Database Driven Speech Synthesis Systems

Thomas Wiener
twiener@sbox.tugraz.at

Overview

1. Introduction
2. Database building
3. Units
4. Concatenation Synthesis
5. Unit Selection
6. Examples
7. Summary

1 Introduction 1/3

Important criterion for speech synthesis systems:

→ Naturalness

Rule-based systems vs. Concatenation of real speech-based systems

1 Intoduction 2/3

- Reproductive Speech Synthesis

  Replay of pre-recorded words and phrases

  Example: talking toys

- Text-to-Speech Synthesis

  potentially unlimited vocabulary

  subword concatenation

1 Introduction 3/3

Demands on Database-Driven Speech Synthesizers

- Intelligible
- Natural
- Scalable
- Retrainable
- Consistent quality
- Real-time

2 Database Building 1/3

- High quality
- Phonetically rich sentences and words
- Experienced speaker
- Reconstructable environment

2 Database Building 2/3

- Targeted training text
  (news for newspaper reader)

- Recording of additional information
  Laryngograph signal (used for pitchmark extraction for pitch-synchronous resynthesis techniques (LPC, PSOLA)

2 Database Building 3/3

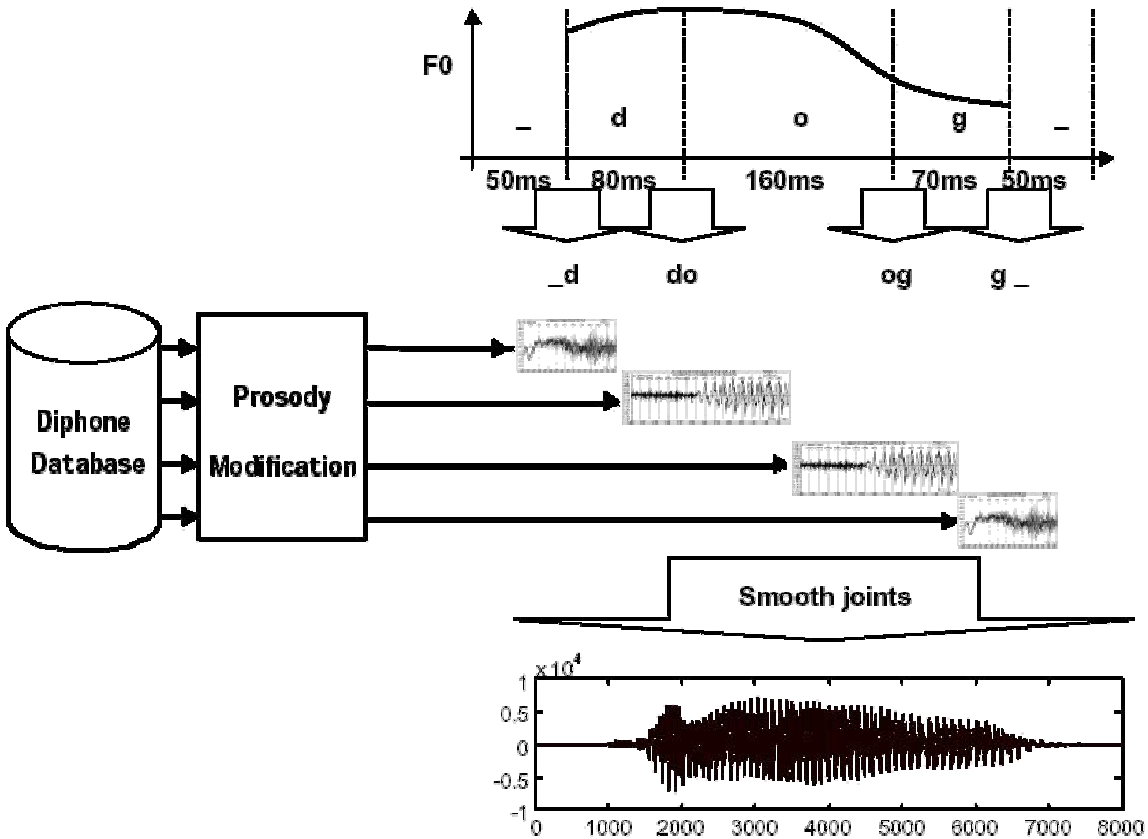
- Time-consuming
- Requires consistent speaker

3 Units

- Phones (40-50)
- Diphones (2.000)
- Triphones (>10.000)
- Demi-syllables (5.500)
- Syllables (11.000)
- (Words)

- Half-phonemes
- Non-uniform units

4 Concatenation Synthesis 1/4



4 Concatenation Synthesis 2/4

Inventory Creation

- Diphone or demi-syllable based
- Units extracted by hand from database

One instance of each unit!

→ Small inventory database

4 Concatenation Synthesis 3/4

Signal Processing

- Prosodic modification
  to match the context
- Smoothing algorithms
  for concatenation points

→ Distortion of the natural waveforms

4 Concatenation Synthesis 4/4

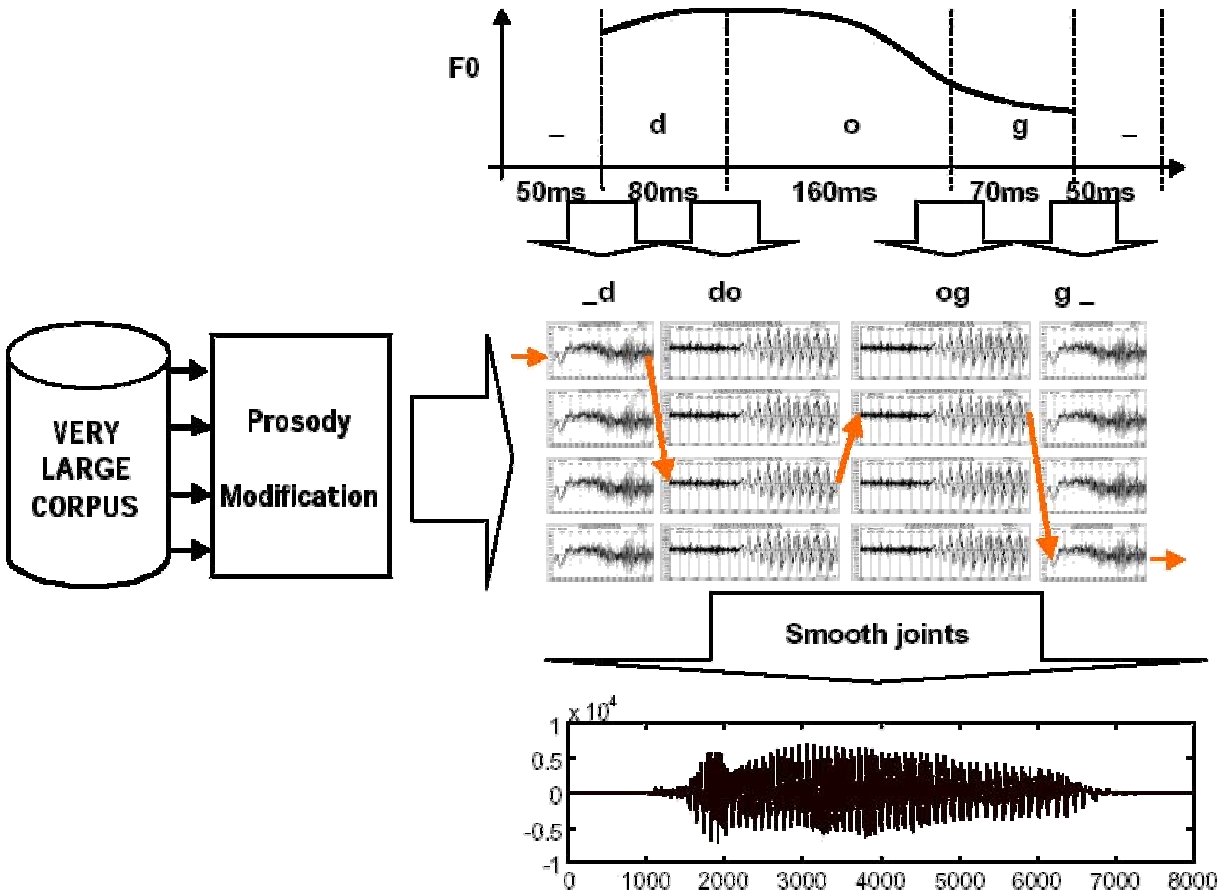
Benefits

- Consistent quality

Demerits

- Lack of naturalness
- Lack of flexibility

5 Unit Selection 1/13



5 Unit Selection 2/13

Inventory Creation

- Phone or sub-phone based
- Units extracted automatically from database

Multiple instances of each unit!

→ Large inventory database

5 Unit Selection 3/13

Signal Processing

- As little as possible!

→ Choice of context-matching and smoothly concatenable units

5 Unit Selection 4/13

Benefits

- Mostly almost natural output

Demerits

- Some very poor examples (inconsistent!)

5 Unit Selection 5/13
CHATR 1/8

CHATR (ATR, Japan)

- Phone-based
- Units represent a fully connected state-transition network
- Choice of units by means of cost-functions

5 Unit Selection 6/13
CHATR 2/8

Database Analysis

Each unit is characterised by a *p*-dimensional feature-vector

- phoneme label
- duration
- power
- F0
- characteristics of neighbouring units
- …

5 Unit Selection 7/13
CHATR 3/8

→ Clustering of similar units

→ Pruning

- Atypical units
- "Equal" units

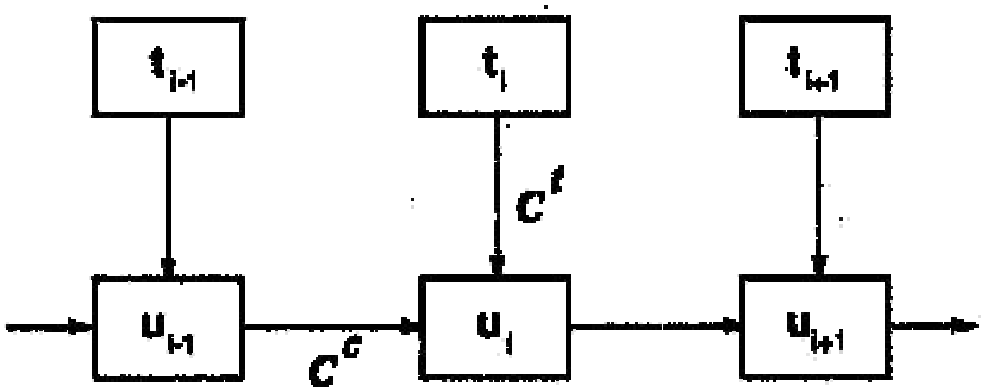
5 Unit Selection 8/13
CHATR 4/8

Synthesis

→ Based on two cost-function

5 Unit Selection 9/13 *CHATR* 5/8

Two Cost-Functions



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

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

5 Unit Selection 10/13 *CHATR* 6/8

Cost-function of a sequence of n units:

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

Including the sub-costs:

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

Goal:

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

5 Unit Selection 11/13 *CHATR 7/8*

Search Algorithm

For each target segment:

- Find units with the same name
- Compute target cost of each unit
- Prune
- Compute concatenation cost
- Prune

- Perform Viterbi-Search (beam-width 10..20)

5 Unit Selection 12/13 *CHATR 8/8*

Training the cost-functions

1. Assume a set of weights
2. Determine best set of units
3. Sythesize waveform
4. Determine distance from the natural waveform

Repeat 1-4 for a range of weight sets and multiple utterances

→ Choose the best(?) weight set

5 Unit Selection 13/13
Whistler

Whistler (Microsoft)

Whisper Highly Intelligent Stochastic TaLkER

- Sub-phonetic units: *senones*
- Probabilistic learning methods
- *Whisper* Speech Recognition System to segment the units from the database corpus
- Part of Windows 2000 and XP

6 Examples

- SVox — diphone concatenation
- CHATR — unit selection
- ≤ Whistler — unit selection

7 Summary 1/2

1. Why database-driven speech synthesizers?
2. How is a speech database recorded?
3. Which types of units can be concatenated?
4. Concatenation Synthesis
   - Inventory creation
   - Prosody modification, smoothing algorithms

7 Summary 1/2

5. Automatic Unit Selection
   - CHATR
     - Database analysis
     - Synthesis by means of cost functions
   - Whistler
6. Examples

References

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

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

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

http://www.research.microsoft.com/research/srg/ssproject.aspx