Database-Driven Speech Synthesis Systems

Thomas Wiener, twiener@sbox.tugraz.at

November 26, 2003

1 Introduction

An important quality criterion for speech synthesizers is naturalness. Rule-based sythesis systems have turned out to produce rather robotic speech. A different approach is database-driven speech synthesis systems, which concatenate units of human speech recorded on a database.
First of all two major systems have to be distinguished:

- Reproductive Speech Synthesis
  Such systems simply concatenate and replay pre-recorded words and phrases. They are often used for synthesis when only a limited number of utterances are required (telephone services, talking toys etc.). Reproductive speech synthesis systems will not be considered in this article.

- Text-to-Speech Synthesis
  For text-to-speech synthesis, with a potentially unlimited vocabulary, the task of pre-recording all possible words for use in a word concatenation system is prohibitive and therefore some form of subword concatenation unit must be used from which all words can be constructed as required.

Demands on Database-Driven Speech Synthesizers

Besides Intelligibility and Naturalness, sophisticated synthesizers should be scalable and easily be retrainable on other voices and languages and work in real-time. The output speech quality should be consistent

2 Database building

The speech corpus has to be recorded with high quality in a controlled environment, containing phonetically rich sentences and words, in order to contemplate as many different phonetic contexts (left and right) as possible. This task is extremely time-consuming and requires a profound linguistic knowledge, an experienced, consistent speaker and a reconstructable environment, because if re-recordings are necessary, exactly the same conditions will have to be provided (time of day, arrangements, settings etc.). Ideally, the training text should reflect the task for which the sythesizer is to be used.
Additionally, some systems require the recording of the laryngograph signal. A laryngograph is a device which measures the impedance between two electrodes held to the neck either side of the larynx. The obtained signal represents the

movement of the vocal cords. This will be used for pitchmark extraction for pitch-synchronous resyntesizing techniques (LPC, PSOLA).

3 Units

There is a great variety of different segment-sizes used as basic units for concatenation. Each of them has benefits and drawbacks. Usually a compromise between memory and linking points has to be made.

Phones (about $40 - 50$)

Diphones (about 50^2)
Describe a period of time between the middle of a phone and the middle of its successor. The unit endpoints are often in regions of relative spectral stability. These regions are often spectrally similar for a given phone across phonetic contexts, and therefore diphones concatenate relatively smoothly. In German more than 75% of the possible diphones are actually used in speech.

Triphones (more than 10000 in English)
Phone with the immediate left and right context

Demi-syllables (about 5500 in German)

Syllables (about 11000 in German)

Words Weak co-articulation effects, but extremely (too) large database

Half-phonemes

Non-uniform units

4 Concatenation Synthesis

Traditionally, concatenative speech synthesis systems use a set of synthesis units all of the same type. In English and other European languages, the diphone (or dyade) is often the unit of choice. It is assumed, that co-articulatory effects never go over more than two phones. The diphones used in the synthesis systems are typically segmented by hand from nonsense words specially prepared to contain the required units. Because the database usually contains just one sample of each diphone, the obtained diphones never correspond exactly to the ones to be produced, in terms of intonation and duration. This makes prosodic manipulation inevitable. Another problem, caused by the limited number of units are annoying discontinuities at concatenation points. Smoothing algorithms (very often this is based on a linear interpolation of the spectrum envelope) are used in order to remove these mismatches in the spectrum envelope. As digital storage problems eased, and concatenative systems were more thoroughly researched, augmented diphone systems were introduced. In these systems, longer polyphone units were used to improve concatenation smoothness in those contexts in which diphones joined least smoothly.
Concatenative synthesizers, however, have a fixed inventory, and cannot reasonably be made to produce anything outside their pre-defined vocabulary. But this might be necessary to articulate a foreign word in a given text.

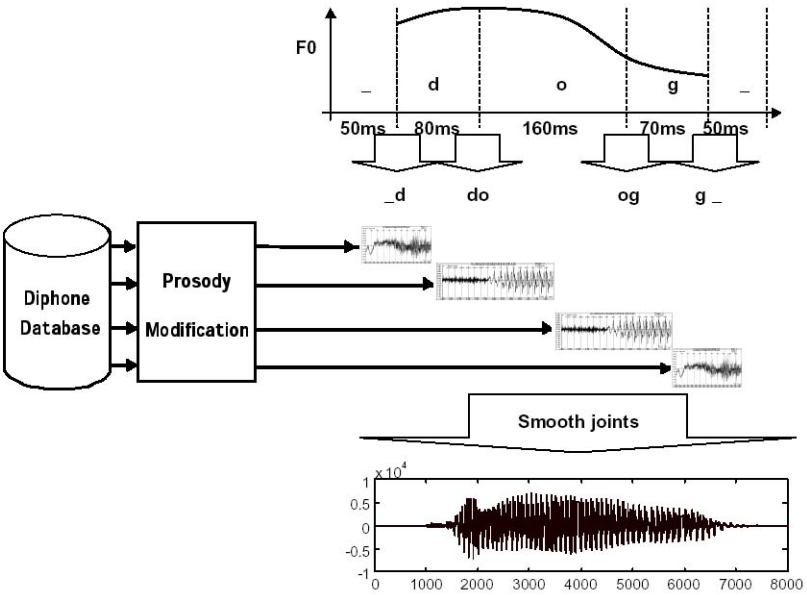

Figure 1: Synthesis of the word "dog" by diphone concatenation

5 Automatic Unit Selection

Unit Selection Systems use large, single-speaker speech databases, which do not contain just one instance of each unit but as many as possible. The greater variability in such natural speech segments allows closer modeling of naturalness and differences in speaking styles, and eliminates the need for specially-recorded, single-use databases. With the greater variability, however, comes the problem of how to select between the many instances of units in the database.

5.1 CHATR (ATR, Japan)

5.1.1 Database Analysis

Each instance of a unit in the database is labelled with a vector of features, which may be discrete or continuous. Typical features are phoneme label, duration, power, and F_0. Also, acoustic features such as spectral tilt are included in some of the databases. Other features describe the context of the unit: phoneme labels of neighbouring units, position in phrase, or direction of pitch/power change, etc. Distance measures between features of the same type have to be possible.
The next step is the clustering of similar units and distance measures between units in each cluster. To reduce the database size, the clusters are pruned. This has two effects. The first is to remove spurious atypical units which may have been caused by mislabelling or poor articulation in the original recording. The second is to remove those units which are so common that there is no significant distiction between canditates.

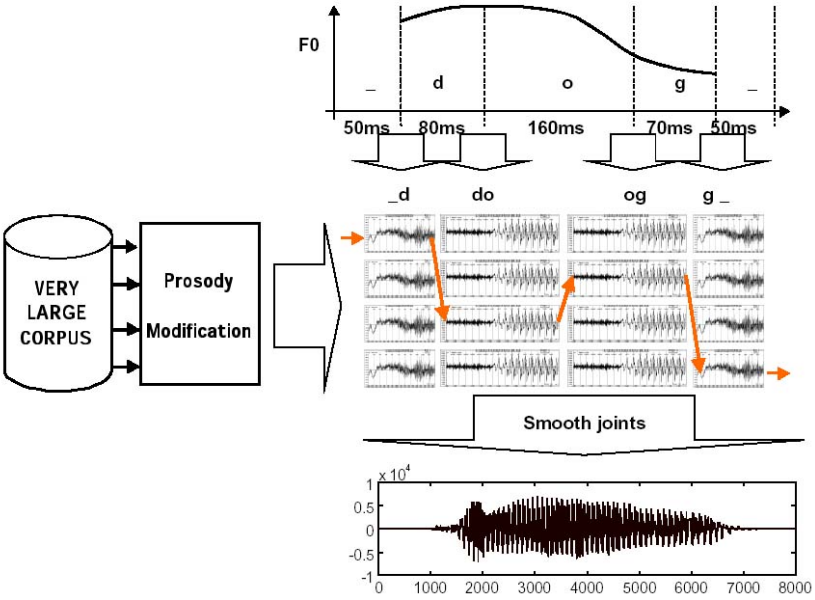

Figure 2: Synthesis of the word "dog" by unit selection from a large database

5.1.2 Synthesis

The first stages of synthesis transform the input text into a target specification (or simply target). The target for an utterance defines the string of phonemes required to synthesize the text, and is annotated with prosodic features (pitch, duration and power) which specify the desired output in more detail.
The selection of the unit to concatenate is based on two cost functions. The

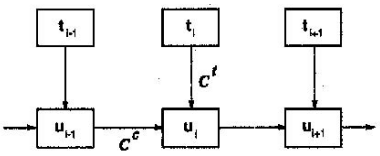

Figure 3: Unit Selection Costs

target cost $C^t(u_i, t_i)$ is an estimate of the difference between a database unit u_i and a target unit t_i. The concatenation cost $C^c(u_{i-1}, u_i)$ is an estimate of the quality of a join between consecutive database units u_{i-1} and u_i. The database can be considered as a state transition network with each unit in the database represented by a seperate state. Because any unit can potentially be followed by any other, the network is fully connected. Given the target specification $t_1^n = (t_1, ..., t_n)$, the task of waveform synthesis is to find the path through the state transition network, i.e. the sequence of database units $u_1^n = (u_1, ..., u_n)$, with the minimum cost. Each target phoneme and each canditate (database) phoneme is characterised by a p-dimensional feature vector. ($20< p <30$) The

target cost is calculated as the weighted sum of the differences between the elements of the target and the canditate feature vectors. The target cost, given weights w_j^t for the sub-costs, is calculated as follows:

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i)$$

The concatenation cost is also determined by the weighted sum of q concatenation sub-costs. (q =3). The concatenation cost, given weights w_j^c for the sub-costs, is calculated as follows:

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i)$$

As a special case, if u_{i-1} and u_i are consecutive units in the synthesis database, then their concatenation is natural and therefore has a cost of zero. This condition encourages the selection of multiple consecutive phonemes from the database. The total cost for a sequence of n units is the sum of the target and concatenation costs:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad (1)$$

where S denotes silence, and $C^c(S, u_1)$ and $C^c(u_n, S)$ define the start and end conditions given by the concatenation of the first and last units to silence. Expanding Equation 1 to include the sub-costs we obtain the following:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n}\sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) + \sum_{i=2}^{n}\sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad (2)$$

The unit selection procedure is the task of determining the set of units $\bar{u}_1^n$ so that the total cost defined by Equation 2 is minimised:

$$\bar{u}_1^n = \min_{u_1,\ldots,u_n} C(t_1^n, u_1^n) \quad (3)$$

5.1.3 Search Algorithm

Given a set of targets representing the utterance to be synthesized, for each target segment units in the database with low phonetic distance from the target are identified. Next we find the target costs for these units and prune this list taking the m best ones (m is typically between 20 and 50). Next the concatenation costs between all canditates from the previous target are found. This list is pruned to the n best costed pairs (n is typically 20-50-but need not necessarily equal m). Pruning appears to have little effect on the output quality.

5.1.4 Training the cost functions

The most complex issue to be adressed is the training of the weights of the cost functions. Although these weights can be tuned by hand, a more systematic method of tuning these produces better results. One approach is to assume a

set of weights, determine the best set of units from the database, sythesize the waveform and determine its distance from the natural waveform. This process is repeated for a range of weight sets and for multiple utterances. The best weight set is chosen as the one that performs most consistently across the utterances.

5.2 Whistler, Whisper Highly Intelligent Stochastic TaLkER (Microsoft)

Whistler is a trainable Text-to-Speech system, that automatically learns the model parameters from a corpus. Both prosody parameters and concatinative speech units are derived through the use of probabilistic learning methods. To segment the speech corpus, *Whisper* Speech-Recognition is used. This system provides automatical allignment of input waveforms to phonetic symbols and associating them to HMMs states. Whistler uses decision tree based senones as the synthetic units. A senone is a context-dependent sub-phonetic unit which is equivalent to a HMM state in a triphone. The senone decision trees are generated automatically from the analysis database to obtain minimum within-unit distortion. The decision-tree is a binary tree with a categorial question associated with each branching node.
Whistler is said to be easily retrainable.

References

[1] A.W.Black and P.Taylor. Automatically Clustering Similar Units for Unit Selection in Speech Synthesis. In EUROSPEECH '97 Rhodes, Greece, Sept.1997

[2] A.W.Black and N.Campbell. Optimising Selection of Units from Speech Databases for Concatenative Synthesis. In EUROSPEECH '95, Madrid, Spain, Sept.1995

[3] X.Huang et al. Whistler: A Trainable Text-to-Speech System. International Conference on Spoken Language, Philadelphia, Okt.1996

[4] P.Vary, U.Heute and W.Hess. Digitale Sprachsignalverarbeitung. B.G. Teubner, 1998.

[5] A.W.Black and K.A.Lenzo. Building Synthetic Voices. Preliminary version, 1999-2003.

[6] `http://www.research.microsoft.com/research/srg/ssproject.aspx`