Synthesis of Singing

Paul Meissner
Robert Peharz

June 26, 2008

Abstract

This paper describes three methods for the synthesis of singing speech. For each system, the speech synthesis process is explained in detail. Each of the described methods uses a different technique to produce synthetic speech. Finally the performance of these systems is discussed and possible further improvements are mentioned.

1 Introduction

The intention of this paper is to give an overview of methods for synthesizing singing speech. As this is a very current research topic, there are some very different basic approaches. Three of them are described in detail.

Synthesizing singing speech differs from the synthesis of spoken speech in several points. First of all the musical score has to be integrated. It contains instructions for pitch heights and note durations as well as overall properties of the song like tempo and rhythm. Secondly, this score should not be followed too strictly because that would lead to unnatural sounding speech. Thus several singing effects like vibrato, overshoot and preparation have to be considered and modeled by the systems. Another important point is the avoidance of "perfect" synthesis. Personal variations in the voice of singers have to be taken into account to produce decent results.

This paper is organized as follows: In section 2 an HMM-based approach is presented as an extension of an existing speech synthesis system [1]. Section 3 deals with an articulatory speech synthesizer [3] whereas in the sections 4, 5 and 6 a vocal conversion method based on the speech analysis system STRAIGHT is presented [6], [7]. Section 7 finally draws conclusions and compares the performance of the presented systems.

2 HMM-based synthesis of singing

This system uses a Hidden-Markov-Model based approach to produce synthetic speech presented in [2]. The usage of the HMM-based synthesis technique is justified by the necessary amount of recorded singing voice data, especially in comparison to the unit-selection method. The latter requires a huge amount of data to be able to take the large number of combinations of contextual factors that affect singing voice into account. An HMM-based system on the other hand can be trained with relatively little training data.

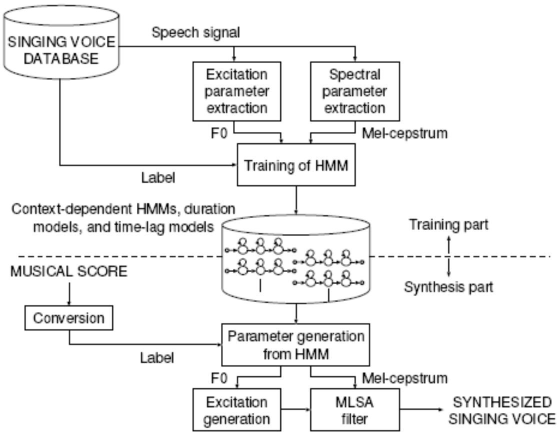

SINGING VOICE DATABASE
Speech signal
Excitation parameter extraction
Spectral parameter extraction
F0
Mel-cepstrum
Label
Training of HMM
Context-dependent HMMs, duration models, and time-lag models
Training part
MUSICAL SCORE
Synthesis part
Conversion
Label
Parameter generation from HMM
F0
Mel-cepstrum
Excitation generation
MLSA filter
SYNTHESIZED SINGING VOICE

Figure 1: Overview of the HMM-based system [1]

An overview of the speech synthesis system together with the analysis part can be seen in figure 1. In the upper part (analysis), there is a singing speech database from which labels and speech parameters are extracted. The latter are mel-cepstral coefficients (MFCCs) for the spectral features and F_0 for the excitation parameters. These parameters are used for the training of the context dependent phoneme HMMs. Also the state duration models and the so-called time-lag models, which will be described later, are trained.

In the synthesis stage, the given musical score together with the song lyrics are converted into a context-dependent label sequence. The overall song HMM is a concatenation of several context dependent HMMs which are selected by this label sequence. In the next step, the state durations together with the time-lags are determined. A speech parameter generation algorithm [2] is used to get the parameters for the Mel-log spectrum approximation (MLSA) filter, which finally produces the synthetic speech.

The system is very similar to an HMM-based reading speech synthesizing system presented in [2]. However, there are two main differences in the synthesis of singing: Contextual factors and the time-lag models, which will be described in the next subsections.

2.1 Contextual factors

According to [1], the contextual factors that affect singing voice should be different from those that affect reading voice. The presented method uses the following contextual factors:

- Phoneme
- Tone (as indicated by the musical notes)
- Note duration and
- Position in the current musical bar

For each of these factors, the preceding, succeeding and current one is taken into account. These factors are determined automatically from the musical score, however the paper does not go into detail about that.

2.2 Time-lag models

The time-lag models seem to be the main feature of this method. Their principal purpose can be explained in the following way: If a singing voice is synthesized that exactly follows the instructions given by the musical score, the result will sound unnatural. This is due to the fact that no human singer will ever strictly follow the score. There are always variations in any of the parameters and the time-lag models take variations in the note timing into account.

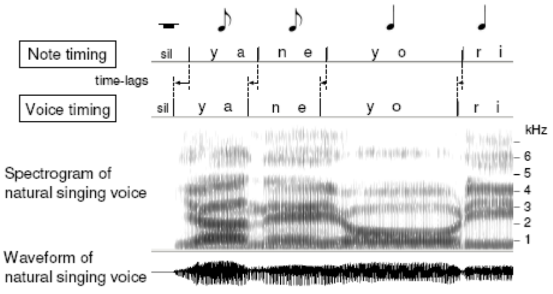


Figure 2: Usage of time-lag models [1]

The effect can be seen in figure 2. Time-lags are placed between the start of the notes given by the score and the start of the actual speech. The authors mention for example the well-known tendency of human singers to start consonants a little earlier than indicated by the score [1].

Determination of these time-lags is in principle analogous to the other speech parameters like pitch and state duration: There is a context-clustering using a decision tree. Context dependent labels are assigned to the time-lags and so they can be selected. Like the state duration models, the time-lag models are in fact just one-dimensional Gaussians. The process can be seen in figure 3.

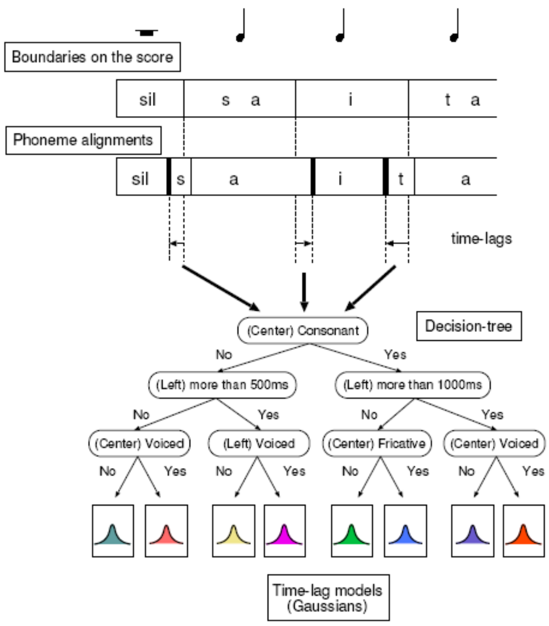


Figure 3: Decision tree clustering of the time-lag models [1]

At the synthesis stage, the concrete time-lags have to be determined. This is done by firstly taking each note duration from the musical score. Secondly, the state durations and time-lags are determined simultaneously such that their joint probability is maximized:

$$\begin{aligned} P(\mathbf{d},\mathbf{g}|\mathbf{T},\mathbf{\Lambda}) &= P(\mathbf{d}|\mathbf{g},\mathbf{T},\mathbf{\Lambda})P(\mathbf{g}|\mathbf{\Lambda}) \qquad (1) \\ &= \prod_{k=1}^{N} P(\mathbf{d_k}|T_k,g_k,g_{k-1},\mathbf{\Lambda})P(g_k|\mathbf{\Lambda}) \end{aligned}$$

where d_k are the start durations of the k^{th} note, g_k is the time-lag of the start timing of the $k+1^{th}$ note and T_k is the duration of the k^{th} note from the score. Finding the values of d and g that maximize this probability leads to a set of linear equations that can be solved quite efficiently.

2.3 Experimental evaluation

The authors say that they could not find a suitable and available singing voice database, so they recorded one by themselves for which they took a non-professional japanese singer. Manual corrections were done to enhance the quality. Speech analysis, HMM training and context clustering were performed. A subjective listening test was performed where they took 14 test persons and played them 15 randomly selected musical phrases synthesized with their system. An important result was that the incorporation of the time-lag models substantially improved the perceived speech quality. The test persons also found that the voice characteristics of the original singer were found in the synthetic speech. An example for this is that the original singer had the tendency to sing a little too flat, which was reflected by the synthesized F_0-pattern.

3 Articulatory synthesis of singing

This method, which is described in [3] and [4], uses a completely different approach to synthesize speech sounds. Like the HMM-based system from the previous section, it is also an extension of an already existing speech synthesizer that was modified to be able to produce singing speech. This was done for the synthesis of singing challenge at the *Interspeech 2007* in Antwerp, Belgium [5].

This method consists of a comprehensive three-dimensional model of the vocal tract together with additional steps and other models to simulate this model in order to get speech sounds out of it. The geometric model is converted into an acoustic branched tube model and finally to an electric transmission line circuit. Another interesting feature is the way how this method is controlled. All these points are explained in more detail in the following subsections.

3.1 Overview of the synthesizer

Figure 4 shows an overview of the articulatory speech synthesizer [3]. On top, the input to the system is missing but that will be described later. The system consists of three parts: The three-dimensional wireframe vocal tract representation (upper part of the figure), the acoustic branched tube model (middle part) and the simulated electrical transmission line circuit (lower part).

Shape and position of all movable structures in the vocal tract model are a function of 23 parameters, like horizontal tongue position or lip opening for example. To create the wireframe model, magnetic resonance images (MRI) of a german male speaker were taken during the pronunciation of each german vowel and consonant. This MRI data was used to find the parameter combinations.

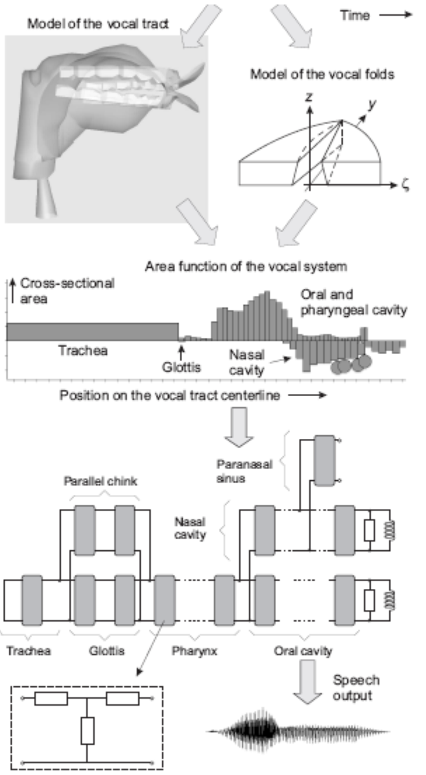


Figure 4: Overview of the articulatory synthesizer [3]

It is well-known that vowels and consonants do not stand for themselves concerning their articulation. There is the important topic of *coarticulation*. If you for example say the german utterances "igi" and "ugu" you will find out that your vocal tract behaves differently for both times you pronounce the consonant *g*. Your tongue will be raised both times, so the vertical tongue position is likely to be important for the pronunciation of a *g*. The horizontal tongue position however is different: For the "igi", the tongue will be more in front of the mouth than it is for the "ugu". So the horizontal tongue position for a *g* is an example for coarticulation, some parameters of the vocal tract depend on surrounding vowels or consonants. This method takes this into account by a so-called dominance model [4], which consists of a weighting of the vocal tract parameters for consonants and vowels. A high weight means that the corresponding parameter is important for this letter, a low weight indicates coarticulation.

The next step is the acoustical simulation of the model via a branched tube model that represents the vocal tract geometry. It consists of short adjacent elliptical tube sections which can be represented by an overall area function (see figure 4, middle part) and a discrete perimeter function.

This tube model can be transformed into an inhomogeneous transmission line circuit with lumped elements (see figure 4, lower part). This is done by using an analogy between acoutic and electric transmission that both deal with wave propagation along a path where there are impedance changes. Each of the tube sections is represented with a two-port T-type network, whose elements are a function of the tube geometry. Speech output is produced by simulating this network by means of finite difference equations in time domain. Many additional effects that can occur in the vocal tract are taken into account by making the electrical network more complex. There are for example parallel circuits for the paranasal sinus or parallel chinks in the vocal tract. The author says that all major speech sound for German are possible with this method [3].

3.2 Gestural score

In figure 4, the overall input to the system was missing. Utterances can be produced by certain combinations and movements of 23 vocal tract parameters but until now there is no way of controlling these parameters. The author developed a method called *gestural score* [3], [4] which fills the gap between musical score and lyrics on the one hand and the vocal tract parameters at the other hand. It is important to mention that this gestural score does not contain the vocal tract target parameters themselves, but are used for their generation. The author calls them "goal-oriented ariculatory movements" [4], so they more or less show what has to be done by the vocal tract, but not how.

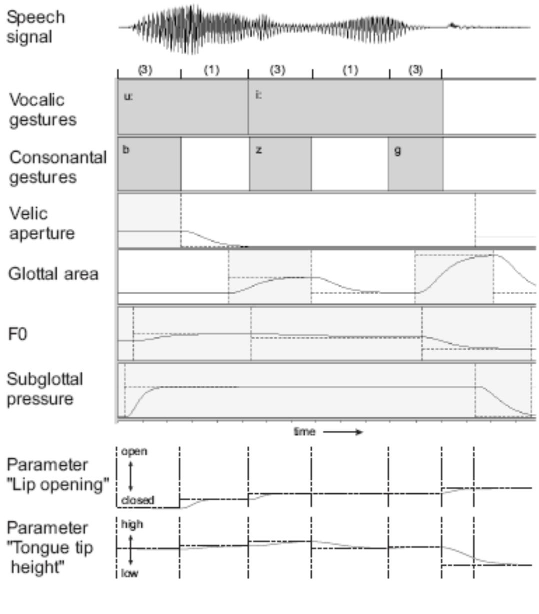


Figure 5: Gestural score with an example [4]

The way this gestural score works is explained by an example given in figure 5, the german utterance "musik" [4]. Below the speech signal there are six rows, which correspond to the six types of gestural scores. The first two are simply vocalic, in this case the *u* and *i* and consonantal gestures, here *m*, *s* and *k*. At the first glance it is striking that there seem to be the wrong consonants, but it is well-known that certain groups of consonants use very similar vocal tract shapes. The group (b,p,m) is an example for this. These consonants are produced by a common vocal tract configuration with minor variations. The second conspicuity is the overlapping of consonants and vowels. This is again due to the coarticulation phenomenon mentioned in section 3.1.

The other four gestural score types are the targets for velic aperture, glottal area, target F_0 and lung pressure. Below those there are two examples of concrete vocal tract parameters, the lip opening and the tongue tip height. These are generated from the gestural score and are target functions for the vocal tract parameters. They are realized using critically damped, third-order dynamical systems with the transfer function:

$$H(s) = \frac{1}{(1+\tau s)^3} \tag{2}$$

where τ is a time constant which can be used to control the speed of the parameter change.

The author derives the gestural score by using a rule-based transformation of a self-defined XML-format that represents a song including its score and lyrics.

3.3 Pitch dependent vocal tract target shapes

It is well-known that singers use different vocal tract shapes for the same vowel at different pitches. The original articulatory speech synthesizer did not take this into account and used just one general target shape. Figure 6 explains the occurring effects.

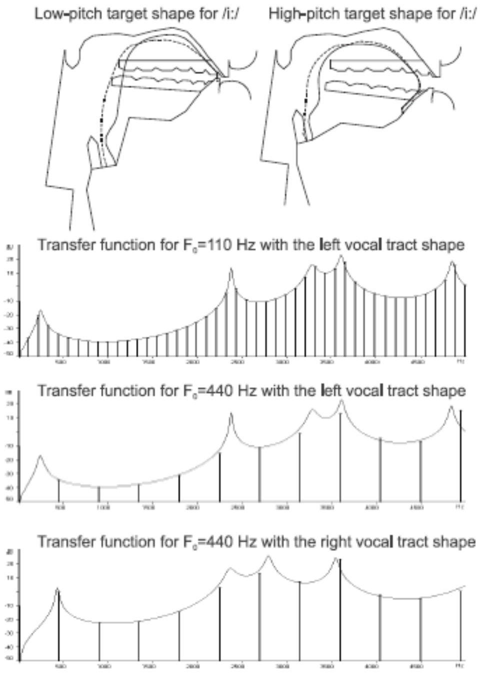


Figure 6: Pitch dependent vocal tract target shapes [3]

The solid line in the graphs represents the vocal tract transfer function and the spectral lines are the harmonics of the voice source. The first of the three graphs shows these for an /i:/, sung at $F_0 = 110$ Hz and the conventional, low pitch vocal tract shape (on the upper left). If that /i:/ is produced by the same vocal tract shape, but at $F_0 = 440$ Hz, this will result in the second graph that is shown. One can clearly see that the first formant of the vocal tract does not match the first formant of the voice source at all. To overcome this problem, a second, high-pitch (440 Hz) target shape, shown on the upper right, was created. So this and the conventional (110 Hz) shape are the two ”extreme” target shapes. The lowest graph in figure 6 shows the high-pitch /i:/ with the high-pitch shape. Here one can see that the first harmonic of the source and the first vocal tract formant match well.

Between these two vocal tract shapes, a linear interpolation is performed.

3.4 Evaluation

Articulatory speech synthesis is a very interesting approach to speech synthesis in general, because it reflects the natural way speech is produced. At the synthesis of singing challenge 2007, this method finished at the second place [5] out of six contestants. It is also worth mentioning that this method seems to need a lot of manual fine tuning, especially for optimizing vocal tract shapes. The author also mentions the guidance of this fine tuning by a professional singer as one possible future improvement.

4 Converting Speech into Singing Voice

The method of the winner of the Synthesis of Singing Contest 2007 [5], [6] is presented. The main idea here is to analyse a speaking voice reading the lyrics of song and to convert it to a singing voice by adapting the speech parameters according to a musical score and some know-how about singing voices. The speaking voice is analysed by a system called STRAIGHT. After adapting the parameters to represent a singing voice, they are re-synthesised. The next section describes the basic ideas of STRAIGHT, the main tool used here. Then the conversion system is discussed in more detail.

5 STRAIGHT

STRAIGHT stands for ”Speech Transformation and Representation using Adaptive Interpolation of weighted Spectrum” and was proposed by Kawahara et al. [7]. The idea for STRAIGHT was introduced by the need for flexible and robust speech analysis and modification methods. In its first version, it consists of a robust and accurate F0 estimator and a spectral representation cleaned from distortions which normally occur in the standard spectrogram.

5.1 Principle

The STRAIGHT system is derived from the channel vocoder, which is illustrated in figure 7. The channel vocoder detects whether the input signal x(k) is voiced or unvoiced and encodes this information in a binary variable S. If the input signal is voiced, the F0 (N0) is extracted, normally by measuring the fundamental period. Additionally, the input is processed by a band pass filter bank with central frequencies covering the frequency range of x(k). After each band pass filter, the envelope of the channel signal is determined giving a gain factor for each channel.

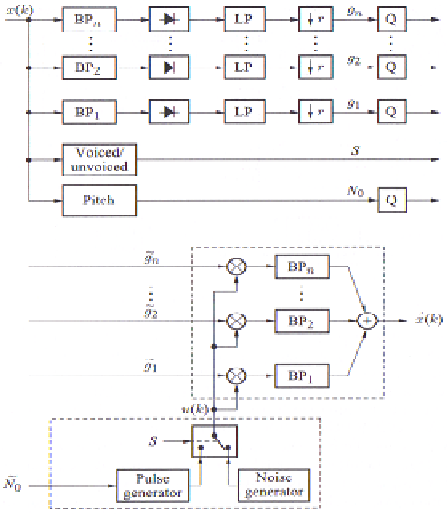


Figure 7: Channel vocoder

On the receiving side of the channel vocoder, an artificial excitation signal is generated from S and N0. This excitation is processed by an identical filter bank like on the transmitting side and amplified by the gain factors. The gain factors together with the filter bank model the vocal tract in the well known source-filter model (figure 8) widely used in speech processing. Note that a band-pass of the filter bank can be seen as a modulated version of a prototype low-pass, if the shapes of the band-passes are identical. Further the filter bank can be described in terms of of the Short Term Fourier Transform (STFT) using the impulse response of the prototype low-pass as windowing function [8]. In that way, the power spectrogram models the vocal tract filter.

The advantages of the channel vocoder are the simple and easy to understand concept, the intelligible speech quality and a robust possibility to change speech param-

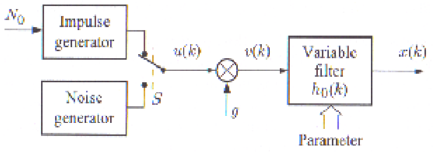


Figure 8: Source-filter model

eters.

The disadvantage is that the vocoder produces bad quality in sense of naturalness. A normal vocoder voice sounds mechanic and robot like. In some cases this desired. For example, it is a nice effect used in computer music to take the signal of an instrument as excitation of the vocal tract filter. The instrument still can be heard clearly, but it is coloured by the singing voice. As an example hear the song "Remember" by the group Air. However, the typical vocoder voice is not desired if the goal is natural sounding synthesised speech.

5.2 Spectrogram Smoothing

One of the main problems of the vocoder is a certain buzziness when the excitation is plosive. There are already affective approaches to reduce this problem. The other problem are interferences in the estimation of the spectrogram, introduced by periodic excitations, i.e. voiced sounds. In the Vocoder concept, the estimation of spectrogram is equivalent to the identification of the vocal tract filter. It is clear, that this identification is easier if a noise like input signal, i.e. unvoiced sounds, is used. However, if the excitation is quasi-periodic, the spectrogram exhibits interferences, which appear as periodic distortions in the time domain and in the frequency domain. Therefore information of F0 and the window length is visible in the whole spectrogram and a clean separation of excitation and vocal tract is not achieved.

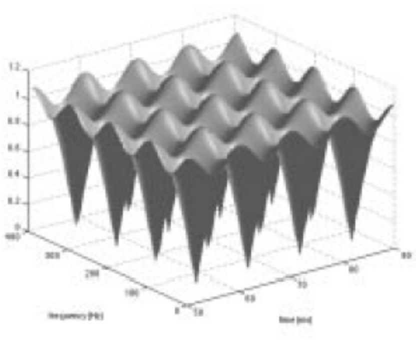


Figure 9: Spectrogram of a regular pulse train with interferences

The solution proposed by Kawahara et al. is to regard the periodic excitation signal as 2 dimensional sampling operator, which provides information every t0 and F0. Due to this, the spectrogram can be seen as 3D surface, where time and frequency are on the abscissae and the power is on the ordinate. In that way, spectral analysis can be seen as a surface recovery problem. The first approach proposed by the authors was to use a 2D smoothing kernel, which is computational intensive. The next approach they presented was to reduce the recovery problem to one dimension. If the window of the STFT matches the current fundamental period of the signal, the variations in the time domain are eliminated and the surface reconstruction problem is reduced to the frequency domain. For that, an exact and robust F0 estimator is needed, and will be discussed later as part of the STRAIGHT strategy.

The easiest method to recover the 1 dimensional frequency surface is to connect the frequency pins with straight line segments. An equivalent approach which is more robust against F0 estimation errors, is the convolution with a smoothing kernel. Luckily, convolution in frequency domain is equivalent to multiplication in time domain and can be achieved by selecting an appropriate form of the pitch adaptive time window. The authors chose a triangular window, since it corresponds to a $(\sin(x)/x)^2$ function in frequency domain and places zeros on all harmonic pins except the pin at 0. In addition, the triangular window is weighted with a Gaussian window to further suppress F0 estimation errors.

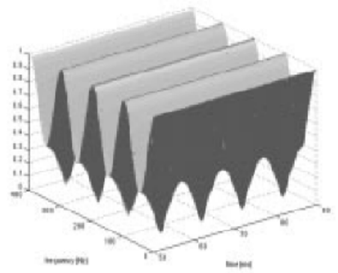


Figure 10: Spectrogram of pulse train using pitch adaptive windows

In figure 10 one can see that this operation eliminates the periodic interferences. One can also see phasic extinctions of adjacent harmonic components, visible as holes in spectral valleys. In order to reduce these, a complementary spectrogram is computed by modulating the original window in the form

$$w_c(t) = w(t)sin(\pi t/t_0) \tag{3}$$

The resulting spectrogram has peaks where the original spectrogram has holes, as it can be seen in figure 11. The spectrogram with reduced phase extinctions in figure 12 is created by blending the original and the complementary spectrogram in the form of

$$P_r(w,t) = \sqrt{P_0(w,t)^2 + \xi P_C(w,t)^2} \qquad (4)$$

The blending factor ξ was determined by a numerical search method and set to 0.13655.

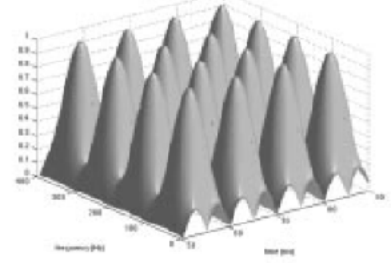


Figure 11: Complementary spectrogram

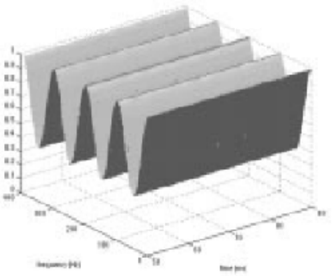


Figure 12: Blended spectrogram

One problem introduced by the method described here is over-smoothing. Using the pitch adaptive triangular window weighted with a Gaussian window is equivalent to apply a Gaussian smoothing kernel followed by a $(\sin(x)/x)^2$-kernel in the frequency domain. This over-smooths the underlying spectral information. To overcome this problem, Kawahara et al. modified the triangular kernel using an inverse filter technique. The new kernel reduced the over-smoothing effect while still aiming at the goal to recover spectral information in the frequency domain [7].

5.3 F0 Estimation

Normally the F0 is estimated by detecting the fundamental period. This approach is hard for speech signals, since they are not purely periodic and their F0 is unstable and time variant. The following representation of a speech waveform is used, which is a superposition of amplitude modulated and frequency modulated sinusoids

$$s(t) = \sum_{k \in N} \alpha_k(t) sin \left(\int_{t0}^{t} k(\omega(\tau) + \omega_k(\tau)) d\tau + \Phi_k \right) \qquad (5)$$

The STRAIGHT method uses a new concept called ”fundamentalness” for the F0 estimation. For this purpose, the input signal is split into frequency channels, where a special shaped filter is used. This procedure is illustrated in figure 13. Note that the filter has a steeper edge at higher frequencies and a slower cut-off at lower frequencies. This shape can contain the fundamental component alone, but will contain lower components if it is moved over higher components. The fundamentalness for each channel is defined as the reciprocal of the product of the FM and the AM components, where the AM component is normalized by the total energy and the FM component is normalized by the squared frequency of the channel. Therefore, the fundamentalness of a channel is high, if the FM and AM magnitudes are low. The F0 is determined by averaging the instantaneous frequencies of the channel with the highest fundamentalness index and its neighbouring channels. The fundamentalness was found to be a good estimator for F0 even at low SNR. Also, a reciprocal relation between the fundamentalness value and the estimation error of F0 was observed. Due to this, the fundamentalness can also be used for the voiced/unvoiced decision.

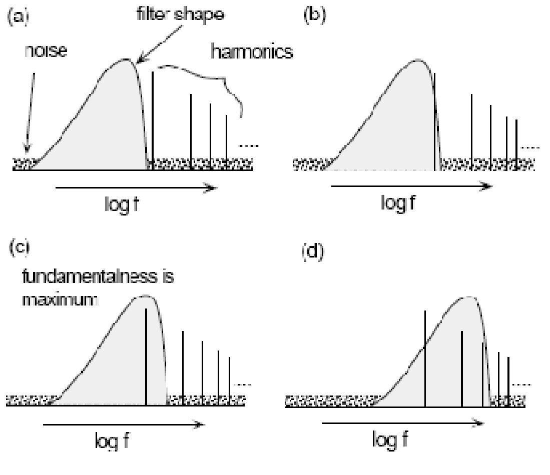


Figure 13: Illustration of fundamentalness

6 Application in the Speech to Singing Voice system

The overall system is sketched in figure 14 [6]. The speaking voice signal and the musical score including the song lyrics are inputs to the system. Additionally, synchronization information between these has to be provided, which is created by hand in the current system (see figure 15). STRAIGHT extracts the F0, the spectral envelope and a time-frequency map of aperiodicity, which is a concept introduced in later versions of STRAIGHT. These parameters are changed in three ways: change of the F0, change of the duration and change of the spectral information.

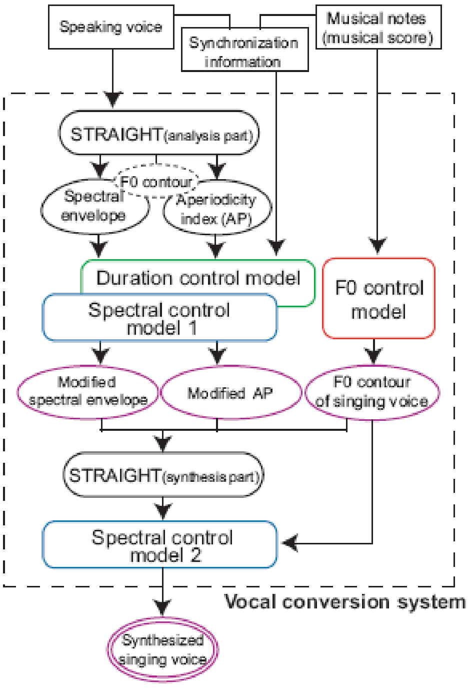


Figure 14: Overall conversion system

6.1 F0

The ideal F0 of the singing voice is completely given by the musical score (see figure 16). Following the pitch exactly would sound very unnatural. Therefore the F0 is changed according to features observed in real singing voices. Firstly, overshoot is added, which is a exceeding over the target note after a jump. Secondly, a vibrato is simulated by a 4-7 Hz frequency modulation. Thirdly, a movement of pitch in opposite direction just before a jump is added, which is called preparation. Fourthly, fine fluctuations (>10 Hz) in F0 are modeled by adding low-pass filtered noise.

6.2 Duration

The duration of the spoken words has to be adapted to the duration of sung words, given by the musical score. A consonant followed by a vowel is modelled as a consonant part, a boundary part of 40ms and a vowel part. The consonant parts are lengthened by fixed rates, dependent on the consonant type. These rates were found empirically. The boundary part is kept unchanged and the vowel part is lengthened, so that the whole combination fills the desired note length.

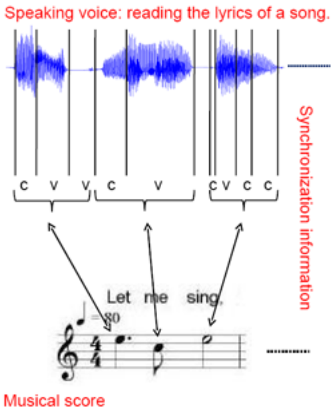
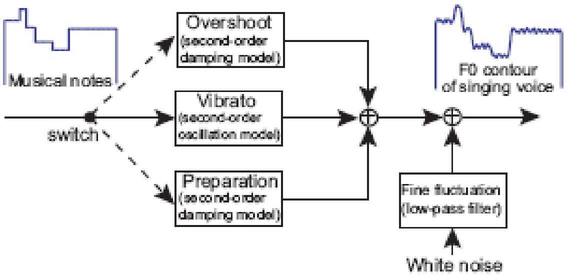

Speaking voice: reading the lyrics of a song.
c v v c v c v c c
Synchronization information
Let me sing
Musical score

Figure 15: Synchronisation information


Musical notes
switch
Overshoot (second-order damping model)
Vibrato (second-order oscillation model)
Preparation (second-order damping model)
F0 contour of singing voice
Fine fluctuation (low-pass filter)
White noise

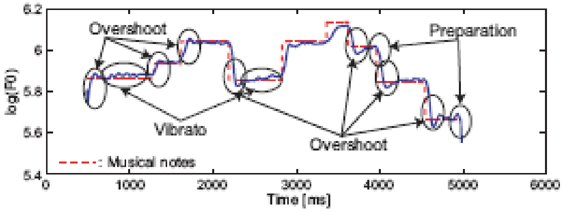


Figure 16: F0 changes

6.3 Spectral Envelope

Different than in speech voices, in singing voices a strong peak can be observed at about 3kHz, a so-called singing formant. In the conversion system, this peak is emphasised in the spectrogram. Another feature the authors implemented is an AM of the formants synchronized with the vibrato of the F0, which also occurs in real singing voices.

7 Conclusion

This paper described three very different methods of synthesizing singing voice. In particular, the underlying

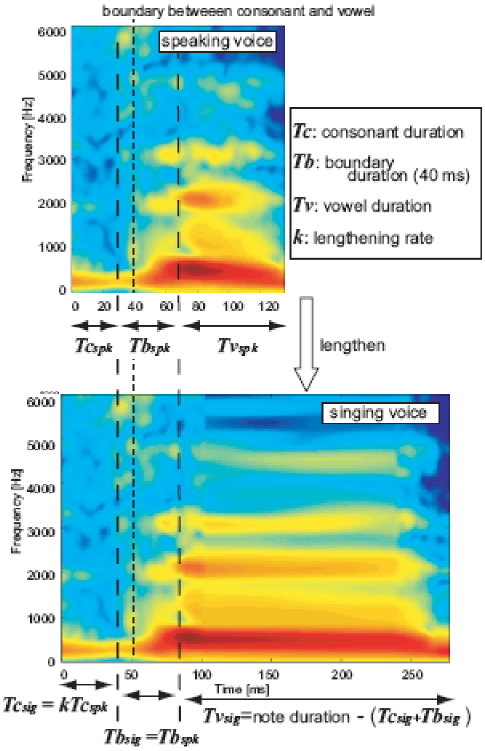


Figure 17: Original and modified spectrogram

techniques of speech synthesis were presented, together with the necessary extensions to produce singing voice.

The choice of the presented methods was made according to their relevance. From the synthesis of singing challenge 2007 [5], the first- and second-placed participants were considered as well as an example for HMM-based singing synthesis. The latter one was chosen because it can be understood as an extension of a speech synthesis system that was presented earlier.

In general, current methods show a surprisingly good performance, although there are many situations in which a still too artificial sounding output is produced. Here, the goal has to be naturalness. Therefore, typical variations in all voice parameters have to be taken into account.

References

[1] Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, Keiichi Tokuda: "An HMM-based Singing Voice Synthesis System", 2006

[2] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, Tadashi Kitamura: "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-based Speech Synthesis", 1999

[3] Peter Birkholz: "Articulatory Synthesis of Singing", 2007

[4] Peter Birkholz, Ingmar Steiner, Stefan Breuer: "Control Concepts for Articulatory Speech Synthesis", 2007

[5] `http://www.interspeech2007.org/Technical/synthesis_of_singing_challenge.php`, 2007

[6] Takeshi Saitou, Masataka Goto, Masashi Unoki, Masato Akagi: "Vocal Conversion from Speaking Voice to Singing Voice Using STRAIGHT", 2007

[7] Hideki Kawahara, Ikuyo Masuda-Katsuse, Alain de Cheveigné: "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 Commun., Vol. 27, pp. 187-207, 1998

[8] Peter Vary, Rainer Martin: "Digital Speech Transmission", Wiley, 2006