On the Construction of Unit Databanks for Text-to-Speech Systems

Vagner L. Latsch and Sergio L. Netto

Abstract— **This work deals with one stage in the development of a text-to-speech (TTS) system, which demands a great amount of time and effort, and is strongly related to the resulting speech quality: The determination of the speech-unit databank. For that matter, we present a software tool, the so-called *Editor*, integrating all major steps in the database determination in a single environment. The whole process includes recording, segmentation, and labeling of speech units to be concatenated in the time domain. The Editor includes a low-cost and precise method for determining the pitch marks, utilizing an auxiliary signal obtained from a contact (throat) microphone. For the phonetic speech labeling, we revise an algorithm for acoustic segmentation, which yields interesting results when proper operation conditions are imposed. The result is a simplified procedure for creating a complete unit database, fully integrated into a single and user-friendly system.**

Index Terms— **Speech signal processing, speech synthesis, and text-to-speech.**

I. Introduction

TEXT-TO-SPEECH (TTS) are systems that generate synthetic speech directly from text. TTS systems are commonly employed in the human-machine interface and have been applied, for example, in voicemail systems or in automatic call-center menu reading.

The whole speech production from text may be decomposed into three main sub-problems: Automatic phonetic transcription, prosody generation, and final speech synthesis. The automatic transcription converts the input text into phonetic units by employing language-dependent pronuciation rules and dictionaries. Prosody generation creates, directly from the linguistic structure on the input text, an intonation pattern for the synthetic speech. The last stage consists of generating the final speech signal, by combining the results from the two previous steps, as illustrated in Fig. 1.

Among the several existing synthesis methods, the most commonly used today is the time-domain concatenation of speech units, due its simplicity and resulting quality. In this approach, basic speech units are concatenated according to the result from the text-transcription stage. Following, the desired intonation pattern is imposed by a prosody manipulation procedure. A popular method for modifying the prosody contour in a speech signal is the so-called time-domain, pitch synchronous, overlap and add (TD-PSOLA) algorithm [1]. The TD-PSOLA is widely employed in several successful TTS systems, due to its intrinsic simplicity and low implementation cost. The TD-PSOLA, however, requires great effort on the development of the speech database, which may become the most demanding stage of the whole TTS system design.

In this context, this paper describes a software tool, the so-called *Editor*, which aggregates several components for the construction of speech-unit databanks for TTS sytems. The Editor system includes facilities that allows one to record, label, and select the speech unit in a semi-automatic fashion. In addition, the Editor includes auxiliary features for pitch-mark detection and speech reproduction. For a complete presentation of the Editor tool, this paper is organized as follows: Section II provides an overview of all Editor features. Section III and Section IV describe the pitch-mark and segmentation algorithms, respectively, built in the Editor system. Section VI closes the paper emphasizing its main contributions.

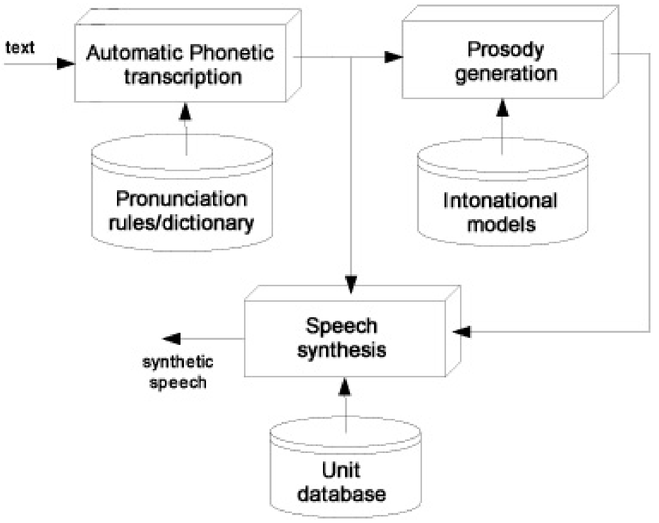

Fig. 1. Block diagram of a TTS system.

II. The Editor System

The Editor software tool includes the following functionalities: sound recording, pitch-mark detection, unit segmentation, labeling, and speech synthesis.

The recording process is made in stereo, as an auxiliary signal is captured along with the speech signal, to facilitate the pitch-mark detection, as described in Section III below. The two signals are automatically aligned in time. The obtained signals can be played back for the system user to assess the recorded quality.

After recording, a pitch detection algorithm is performed and the corresponding pitch marks are automaticaly set. These marks, however, can be hand-editted at the user's discretion if so desired.

The following stage is a voiced/unvoiced segment classification, that assists an acoustic segmentation procedure described in Section IV. Automatic labeling is performed with the phonetic content of the speech signal provided by the user.

When all stages have been properly performed, the user can add the desired speech segment to a given unit database

V. L. Latsch and S. L. Netto are with Electrical Engineering Program, COPPE/UFRJ, POBox 68504, Rio de Janeiro, RJ, 21941-972, Brazil. (emails: {latsch, sergioln}@lps.ufrj.br)

to be used in a TTS system. The recorded file also carries information relative to the pitch-mark positions and phonetic segmentation.

To validade the segmentation process, a concatenation facility has been added into the Editor system, which provides the resulting speech signal and the corresponding spectogram. At this stage, pitch and duration contours can be incorporated to the concatenated signal, using a built-in TD-PSOLA algorithm, for a more realistic validation of the whole segmentation process. If no contours are previously available, the Editor can generate them from a previously recorded speech sample, and transplant the resulting prosody model into the concatenated signal.

III. Editor Pitch-Mark Detection

The TD-PSOLA method manipulates the duration and/or pitch of a given speech signal to introduce a desired intonation pattern. The algorithm applies finite-length windows, centered at pitch marks and with typical width of two complete pitch periods. These segments are then widened or shortened, removed or repeated to modify the original signal according to a given prosody model. The speech segmental quality yielded by the TD-PSOLA is close to perfection [2]. The entire method is based on a perfect pitch-mark location, to avoid phase error in the windowing superposition, specially near the unit extremes. This section describes the pitch-mark determination algorithm incorporated to the Editor system.

Ideally, the pitch marks must be determined during voiced speech segments and can be made constantly-spaced in unvoiced segments [2]. A proper time epoch for locating a pitch mark, in voiced segments, is the glottal closure instant (GCI), where the excitation-signal energy reaches its peak [3]. The exact GCI determination based on the speech signal, however, is very erratic if made automatically or very time-consuming if made under human supervision. The whole process can be made more precise if one captures the vocal chord activity during the speech signal production [4]. This can be done by an electroglottograph (EGG) device, which is a very expensive system, in the range of a few thousand dollars. A simple but effective alternative, incorporated into the Editor tool, is to use a throat microphone to record the vocal chord vibrations, as suggested in [5], [6], although in distinct contexts. The Editor environment then uses both signals, namely the original speech and the corresponding throat signal, in parallel to determine the GCIs in a precise and automatic manner.

In our system, the throat microphone is a ceramic piezoelectric disc, commonly employed in acoustic musical instruments. The disc is mounted on a plastic velcro band, as illustrated in Fig. 2. During the speech production, the microphone must be in direct contact with the lowest part of the speaker's throat, with a reasonable but confortable pressure.

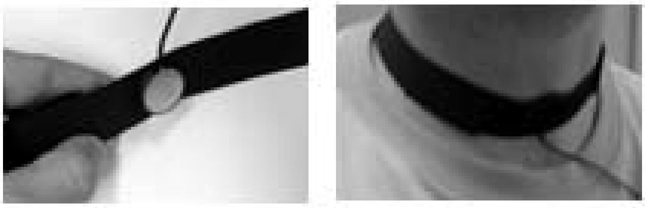

Fig. 2. Detail of piezoelectric disc fixed to a plastic band to record the vocal chord activity during speech production.

The two signals are recorded simultaneously by two distinct microphones and fed into the Editor system through the soundboard line-in input, with the assistance of a two-channel preamplifyer. An automatic delay detection, based on the correlation of the two linear-prediction residues, is performed to guarantee that both signals are in phase. The method employed in [7] is used to avoid confusion generated by different polarity between the two microphones.

The GCI estimation used by the Editor system employs the maximum likelihood epoch-detection algorithm developed in [8]. In that method, we model a voiced speech period $s(n)$ by an autoregressive linear system given by

$$\hat{s}(n) = \begin{cases} \sum_{i=1}^{p} a_i s(n-i), & 0 < n \leq \infty \\ G, & n = 0 \\ 0, & n < 0 \end{cases} , \quad (1)$$

where G is an arbitrary positive constant and p is the model order. It is then supposed that the difference between the observed signal $s(n+n_0)$, for $n \in [0, N-1]$, (where n_0 is a sequency of alignment delays), and the signal estimative $\hat{s}(n)$ is a Gaussian process. The N observations generate an N-dimensional independent Gaussian process with uniform variance σ [8].

Therefore, given $x(n) = s(n+n_0) - \hat{s}(n)$, the likelihood function is given by

$$p(X|\theta) = \frac{1}{(2\pi\sigma^2)^{N\ 2}} \exp \left\{ \frac{-\sum_{n=0}^{N-1} [s(n+n_0) - \hat{s}(n)]^2}{2\sigma^2} \right\}, \quad (2)$$

where the parameter vector is defined by $\theta = \{\sigma, a_1, a_2, a_3, \ldots, n_0\}$. A maximum of the likelihood function, or equivalently of the log-likelihood function, indicates that a GCI has occurred. In [8], the authors conclude that a closed-form expression for the opitmal n_0 can not be determined. In equation 2, however, the term $-\sum_{n=0}^{N-1} [s(n+n_0) - \hat{s}(n)]$, referred to as the maximum-likelihood epoch determination (MLED) function, dominates. To avoid false candidates, the Hilbert transform is applied to the MLED function, as suggested in [8].

An example of the GCI determination, for the voiced consonant /Z/ in the word /meZmu/ (in Portuguese), using the above method is shown in Fig. 3. The upper and lower plots depict the throat-microphone and the standard speech signals, respectively, along with the corresponding pitch-marks obtained from the MLED function.

IV. Editor Acoustic Segmentation

In addition to the pitch marks, the TTS prosody module requires a proper determination of the phonetic frontiers for all databank speech units. This information can be used, for instance, to alter a phoneme duration, according to a given prosody pattern. Determining these marks is a tedious and time-consuming process. If the speech unit is the phoneme, the segmentation or labeling must be very precise. When the speech unit is larger, however, as in the case of diphones, triphones, and so on, which preserve the phone transitions (alophones), the process serves only as a guide to a more accurate procedure, and it does not require complete precision [9].

The Editor tool includes a semi-automatic labeling method based on the segment clusterization of the signal acoustic

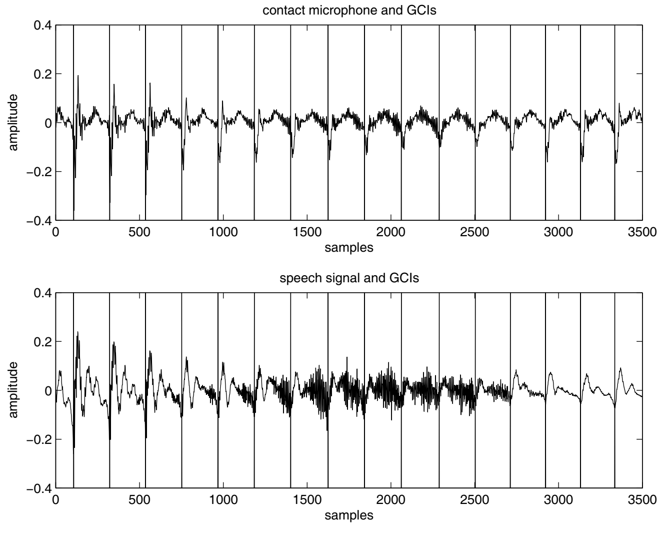

Fig. 3. Editor GCI determination using MLED function. Notice how the throat-microphone (upper) signal yields a better GCI determination, when compared to the original speech (lower) signal.

characteristics. This is equivalent to determining a codebook via a vector quantization procedure, subject to the restriction that all vectors within a given cluster must be consecutive in time [10].

Consider a N-frame signal, composed of $N_s \leq N$ clusters of statistical characteristics, defined by $(N_s - 1)$ transition times in between the distinct clusters $\{n_1, n_2, ...n_{N_s-1}\}$, with $n_0 = 0$ and $n_{N_s} = N$ by definition. The main goal is to find these transition times such that a given distortion function is minimized within a cluster. For that purpose, we use the Euclidean distance between the cepstral-coefficient vectors of all frames in a cluster and the respective centroid vector [10], [11]. Hence, in a given cluster i, defined by the interval $[n_{i-1}, n_i - 1]$, the distortion function is determined as

$$d_i[n_{i-1}, n_i - 1] = \sum_{n=n_{i-1}}^{n_{i-1}} (\mathbf{c_n} - \mathbf{c_i})^{\mathrm{T}}.(\mathbf{c_n} - \mathbf{c_i}), \qquad (3)$$

where $\mathbf{c_n}$ is the cepstrum vector for the nth frame within the cluster and $\mathbf{c_i}$ is the corresponding centroid cepstrum vector. The whole clusterization problem becomes finding the N_s clusters that minimize [12]

$$D_{N_s}(N) = \min_{\substack{n_1\ n_2\ ...n_{N_s-1} \\ n_0=0\ n_{N_s}=N}} \sum_{i=1}^{N_s} d_i[n_{i-1}, n_i - 1]. \qquad (4)$$

Considering that $0 < n_1 < n_2... < n_{N_s-1} < N$, we can rewrite $D_{N_s}(N)$ as

$$\begin{aligned} D_{N_s}(N) &= \min_{\substack{n_{N_s-1} \\ n_{N_s}=N+1}} \min_{\substack{n_1,n_2,\ldots,n_{N_s-2} \\ n_0=0}} \sum_{i=1}^{N_s} d_i[n_{i-1}, n_i - 1] \\ &= \min_{\substack{n_{N_s-1} \\ n_{N_s}=N+1}} \min_{\substack{n_1,n_2,\ldots,n_{N_s-2} \\ n_0=0}} \sum_{i=1}^{N_s-1} d_i[n_{i-1}, n_i - 1] \\ &\quad + d_{N_s}[n_{N_s-1}, n_{N_s} - 1] \\ &= \min_{\substack{n_{N_s-1} \\ n_{N_s}=N+1}} [D_{N_s-1}(n_{N_s-1}-1) + d_{N_s}[n_{N_s-1}, n_{N_s} - 1]] \\ &= \min_{n_{N_s-1}} [D_{N_s-1}(n_{N_s-1} - 1) + d_{N_s}[n_{N_s-1}, N]] \,. \end{aligned} \qquad (5)$$

By imposing that $(N_s - 1) \leq n_{N_s-1} \leq N$, then $D_{N_s}(N)$ can be minimized recursively by dynamic programming for $N = (N_s - 1), N_s, \ldots, (N - 1)$. For that matter, the level-building dynamic programming (LBDP) method can be used, as proposed in [10], [11], whereas the so-caled two-level dynamic programming (TLDP) algorithm was used in [13], although in the context of continuous speech recognition. The main distinction between these two schemes is that the TLDP requires the computation of a distortion matrix for all possible segmentation forms, which was considered a major drawback. However, when the exact number of clusters is unknown, this becomes a positive feature, as it reduces the overall computational cost, thus justifying the TLDP usage in the segmentation problem, as proposed here.

The segmentation algorithm employed by the Editor system is based on the TLDP algorithm, using 16 cepstral coefficients for each speech frame obtained with a 10-ms Hanning window. For a smoother segmentation a 5-ms superposition was employed between two consecutive windows.

For a more precise segmentation, the following steps can be considered:

- The speech units should have a prosody contour as neutral as possible;
- A first level of segmentation can be employed based on the voiced/unvoiced classification by the pitch determination algorithm;
- The number of clusters can be overestimated at first, which can be interpreted as a resolution increase in the vector quantization procedure. The final selection of transition times can be made in a supervised manner, which will be a simple procedure, since the proper frontier will be chosen from a pre-defined small group of candidates. In [10], it is suggested that a 75% supersegmentation is sufficient to guarantee a 99% match, within 30 ms, between the semi-automatic and a complete hand-made segmentation.

Fig. 4 illustrates a phonetic labeling example for the logatom *pavota* (in Portuguese). In this figure, only the indicated marks have been included by the complete segmentation algorithm. The other marks were inserted by the voived/unvoiced classification. The spectogram also included in Fig. 4 indicates the good match achieved by the Editor system in properly labeling this word.

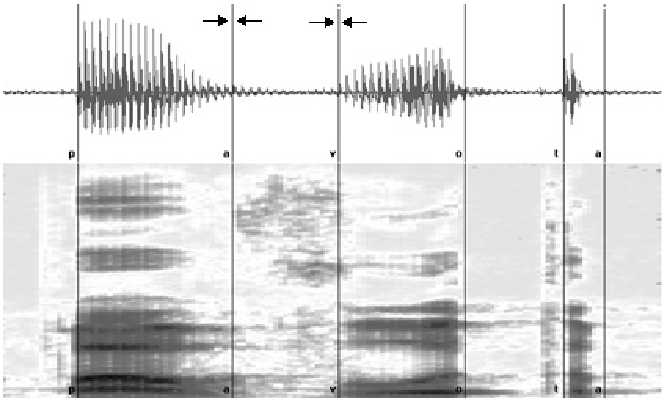

Fig. 4. A phonetic labeling example of the utterance *pavota*. The top figure depicts the speech signal with the resulting phonetic marks. The indicated marks were set by the acoustic segmentation algorithm. The other marks were determined by a simple voiced/unvoiced procedure. The bottom figure shows the corresponding spectogram.

V. Editor Signal Alignment

In order to transplant a given prosody contour into a concatenated signal, which ideally has no intonation pattern, a proper alignment procedure must be performed between two speech signals. In the Editor system, this is done with a dynamic time warping (DTW) algorithm, which uses a similar cost function to the one employed by the segmentation procedure.

Fig. 5 shows the similarity matrix between a given speech signal (y axis) and a synthetic signal (x axis) including the word *transformar* (in Portuguese, /tra~SfoGmaX/). The best aligment path generated by the DTW algorithm using a type-IV path restriction as given in [14] is indicated by the darker line.

Fig. 6 then illustrates the result of the alignment procedure. The upper plot shows the original hand-labeled utterance, for a better reference. The lower plot includes the synthesized speech sample modified by the time-warping curve. After this stage, the original pitch contour can be properly implanted into the concatenated signal.

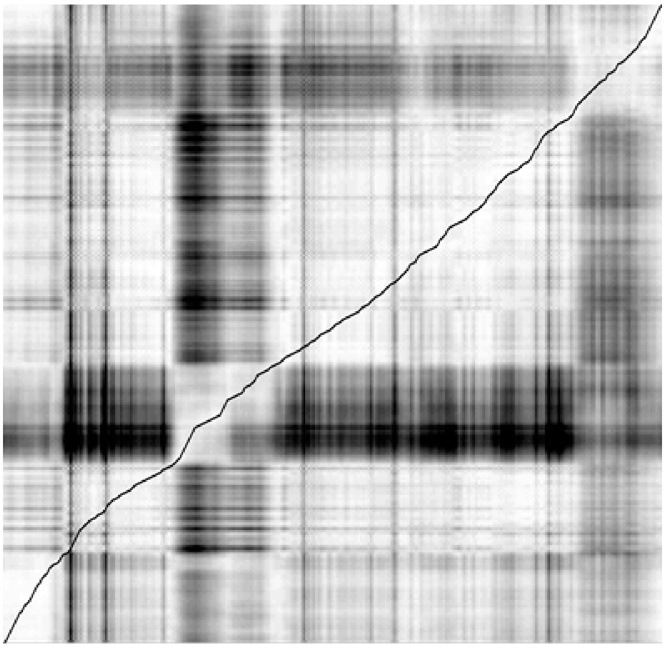

Fig. 5. Similarity matrix for DTW algorithm between recorded and synthetic utterances of *transformar* (/tra~SfoGmaX/).

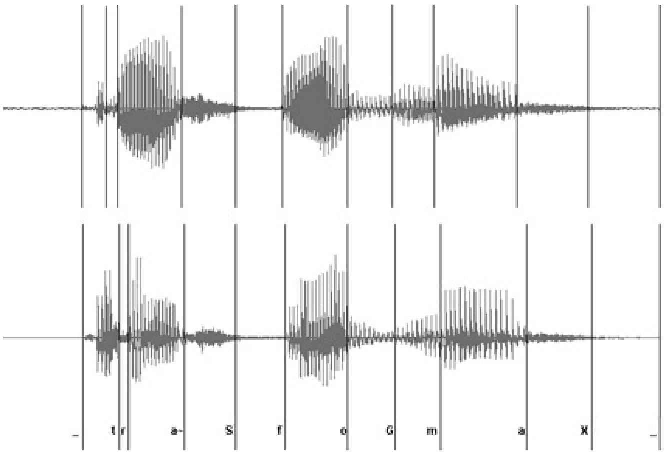

Fig. 6. Signal aligment between recorded and synthesized speech samples

.

VI. Conclusion

A complete tool for the construction of unit database of TTS systems has been presented. The so-called *Editor* system includes features such as sound acquisition, pitch-mark detection, speech segmentation and labeling, and unit recording. The whole database construction can be validated within the Editor system by a few concatenation and prosody insertion functionalities also available.

The built-in pitch-mark algorithm, using an auxiliary signal captured with a throat microphone, was described in detail. The speech-segmentation algorithm along with the similar signal-aligment procedure were also discussed. Examples were included in each section illustrating several of the Editor capabilities. The whole system is being succesfully employed to generate a diphone/poliphone TTS database for the Brazilian Portuguese language.

References

[1] F. Charpentier and E. Moulines, "Pitch-synchronous wave form processing techniques for text-to-speech synthesis using diphones," in *Proc. Eurospeech '89*, vol. 2, 1989, pp. 13–19.

[2] T. Dutoit, *An Introduction to Text-to-Speech Synthesis*. Dordrecht: Kluwer, 1997.

[3] R. Smits and B. Yegnanarayana, "Determination of instants of significant excitation in speech using group delay function," in *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 5, 1995, pp. 325–333.

[4] A. K. Krishnamurthy and D. G. Childers, "Two-channel speech analysis," in *IEEE Trans. on Acoustics, Speech, and Signal processing*, vol. 34, no. 4, 1986, pp. 730–743.

[5] M. Graciarena, H. Franco, K. Sonmez, and H. Bratt, "Combining standard and throat microphones for robust speech recognition," in *IEEE Signal Processing Letters*, vol. 10, no. 3, 2003, pp. 72–74.

[6] A. Askenfelt, J. Gauffin, and J. Sundberg, "A comparison of contact microphone and electroglottograph for the measure of fundamental frequency," in *Journal of Speech and Hearing Research*, vol. 23, no. 2, 1980, pp. 258–273.

[7] T. V. Ananthapadmanabha and B. Yegnanarayana, "Epoch extration from linear prediction residual for identification of closed glottis interval," in *IEEE Trans. on Acoustics, Speech, and Signal processing*, vol. 27, no. 4, 1979, pp. 309–319.

[8] M. Y. Cheng and D. O'Shaughnessy, "Automatic and reliable estimation of glottal closuse instants and period," in *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, 1989, pp. 1805–1815.

[9] R. E. Donovan, "Trainable speech synthesis," Ph.D. dissertation, Cambridge University Engineering Department, UK, 1996.

[10] T. Svendsen and F. Soong, "On the automatic segmentation of speech signal," in *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 1, April 1987, pp. 77–80.

[11] M. Sharma and R. Mammone, "Blind speech segmentation: Automatic segmentation of speech without linguistics knowledge," in *Proc. of ICSLP 96*, Philadelphia, PA, USA, Oct. 1996.

[12] S. Kay and X. Han, "Optimal segmentation of signals based on dynamic programming and its application to image denoising and edge detection," 2002, unpublished.

[13] C. Myers and L. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," in *Proc. IEEE Transactions on Acoustics, Speech, and Signal processing*, vol. 29, no. 2, April 1981, pp. 284–297.

[14] L. Rabiner, *Fundamental of Speech Recognition*. New Jersey: Prentice Hall, 1993.