A MULTILINGUAL TTS SYSTEM WITH LESS THAN 1 MBYTE FOOTPRINT FOR EMBEDDED APPLICATIONS

R. Hoffmann, O. Jokisch, D. Hirschfeld, G. Strecha, H. Kruschke, U. Kordon, U. Koloska**

Dresden University of Technology; * voice INTER connect GmbH Dresden, Germany
Ruediger.Hoffmann@ias.et.tu-dresden.de

ABSTRACT

Text-to-speech (TTS) systems have improved their quality during the last years in large extent. This development resulted in memory requirements of several megabytes that cannot be accepted in many applications especially in embedded systems. Such applications are usually limited to a footprint of as much as 1 megabyte and are requiring the processing power being as low as possible. These requirements may be met if the text processing is changed from the usual data-driven algorithms to rule-based processing. Furthermore, the inventory should be as small as possible (diphone inventory) and should be stored in a compressed manner. This is demonstrated by a modified version of the Dresden speech synthesis system DRESS which was called microDRESS. Compared to the baseline system, microDRESS does not show essential quality losses apart from the influences of the telephone bandwidth which is appropriate for many embedded applications.

1. INTRODUCTION

This paper describes the minimized version of the multilingual TTS system DRESS which was developed to meet the general requirements for applying TTS in embedded systems. These requirements which we discussed in [1] include

- reduced computing power,
- small code size,
- optimized data flow,
- small external memory (this means especially small inventory and other speech databases),
- design for easy portability.

In [2] Sheikhzadeh et al. also dealt with low-resource synthesis basing on PSOLA technique, but they only implemented the (remarkably) small acoustic component. The introduced (complete) TTS system microDRESS meets such low-resource requirements and shows a footprint of less than 1 megabyte if equipped with one single language. In the following sections, we shortly present the baseline system DRESS and describe the procedure which was necessary to obtain the minimized result.

2. THE BASELINE SYSTEM DRESS

DRESS was originally a diphone-based TTS system which has been used for numerous applications. A detailed description is given in [3]. The system was used for a number of developments in prosody modeling which were evaluated in [4] and more recently in [5]. To be more flexible in the inventory development for different languages, the unit selection was improved to allow mixed-unit inventories and search strategies including large databases for corpus-based synthesis [6].

Table 1 describes the languages which are available in the current version. The system runs on Linux/ Windows PC and on several Unix platforms.

Table 1: Inventories for the multilingual speech synthesis with DRESS.

Language	**Units**	**Speaker**	**Inventory size** (a)
German	Corpus	1 female	58.6 MByte
German	1212 diphones	1 male, 3 female	5.0 MByte
US English	1595 diphones	1 female	7.6 MByte
Russian (b)	572 allophones	1 male	0.5 MByte
Mandarin Chinese	3049 syllables	1 male	27 MByte
Italian	1224 diphones	1 male	4.2 MByte
Czech (c)	1177 diphones	1 male	4.6 MByte
Klingon	299 allophones	1 male	1.3 MByte

(a) Values for sampling rate of 16 kHz and linear PCM (16 bit).
(b) Cooperation with Belarus Academy of Sciences.
(c) Cooperation with Czech Academy of Sciences.

3. DESIGN OF microDRESS

3.1. Modular structure

The baseline system has been designed with numerous different modules to assure flexibility for experiments and different applications. To reduce necessary data traffic, this structure was optimized to a small number of modules and databases. The final architecture is shown in Figure 1.

3.2. Symbolic processing

3.2.1. Overview

Figure 2 presents the components for processing at symbolic level. At first, the input text is split into sections. The sentence boundary detection determines the type of all sentences.

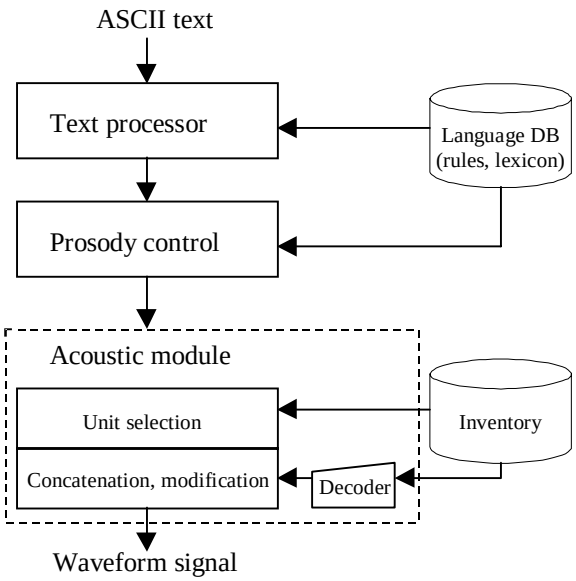

Figure 1: Block diagram of the TTS system microDRESS.

The rules for processing numerical formats and text anomalies like abbreviations and names are tried first, to generate a phonemic form of the input text. On all remaining text parts, a tokenizer is applied. By means of a special lexicon, function words are processed. If the process was not yet successful, the abbreviation lexicon is applied to the tokens. The phonemic rule set and afterwards the spelling module are applied to convert the input text. Finally, phrase boundary detection yields a finer subdivision of the sentences into prosodic phrases.

This overview emphasizes that rule-based processing is preferred wherever it is possible, instead of making widespread use of memory-consuming dictionaries.

3.2.2. Rule-based transcription of text anomalies and numbers

Usually, text contains numbers and related special formats (date and time formats) that are organized according to language-dependent rules. For the slim approach, a flexible algorithm was developed where numbers and special formats can be expressed as sequence of numbers, formatting symbols and words. Analysis rules check the match of their templates with symbol chains of variable length from the input text and generate – in case of success – a phonemic transcription for the special format.

The rule creating the following example expands a simple German date. The leftmost part splits the date into components with reference to the parentheses where components are numbered \1, \2 etc. according to their place of occurrence.

```
{date1}:
([0-3]?[0-9]).([0-1][0-9]).([12][09][0-9] [0-9])
1 <= \1 <= 31; 1 <= \2 <= 12; 1900 <= \3 <= 2199
{ord}(\1) {ord}(\2) {card}(\3)
```

The second part of the rule checks the format of the components and ensures that a number greater than 31 is not transcribed as a day. Finally, the components of the date are converted to their pronunciation form by the third rule part by calling ordinal and cardinal rules.

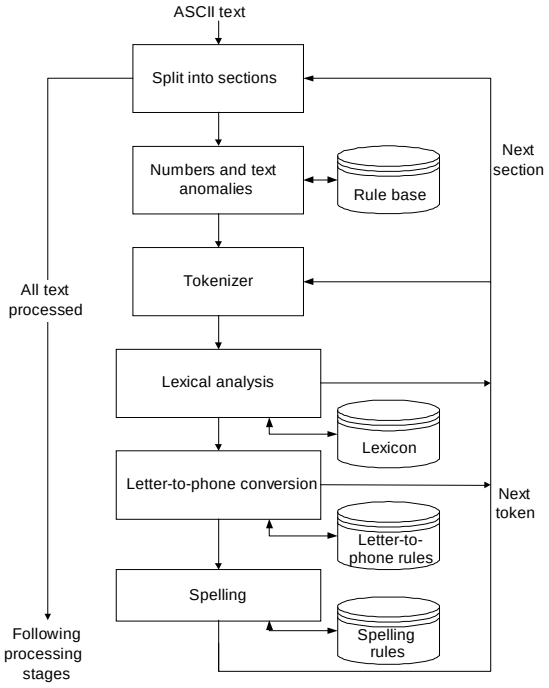

Figure 2: Symbolic processing in microDRESS**.**

3.2.3. Phonetic rule base

The construction of the phonetic rule base follows an approach described in [7]. All phonetic rules are organized in graphemic prefix, rule body, suffix and a phonemic result.

```
prefix rule_body suffix result [# comment]
```

The rules are used as follwos: `result` is the sequence of phoneme symbols the grapheme sequence `rule_body` is substituted with, if the preceding symbol sequence `prefix` and the subsequent symbol sequence `suffix` are matching as well. Prefix and suffix can be empty.

The rules are ordered from the most special to the most general case and are grouped alphabetically to subsets after the first symbol(s) in the `rule_body`. For rule processing, all subsets are searched for matching symbols. Afterwards the rule processing is initiated with the first rule of the according subset and is proceeded in a strict sequential order. Whenever a rule is found were rule body as well as prefix and suffix are matching, the rule result is added to the resulting phone string and the rule processing is aborted. The pointer to the grapheme sequence is updated shifting it by the length of the rule body. The transcription of the word rest is re-started again.

Rules are formulated in lowercase letters. Symbol `@` acts as a placeholder for any letter, `#` denotes a word boundary. Uppercase letters stand for substitutions which are defined at the beginning of the rule set. In German, for instance `V` stands for a vowel from the set `(a, e, i, o, u, ä, ö, ü)`, `I` represent a verbal ending from the set `(etest, eten, etet, test, tet, est, ete, en, et, e)`. If an uppercase letter is detected, all substitutions are checked for matching before the next letter from the prefix (or suffix) is analyzed.

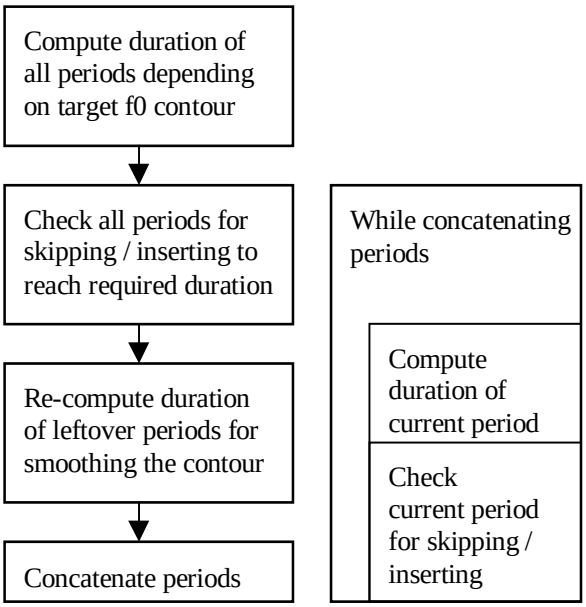

Figure 3: Conventional algorithm for prosodic manipulation (left) versus optimised version used in microDRESS (right).

3.3. Acoustic processing

For obtaining an efficient acoustic component for the TTS system, the universal unit selection algorithm of DRESS was minimized. However, it is still able to handle different unit sizes. Therefore, microDRESS can process mixed inventories.

Furthermore, the algorithm for the prosodic manipulation was optimized essentially. We give a short overview because the algorithm will be published in more details elsewhere.

The prosody manipulation applies fundamental frequency (f0) contour and duration of phonemes to the speech signal. Basically, both mechanism behave contrary. The duration of a phoneme is the sum of the duration of its periods which are obviously changing while the fundamental frequency is manipulated. This implies that f0 cannot be modified independently of the duration. Id est, manipulating the prosody of a speech signals should be an iterative process. Both frequency and duration have to be shifted step-by-step to their target values (Figure 3, left part). The disadvantage of this algorithm is the time consumption and the necessary code size because of its sequential processing. For applications in low-resource systems the signal generation needs to speed up essentially. The new algorithm of microDRESS is shown in the right part of Figure 3.

The algorithm merges step 1 and step 2 while processing step 4. In contrast to the signal generation of conventional systems only one loop is passed through - the concatenating loop.

The advantage of this approach is an "on-the-fly" decision for skipping or inserting periods so that the algorithm does not perform a planning phase before concatenating. The resulting algorithm is remarkable fast and uses a very small code size.

3.4. Inventory compression

The reduction of the inventory size is one of the essential steps in reducing the footprint. As a first step, the bandwidth of the signals is reduced to telephone bandwidth. With regard to the halved sampling rate the size reduction amounts to 50 %. For further reduction, we evaluated different coding methods with 15 listeners [8]. The results are summarized in Table 2. The intelligibility was measured by means of 20 logatoms. The Mean Opinion Scores (MOS) are resulting from listening 15 single words and 10 sentences.

Table 2: Comparison and evaluation of different coding compression algorithms useful for reducing the inventory size.

Method	Com-pres-sion	Intelligi-bility (%)	Clearness (MOS)	Com-fort (MOS)	Free of defects (MOS)
Uncoded	1.0:1	85.8	3.83	3.51	4.00
GSM	3.2:1	84.2	3.63	3.43	3.86
ADPCM 3 bit	2.4:1	84.2	3.70	3.45	3.94
ADPCM 4 bit	2.1:1	86.0	3.88	3.57	4.03
MPEG 2/2/8*	1.8:1	86.0	3.80	3.43	4.01
MPEG 2/2/1*	14.8:1	82.8	3.53	3.19	3.72
MPEG 2/1/2*	7.0:1	85.0	3.72	3.37	3.96
MPEG 1/2/4*	3.8:1	84.6	3.68	3.45	3.99
MPEG 1/1/4*	3.8:1	85.8	3.87	3.53	3.98
* Layer / psychoacoustic model / bits per sample					

For the required data reduction in the introduced project, the authors used an ADPCM technique. Further progress will be possible using the MPEG algorithms mentioned in Table 2.

4. RESULTS

4.1. Evaluation of the GPU rules

The replacement of a 960 kByte word dictionary by a 62 kByte rule base causes certain degradation in the grapheme-phoneme conversion. With respect to the complete lexicon, a symbol accuracy of 91 % is obtained. Fortunately, the remaining 9 % include many confusions (e. g., “r”/ ”6”) that are perceptually almost irrelevant.

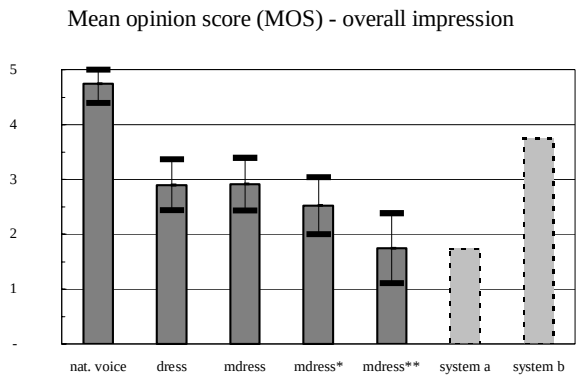

Figure 4: Evaluation of microDRESS (German database). *compressed / 16 kHz; ** compressed / 8 kHz

4.2. Evaluation of the signal quality

The most interesting aspect is the quality of the synthesized speech. For the German database, the authors performed a preliminary listening test with 10 sentences and 20 listeners. A summary of results is presented in Figure 4.

Two systems: a and b from a recent evaluation of six commercial, full-resource English TTS systems [9] are additionally presented. Systems a and b mark lower and upper quality level of the current user expectation and are considered as tolerance scheme for possible quality losses in current low-resource systems. (Nevertheless, the perceptual results of both studies can not be compared.)

The detailed results of the low-resource TTS system may be summarized as follows (see Figure 4):

- With uncompressed inventory, the quality of microDRESS (`mdress`) is not degraded compared to the baseline system DRESS (`dress`). In some cases, the rating was even better.
- Id est especially that the optimization of the prosodic manipulation described in section 3.3 does not influence the quality although one smoothing step is omitted compared to the baseline system.
- An inventory coding similar to ADPCM (`mdress*`) cause an acceptable audible quality loss.
- The additional limitation to telephone bandwidth results in a clear quality loss of almost 1 MOS degree (`mdress**`).
- Indeed, experience shows that telephone bandwidth is widely accepted by users if presented via telephone set.

Independently from low-resource considerations, the remaining gap between natural voice (MOS=4.74) and synthesis output (`mdress`: MOS=2.91) emphasizes significant research potential.

4.3. Multilinguality

All rules were implemented with a strict separation between rule interpreter (universal knowledge) and the rules base (language specific knowledge). Code and data are also strictly separated. Therefore, the different language databases from Table 1 can be exchanged without code adaptation.

4.4. Resources

The system is implemented in ANSI-C and runs on several machines and operating systems (PC with Linux/ Windows, Sun with Unix, etc.). The object code size of the whole TTS system amounts approximately to 70 kB.

With an uncompressed speech database (256 kbit/s), the system works approximately 200 times faster than real time. (Example: To produce 200 s speech signal, `mdress` took 0.955 s on standard Pentium ® 2GHz. With a compressed speech database (32 kbit/s), `mdress**` took 1.015 s on the same processor.) The PC measured time intervals contain slow read/write operations on hard disk. Using this type of compression, a total system footprint of less than 1 MByte is obtained for languages like German or English.

5. CONCLUSION

It was shown that it is possible to develop a TTS system which suffices the restrictions of low-resource systems while maintaining good quality. Some algorithmic details of the system were discussed in this paper.

The low-resource TTS system was implemented by Infineon Technologies AG and the Dresden University of Technology. The authors want to express their cordial thanks to the Speech Interface Group of Infineon Technologies AG in Munich - especially to Michael Küstner and Markus Schnell.

6. REFERENCES

[1] M. Schnell, O. Jokisch, R. Hoffmann, M. Küstner, „Text-to-speech for low-resource systems“, *IEEE Workshop Multimedia Signal Processing (MMSP)*, St. Thomas 2002 (in print).

[2] H. Sheikhzadeh, E. Cornu, R. Brennan, T. Schneider, “Real-time speech synthesis on an ultra low-resource, programmable DSP system”, *Proc. ICASSP*, Orlando 2002, vol. 1, 433-436.

[3] R. Hoffmann, “A multilingual text-to-speech system”, *The Phonetician*, 80 (1999/II), 5-10.

[4] R. Hoffmann, D. Hirschfeld, O. Jokisch, U. Kordon, H. Mixdorff, D. Mehnert, “Evaluation of a multilingual TTS system with respect to the prosodic quality”, *Proc. ICPhS*, San Francisco 1999, vol. 3, 2307-2310.

[5] O. Jokisch, H. Ding, H. Kruschke, “Towards a multilingual prosody model for text-to-speech”, *Proc. ICASSP*, Orlando 2002, vol. 1, 421–424.

[6] D. Hirschfeld, M. Wolff, “Universal and multilingual unit selection for DRESS”, *Proc. ICSLP*, Beijing 2000, vol. 1, 717–720.

[7] K. Wothke, “Letter-to-phone rules for German“, *Technical Report 75.91.04*, Feb. 1991, IBM Heidelberg Scientific Center.

[8] U. Kordon, “Options for data reduction of speech unit inventories in speech synthesis (in German)“, *Proc. KONVENS*, Ilmenau 2000, 255 – 258.

[9] Y. Alvarez, M. Huckvale, “The reliability of the ITU-T P.85 standard for the evaluation of text-to-speech systems”, *Proc. ICSLP*, Denver 2002, vol. 1, 329-332