Low-Delay Singing Voice Alignment to Text

Alex Loscos, Pedro Cano, Jordi Bonada
Audiovisual Institute, Pompeu Fabra University
Rambla 31, 08002 Barcelona, Spain
{aloscos, pcano, jboni }@iua.upf.es http://www.iua.upf.es

[Published in the Proceedings of the ICMC99]

Abstract

In this paper we present some ideas and preliminary results on how to move phoneme recognition techniques from speech to the singing voice to solve the low-delay alignment problem. The work focus mainly on searching the most appropriate Hidden Markov Model (HMM) architecture and suitable input features for the singing voice, and reducing the delay of the phonetic aligner without reducing its accuracy.

1 Introduction

An aligner is a system that automatically time aligns speech signals with the corresponding text. This application emerges from the need of building large time-aligned and phonetically labeled speech databases for Automatic Speech Recognition (ASR) systems. The most extended and successful way to do this alignment is by creating a phonetic transcription of the word sequence comprising the text and aligning the phone sequence with the speech using a Hidden Markov Model (HMM) speech recognizer [1].

The phoneme alignment can be considered speech recognition without a large portion of the search problem. Since we know the string of spoken words the possible paths are restricted to just one string of phonemes. This leaves time as the only degree of freedom and the only thing of interest then is to place the start and end points of each phoneme to be aligned. For the case of aligning singing voice to the text of a song, more data is available out of the musical information: the time at which the phoneme is supposed to be sung, its approximate duration, and its associated pitch.

We have implemented a system that can align the singing voice signal to the lyrics in real time. Thus, as the singer performs, the signal can be processed and different specific audio effects applied depending on which phoneme of the lyrics is currently being sung. This pursues the idea of content based processing.

2 Singing voice to text aligner

In this section, we consider the main differences between speech and singing voice, and present our proposal for the singing voice to text aligner by searching the most appropriate HMM architecture and suitable input features for the singing voice. Finally we show how to build the composite Finite State Network (FSN) of the song.

2.1 Speech and Singing Voice

Although speech and singing voice sounds have many properties in common because they originate from the same production physiology, there are some differences to bear in mind.

-Voiced/unvoiced ratio: The ratio between voiced, unvoiced sounds, and silence is about (60%, 25%, 15%) in the case of normal speech. In singing, the percentage of phonation time can increase up to 95% in the case of opera music.

-Dynamic: The dynamic range as well as the average loudness is greater in singing than in speech. The spectral characteristics of a voiced sound change with the loudness [2].

-Fundamental frequency: In speech, the fundamental frequency variations express an emotional state of the speaker or add intelligibility to the spoken words. This frequency range of f_0 is very small compared to singing where it can be up to three octaves.

-Vibrato: Two types of vibrato exist in singing. The classical vibrato in opera music corresponds to periodic modulation of the phonation frequency, and in popular music the vibrato implies an added amplitude modulation of the voice source [3]. In speech, no vibrato exists.

-Formants: Because in singing the intelligibility of the phonemic message is often secondary to the intonation and musical expression qualities of the voice, in cases like high pitch singing, wide excursion vibratos, hoarse and aggressive attacks or very loud singing, there is an alteration of the formants position, and therefore the perceived vowel is slightly modified.

2.2 HMM Architecture

As the task of the alignment can be considered as a simplified speech recognition, it is natural to adopt a successful paradigm of the ASR, namely HMM, for the alignment. Our approach attempts to use this model for the singing voice case and tune its parameters to make the model singing voice case specific. This tuning has to take into account the following considerations:

(a) No large singing voice database is available to train the model.
(b) The final system will have to align with the minimum possible delay.
(c) The alignment will have phoneme resolution.

The aligner will be a phoneme-based system (c). In this type of systems, contextual effects cause large variations in the way that different sounds are produced. Although training different phoneme HMMs for different phoneme contexts (i.e. triphonemes) would present better phonetic discrimination, this is not recommended in the case (a) no large database is available.

HMMs can have different types of distribution functions: discrete, continuous, and semi continuous. Discrete distribution HMMs match better with small train database and are more efficient computationally [4]. Because of this and considerations (a) and (b), in this first approach, the nature of the elements of the output distribution matrix will be discrete.

The most popular way in which speech is modeled is as a left-to-right HMM with 3 states. We also fit 3 states to most of the phonemes (except for the plosives) as an approach to mimic the attack, steady state and release stages of a note. The plosives are modeled with 2 states to take into consideration somehow their intrinsic briefness, and the silence is modeled with 1 state as it is in speech.

2.3 Front-end Parameterization

The function of this stage is to extract the features that will be used as the observations of the HMMs. To do so the input signal is divided into blocks and from each block the features are extracted. For the singing voice we keep the speech assumption that the signal can be regarded as stationary over an interval of a few milliseconds. Various possible choices of vectors together with their impact on recognition performance are discussed in [5]. Our choice of features to be extracted from the sound in the front-end is the following:

Mel Cepstrum: 12 coefficients
Delta Mel Cepstrum: 12 coefficients
Energy: 1 coefficient
Delta Energy: 1 coefficient
Voiceness: 2 coefficients

With:

Window Displacement: 5.8 ms
Window Size: 20 ms
Window Type: Hamming
Sampling Rate: 22050 Hz

To compute the Mel frequency cepstral coefficients (MFCC) the Fourier spectrum is smoothed integrating the spectral coefficients within triangular frequency bins arranged on the non-linear Mel-scale. The system uses 24 of these triangular frequency bins (from 40 to 5000 Hz). In order to make statistics of the estimated speech power spectrum approximately gaussian, log compression is applied to the filter-bank output. The final processing stage is to apply the Discrete Cosine Transform to the log filter-bank coefficients.

The voiceness vector consists of a Pitch Error measure and the Zero Crossing rate. The Pitch Error component is a byproduct from the fundamental frequency analysis, which is based on [6]. The zero crossing rate is calculated by dividing the number of consecutive samples with different signs by the number of samples of the frame.

The acoustic modeling assumes that each acoustic vector is uncorrelated with its neighbors. This is a rather poor assumption since the physical constraints of the human vocal apparatus ensure that there is continuity between successive spectral estimates. However, considering differentials to the basic static coefficients greatly reduces the problem. This differential ponders up to two frames in the future and two frames in the past.

2.4 Composite FSN

The alignment process starts with the generation of a phonetic transcription out of the lyrics text. This phonetic transcription is used to build the composite song FSN concatenating the models of the phonemes transcribed.

The phonetic transcription previous to the alignment process has to be flexible and general enough to account for all the possible realizations of the singer. It is very important to bear in mind the non-linguistic units silence and aspiration as their appearance cannot be predicted. Different singers place silences and aspirations in different places. This is why while building the FSN, between each pair of phoneme models, we insert both silence and aspiration models. In the transition probability matrix of the FSN, the jump probability a_{ij} from each speech phonetic unit to the next silence, aspiration or speech phonetic unit will be the same as shown in figure 1.

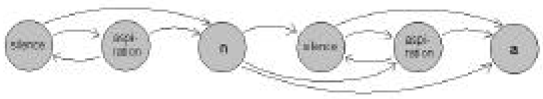

Figure 1: Concatenation of silences and aspirations in the FSN

The aspiration is problematic since in singing its dynamic is more significant. This causes that the aspiration can be easily confused with a fricative.

Moreover, different singers not only sing differently but also, as in speech, pronounce differently. To take into account these different pronunciations we modify the FSN to add parallel paths as shown in figure 2.

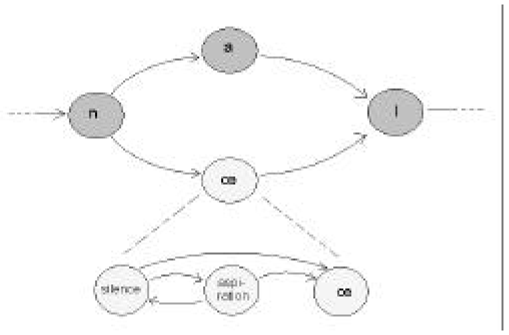

Figure 2: Representation of a phonetic equivalence in the FSN

The alignment resultant from the Viterbi decoding will follow the most likely path, so it will decide if it is more probable that it was phoneme [a] or phoneme [œ] the one sang.

3 Low delay alignment

In this section we modify the Viterbi algorithm to allow low-delay. To compensate the loss of robustness caused by this, some strategies on discarding phony candidates are introduced to preserve a good accuracy.

3.1 Low-delay Viterbi decoding

The usual decoding choice in the text to speech alignment problem is the Viterbi algorithm [7]. This algorithm gives as a result the most probable path through the models, giving the points in time for every transition from one phoneme model to the following.

Most applications perform the backtracking at the end of the utterance. In the case of a limited decoding delay, backtracking has to be adapted in order to determine the best path at each frame iteration. If we consider a decoding delay of m frames, we will have to follow the backtracking pointers of the selected best path to determine the associated phone index in the FSN m frames before. Strategies for low delay backtracking are discussed in [8] for the analogous case of recognition.

In general, the best path at frame m will be different from the best path at the end of the utterance. As a general rule, the reduction in the delay causes an important degradation in performance. However, since we want to be able to offer real time audio effects, we will work with the most extreme case, deciding for each input frame the current singer position in the lyrics with a decoding delay of m=0. To avoid a large amount of jumps from one path to a complete different one, we introduce some strategies.

3.2 Strategies for discarding candidates

During the low-delay alignment we have several hypothesis on our location in the song with similar probability. We use heuristic rules as well as musical information from the score to discard candidates.

An example of a rule for discarding candidates is that once we have decided we are in a certain fricative of the phonetic transcription, since the fricatives are aligned very reliably, the only candidates we consider are the fricative and the phonemes that comes next in the phonetic transcription.

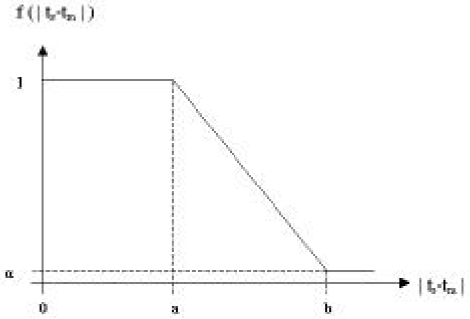

Figure 3: Function of the factor applied to the Viterbi probability

We have also implemented routines that use the information that we have apart from the lyrics. Since we are aligning to a song, we know that the phoneme corresponding to a note in the score is supposed to have certain duration. Moreover, the user, supposedly, sings following the tempo so we take advantage of this fact to better choose a phoneme from the phonetic transcription by modifying the output Viterbi probabilities by the function shown in figure 3.

In this figure 3 t_s is the time at which the phoneme happens in the singer performance, t_m is the time at which this phoneme happens in the song score, parameters *a* and *b* are tempo and duration

This function can be defined differently for the case in which the singer comes from silence and attacks the beginning of a word, and for the case the singer has already started a word, due to the very different behaviors of these two situations.

4 Results

The aligner has been tested over a set of songs and it has proved to be quite accurate and robust for all kind of singers. In order to check the performance of the system, we have implemented a graphical interface where the results of the alignments can be displayed as shown in figure 4.

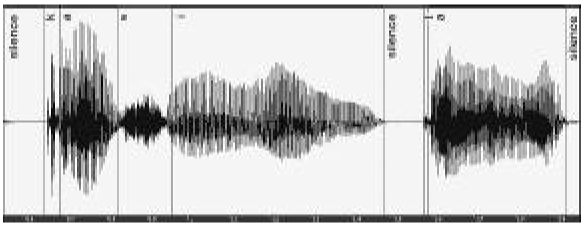

Figure 4: View of the real-time alignment results in the graphical interface

The Time Delay (TD) of the system has been computed from the formulation done in [8], in which only the intrinsic delay of the alignment algorithm is taken into account. Therefore, the following delays are considered: 10 ms due to the Window Size (WS), and 11.6 ms due the Window Displacement (WD) and the Delta Frames (DF). No delay is due to second derivatives, as we are not using acceleration feature computations (AF=0), neither any delay is introduced in the Viterbi decoding step (DD=0). This is:

$$TD = \frac{WS}{2} + WD\,(\,DF + AF + DD\,)$$

This makes a delay of 21 ms, which has to be added to the hardware delay to get the total delay of the system.

5 Conclusions

Certainly the system can be improved, especially in certain phone transitions. We believe taking into consideration the pitch information could bring about some improvements. In the system, the pitch information has been discarded so that singers could be aligned regardless in how in tune they sing. However, if we can rely on the singer's pitch, this information can be very useful to improve the accuracy of the phone boundaries. We can even think of a hybrid system where two parallel alignments, phonetic and musical [9], would merge to complement each other.

We believe that using context dependent phoneme models and using non-discrete symbol probability distributions would bring better results. This is why part of our efforts have to focus on building a large singing voice database, which at this point in time is 22 minutes long.

6 Acknowledgements

We would like to acknowledge the contribution to this research of the other members of the Music Technology Group of the Audiovisual Institute.

7 References

[1] A. Waibel and K. F. Lee. *Readings in Speech Recognition*. Morgan Kaufmann, 1990.

[2] J. Oliveiro, M.A. Clements, M.W. Macon ,L. Jensen-Link and E.B.George. "Concatenation based midi-to-singing voice synthesis" *AES*

Preprint 4591, 103rd Meeting of the AES, September 1997.

[3] J. Sundberg. *The Science of Singing Voice*. Illinois Universitary Press, 1987.

[4] S. Young. “Large Vocabulary Speech Recognition: a Review”. *Technical Report*, Cambridge University Engineering Department, 1996.

[5] R. Haeb-Umbach, D. Geller, and H. Ney. “Improvements in connected digit recognition using linear discriminant analysis and mixture densities”. *Proceedings of the ICASSP*, 1993.

[6] P. Cano. “Fundamental Frequency Estimation in the SMS Analysis”. *Proceedings of the Digital Audio Effects Workshop*, 1998.

[7] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[8] J.A.R. Fonollosa, E. Batlle, J.B. Mariño. “Low Delay Phone Recognition”. EURASIP IX European Signal Processing Conference. EUSIPCO, September 1998.

[9] P. Cano, A. Loscos, and J. Bonada “Score-performance matching using HMMs”. *Proceedings of the ICMC*, 1999.