of stylized fundamental frequency (F_0) tracks and to associate each of these tracks with the terminal leaves of a Classification and Decision Tree, such as

CARTs or via neural networks (for details, see, e.g., section 6.9.2 in [5]). Such representations are driven by syntactic features provided by automatic text analysis or (manual) mark-up.

Most recently, ways have been found of to exploit directly the large speech databases recorded with a single speaker for unit-selection synthesis. Here, the idea is to search the database for a reasonable prosody match at the symbolic level and harvest appropriate F_0 tracks [8]. Challenges of this method include the sparseness of the data even existing in a large speech database and how to fill in the “holes”. Different ways to “fall-back” to fundamental knowledge-based rules exist.

Any of the parametric intonation/prosody models just discussed can be employed within the signal transformation framework of section 3.1. For convincing results, the feature vectors \mathbf{x} and \mathbf{y} of that framework have to include model parameters of the chosen prosody model or representation.

3.3 Voice alteration

Voice alteration is the task of changing (“disguising”) a source speaker’s voice so he/she can no longer be recognized without actually trying to match any peculiar target speaker. Such capability is useful for speech synthesis purposes since any “new” voice obtained through speaker alteration helps in marketing of a TTS system.

In addition to changing pitch as discussed previously, changing the vocal tract size is helpful. Starting from a male voice database, for example, raising the pitch and shortening the vocal tract may simulate a female or child voice. Several methods are available to achieve such alteration. One specific method summarized here exploits the fact that Linear Prediction Coefficients (LPCs, see section 2.2.1 above) can be converted to log (pseudo) areas that define a corresponding acoustic tube. Scaling that tube and converting back to LPCs results in voice alteration.

It can be shown (e.g., [14], section 5.5.2] that stable filter LPCs can be converted into reflection coefficients $r(k), k = p, p-1, \dots, 1$, by starting a backward recursion with initial LPCs $b_{p,0} = 1, b_{p,1} = a_1, \dots, b_{p,p} = a_p$, and setting $r_p = a_p$:

$$b_{k-1,i} = \frac{b_{k,i} - r_k b_{k,k-i}}{1 - r_k^2} \quad \text{for } i = 0, 1, \dots, p-1$$

$$\text{extract} \quad r_k = b_{k,k}$$

$$\text{and check} \quad |r_k| < 1$$
(8)

If any reflection coefficient has a magnitude that is larger than unity, the corresponding LPC filter is unstable.

From reflection coefficients obtained using Eq. (8), we then compute *Log Areas* using

$$\begin{aligned}
LA_0 &= \mathbf{0} \\
LA_{k+1} &= \log\left(\frac{1+r_k}{1-r_k}\right) \\
\text{for } k &= 1, \dots, p
\end{aligned} \tag{9}$$

Fig. 4 shows the log areas corresponding to a LPC filter of order $p=16$ for the vowel /i/. Here the 17 log areas are shown as represented as sections of the corresponding acoustic tube. Note that the ubiquitous $LA_0 = \mathbf{0}$ on the left corresponds to the “lips” (front) while LA_{16} on the right corresponds to the “glottis” (back). Also shown in Fig. 4 is a straight-line representation that can be used for interpolating the geometry of the pseudo vocal tract.

Fig. 5 shows a pseudo tract that is shortened by 25% by “compressing” the interpolated straight line representation and resampling it. Note that the high-order log areas have been “extrapolated” to be of the same value as the highest-order original log area. Different, non-linear stretching or compression schemes are admissible. For example, the “back cavity” (i.e., pharynx section) could be stretched by a different factor as the “front cavity” of the vocal tract. Interesting voices can be created easily this way, for example, “gnome”-like voices for computer games. (Sound Clip examples “prosody”)

4. Conclusions

In this article, we have highlighted algorithms and techniques to transform speech from one speaker using a specific speaking style and being in a specific emotional state to sound like either a different speaker or to fit a different speaking style or representing a different emotional state. All signal aspects related to all parts of the human vocal apparatus have to be included in order to arrive at high-quality results.

For the low-level signal processing, many algorithms can be borrowed from the most advanced speech coding/compression efforts. Besides the strict requirement of transparency (i.e., non-detectability), speech coding is used to reduce storage requirements, which is essential for distributing and incorporating large voice databases with/into Text-to-Speech systems. Nontraditional requirements of speech coding algorithms and speech representations are related to random access and modification of signal parameters. An important open problem that still needs to be solved is the effective representation and modification/transformation of glottal excitation.

In addition to the basic algorithms for signal modifications, there are higher-level algorithms that capture (“learn”) a speaker’s speech on several time scales: frames, segments, words, and whole sentences. The reader should expect future research work being applied in even larger contexts like paragraphs and whole “stories”. For this, advanced machine learning techniques such as Support Vector Machines [29] and Boosting [30] may outperform traditional techniques based on, for example, CARTs.

The most advanced voice modification techniques are still not good enough to deliver convincing “transformed” speech. With increasing quality, however, the need for extensive and comprehensive recording of voice databases will be reduced to just the “essentials”, while more and more “derivative” speaking styles, emotions, and/or speakers will be created through modification of a smaller set of original recordings.

References

- [1] Abe, M., Nakamura, S., Shikano, K. & Kuwahara, H. (1990). ‘Voice conversion through vector quantization’ In *Proc. IEEE ICASSP’88*, S14.1, 665-658.
- [2] Black, A. W. & Taylor, P. A. (1994). ‘CHATR: A Generic Speech Synthesis System’ In *COLING ‘94*, 983-986.
- [3] Caspers, B. & Atal, B.S. (1987). ‘Role of Multipulse Excitation in Synthesis of Natural Sounding Voiced Speech’ In *Proc. IEEE ICASSP*, Dallas, 2388-2391.
- [4] Conkie, A. D. (1999). ‘Robust Unit Selection System for Speech Synthesis’ In *Joint Meeting of ASA, EAA, and DAGA*, paper 1PSCB_10, Berlin, Germany, full paper available on-line at <http://www.research.att.com/projects/tts/pubs.html>.
- [5] Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis*. Boston: Kluwer Academic Publishers.
- [6] Hirose, K., Fujisaki, H. & Kawai, H. (1986). ‘Generation of Prosodic Symbols for Rule-Synthesis of Connected Speech of Japanese’ In *Proc. IEEE ICASSP’86*, Tokyo, 2415-2418.
- [7] Hirschberg, J. (2004) ‘Speech Synthesis Prosody’ This book.
- [8] Huang, X., Acero, A., Hon, H. Ju, Y., Liu, J. Meredith, S. & Plumpe, M.(1997) ‘Recent Improvements on Microsofts Trainable Text-to-Speech System – Whistler’ In: *Proc. IEEE ICASSP’97*, Munich, Germany, 959-962.
- [9] Hunt, A. & Black, A. (1996). ‘Unit selection in a concatenative speech synthesis system using a large speech database’ *Proc. ICASSP*, vol. 1, 373-376.
- [10] Jilka, M., Syrdal, A. K., Conkie, A. D. & Kapilow, D. A. (2003). ‘Effects on TTS quality of methods of realizing natural prosodic variations’ In *Proc. ICPHS*, Barcelona, Spain.
- [11] Kain, A. & Macon, M. (1998). ‘Spectral Voice Conversion for Text-to-Speech Synthesis’ In *Proc. IEEE ICASPP’9*, 285-288.
- [12] Kain, A. & Stylianou, Y. (2000). ‘Stochastic Modeling of Spectral Adjustment for High Quality Pitch Modification’ In *Proceedings ICASSP 2000*, Istanbul, Turkey.

- [13] Kawahara, H., Masuda-Katsuse, I. & de Cheveigne, A. (1999). ‘Restructuring Speech Representations Using a Pitch-Adaptive Time-Frequency Smoothing and an Instantaneous-Frequency-Based F0 Extraction: Possible Role of a Repetitive Structure in Sounds’ *Speech Communication*, 27, vol. 3-4, 187-207.
- [14] Markel, J.D. & Gray, Jr., A.H. (1976) *Linear Prediction of Speech*, Berlin: Springer-Verlag.
- [15] Moulines, E. & Charpentier, F. (1990). ‘Pitch Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones’ *Speech Communication*, Vol. 9, No. 5-6, 453-467.
- [16] Nussbaum, H. C. (2004). ‘Speech Synthesis’, this book.
- [17] Olive, J., van Santen, J., Möbius, B. & Shih, C. (1998). ‘Synthesis’ In *Multilingual Text-to-Speech Synthesis* (R. Sproat, ed.), Boston: Kluwer Academic Publishers.
- [18] Ostendorf, M., Bulyko, I. (2002). ‘The Impact of Speech Recognition on Speech Synthesis’ *Proceedings IEEE 2002 Workshop on Speech Synthesis*, Santa Monica, Keynote Paper.
- [19] Pisoni, D. B., Bernacki, R. H., Nusbaum H. C. & Yuchtman, M. (1985). ‘Some Acoustic-Phonetic Correlates of Speech Produced in Noise’ In *Proc. IEEE ICASSP’85*, 1581-1584.
- [20] Quartieri, T. F. & McAulay, R. J. (1992). ‘Shape Invariant Time-Scale and Pitch Modification of Speech’ *IEEE Trans. Signal Processing*, 40, 3, pp. 497-510.
- [21] Sagisaka, Y., Kaiki, N., Iwahashi, N. & Mimura, K. (1992), “ATR – v-TALK speech synthesis system,” in: Proc. Int. Conf. on Speech and Language Processing 92, Banff, Canada, vol. 1, pp. 483–486, 1992.
- [22] van Santen, J. P. H. (1997). ‘Combinatorial Issues in Text-to-Speech Synthesis’ In *Proceedings Eurospeech’97*, Rhodes, Greece, Keynote Paper.
- [23] Schapire, R. E. (2002) ‘The Boosting Approach to Machine Learning: An Overview’ In *MSRI Workshop on Nonlinear Estimation and Classification*
- [24] Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., Säuberlich, B. (2003). ‘Restricted Unlimited Domain Synthesis’ In: *Proc. Eurospeech 2003*, Geneva, 1321-1324.
- [25] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J. & Hirschberg, J. (1992). ‘TOBI: A standard for labeling English prosody’ In: Proc. ICSLP’92, Banff, Alberta, Canada, 867-870.

- [26] Stylianou, I. (1996) *Modèles Harmoniques plus Bruit combines avec des Méthodes Statistiques, pour la Modication de la Parole et du Locuteur*. Doctoral thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France (in English).
- [27] Syrdal, A. K. (1996). ‘Acoustic Variability in Spontaneous Conversational Speech of American English Talkers’ In *Proceedings of ICSLP 96*, 438-441.
- [28] Toda, T., Black. A. W. & Tokuda, K. (2005). ‘Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter’ In *Proc. IEEE ICASSP’2005*.
- [29] Vapnik, V. N. (1998) *Statistical Learning Theory*, Hoboken, New Jersey: John Wiley & Sons.
- [30] Verhelst, W. & Roelands, M. (1993). ‘An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech’ In *Proc. ICASSP 93-II*, 554-557.

FIGURES

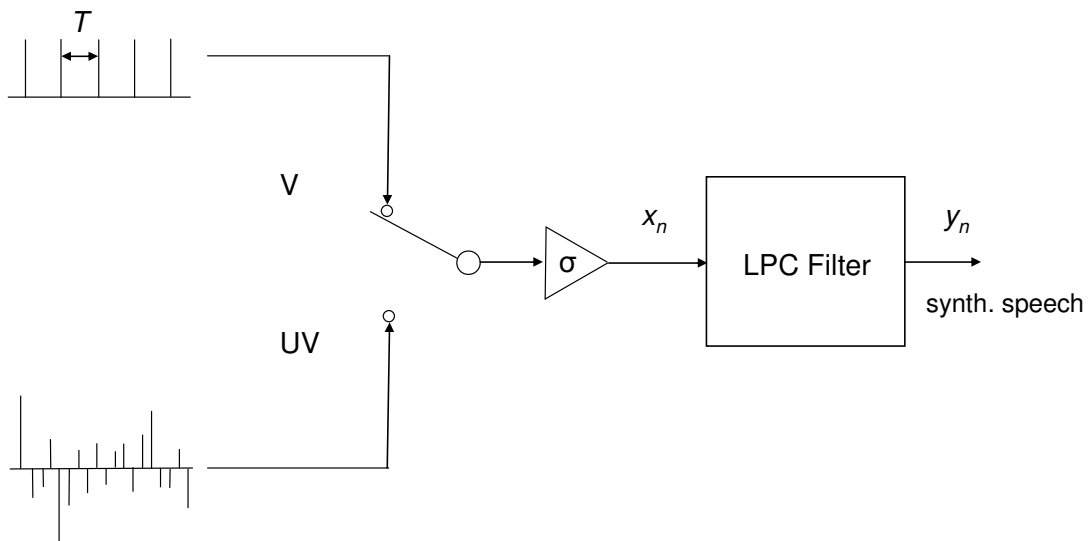


Figure 1: Block diagram of LPC synthesis. Periodic pulses with pitch period T are used for voiced excitation. Gaussian noise of unit amplitude is used for unvoiced excitation. The excitation is scaled in amplitude by gain factor σ .

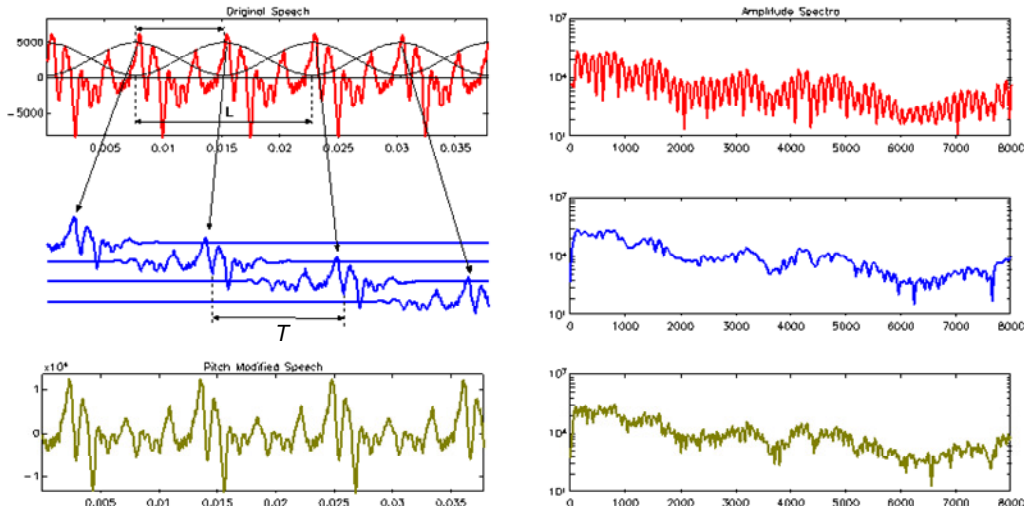


Fig. 2: Time-Domain Pitch-Synchronous Overlap-Add (TD-PSOLA). Adapted from [5], Fig. 10.1, p. 252.

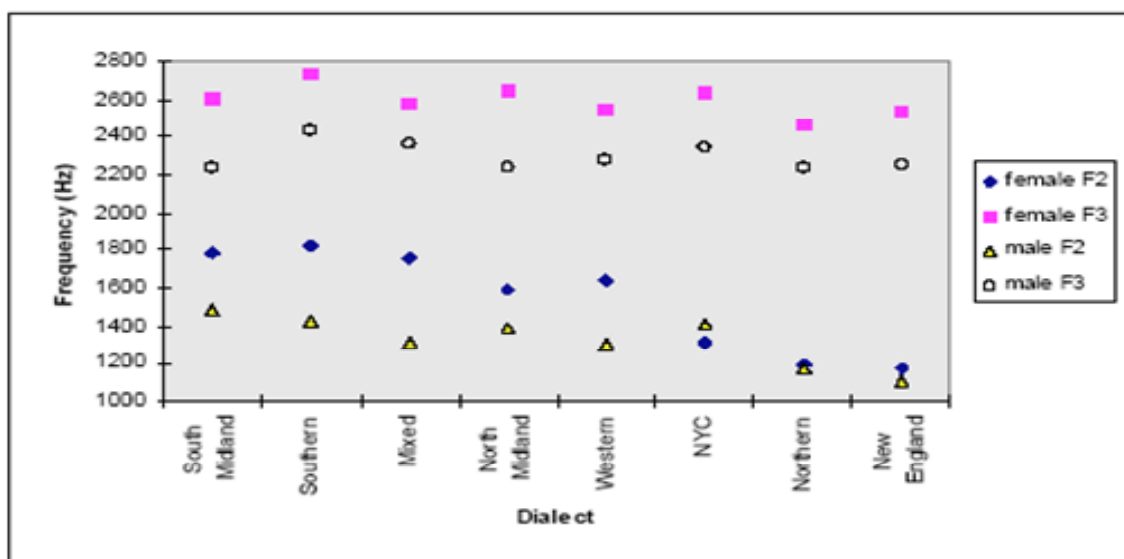


Fig. 3: Second and third formant frequencies for the vowel /u/ by dialect and gender. (Ann Syrdal, priv. comm.)

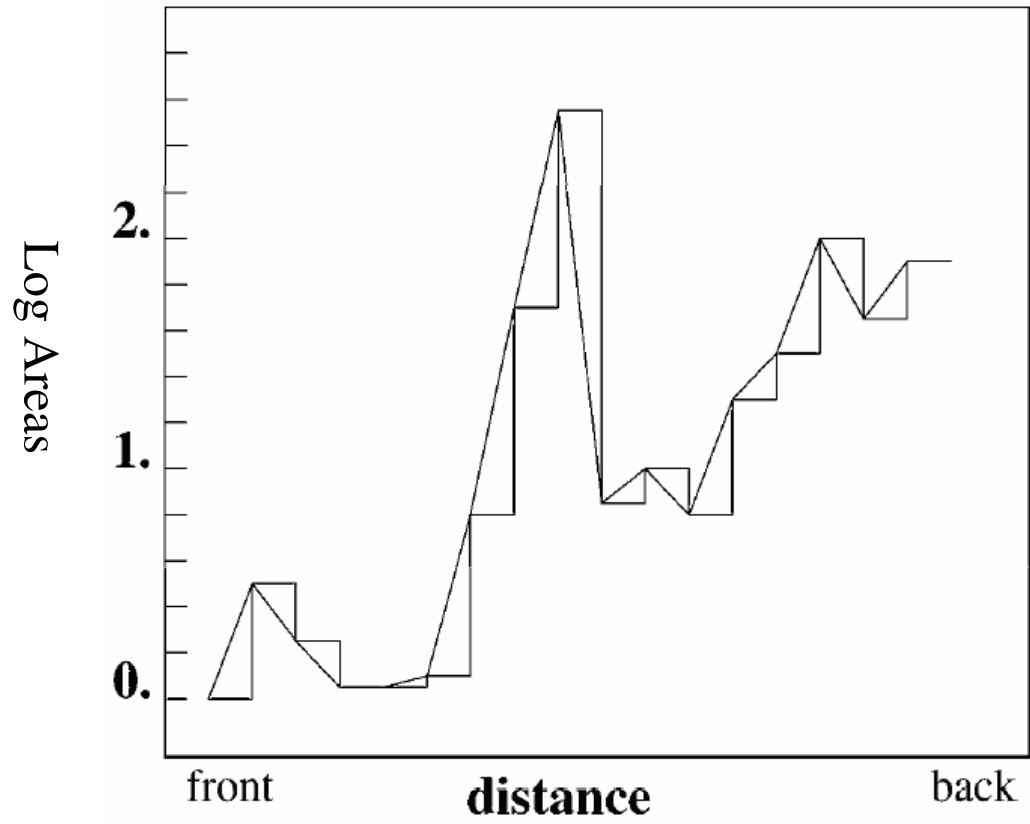


Fig. 4: Log Areas derived from LPC analysis for the vowel /i/.

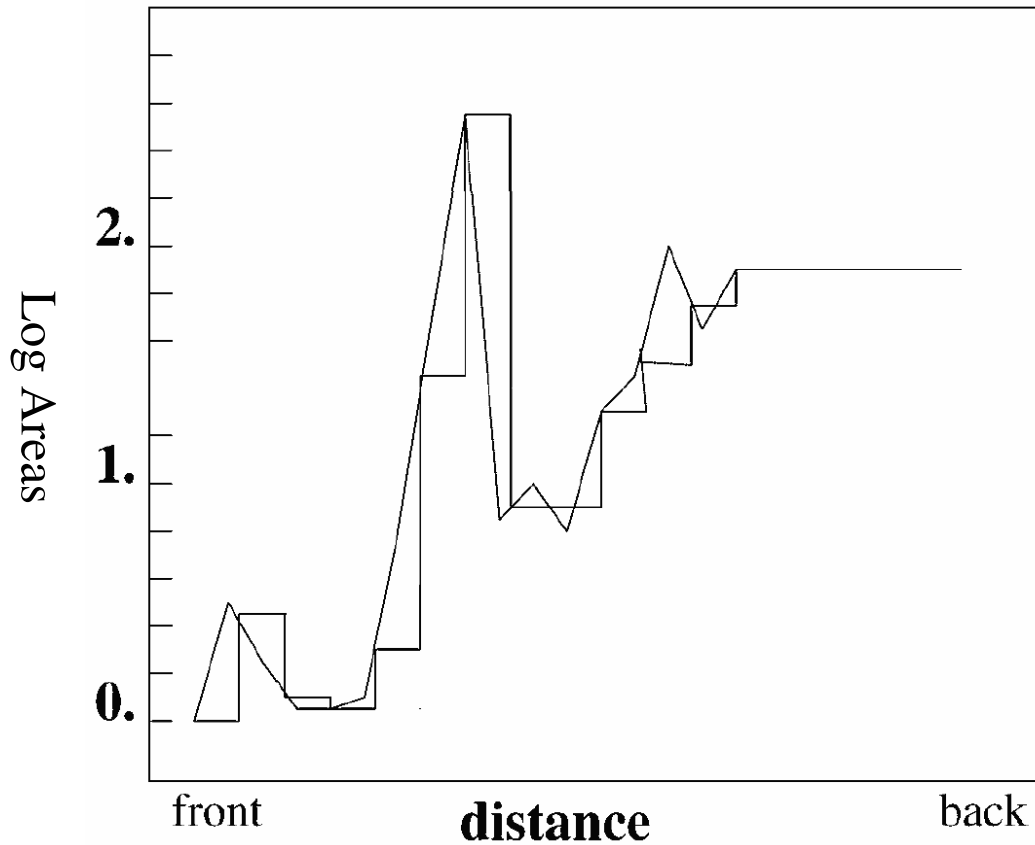


Fig. 5: Same log areas as in Fig.4. However, here the linearly interpolated line segments were “compressed” along the distance axis, representing a 25% reduction in vocal tract length. The final “stair case” representation was sampled from the linear segments and then used to re-synthesize the vowel.