A Flemish Voice for the Nextens Text-To-Speech System

Wesley Mattheyses, Lukas Latacz, Yuk On Kong and Werner Verhelst

Vrije Universiteit Brussel
Dept. ETRO-DSSP
Pleinlaan 2, B-1050 Brussels, Belgium

{wmatthey, llatacz, ykong, wverhels}@etro.vub.ac.be

Abstract

Nextens is an open source text-to-speech system that can be used to convert Dutch text into speech as spoken in The Netherlands. Flemish is the variant of Dutch as spoken in Flanders. These two languages have the same written form, but they sound clearly different. This paper describes how we transformed the Nextens system into a Flemish speaking application. In order to achieve this goal, a high-quality acoustic diphone synthesizer has been developed as the new back-end. This synthesizer is based on a very simple and effective overlap-add technique that can be used to simultaneously solve the problem of waveform concatenation and to perform the necessary prosodic modifications. In addition, some post-lex rules have been adapted to the Flemish speaking style. The resulting Flemish diphone synthesis system has a quality that is comparable to that of a commercial diphone synthesis system.

Flamski govor za sistem Nextens za pretvarjanje besedila v govor

Nextens je odprtokodni sistem za pretvarjanje besedila v govor. Uporabljamo ga lahko za pretvarjanje besedila v nizozemšcini v govor, kakršnega govorijo na Nizozemskem. Flamšcina je razlicica nizozemšcine, ki jo govorijo na Flamskem. Podobno kot britanska in ameriška angleščina imata ta dva jezika isto pisno obliko, zvenita pa razlicno. V prispevku je opisano, kako smo spremenili sistem Nextens v flamsko govoreco aplikacijo. Za dosego tega cilja je bil razvit zelo kakovosten akusticen difonski sintetizator kot novi zaledni del. Sintetizator temelji na zelo preprosti in ucinkoviti tehniki prekrivanja in dodajanja, ki jo lahko uporabljamo za socasno reševanje problema združevanja valovnih oblik in za izvajanje zahtevanih prozodicnih prilagoditev. Poleg tega so bila nekatera postleksikalna pravila prilagojena flamskemu nacinu govora. Kakovost dobljenega flamskega difonskega sistema za sintezo je primerljiva s kakovostjo komercialnih difonskih sistemov za sintezo.

1. Introduction

Dutch is the common name for the main language in both The Netherlands and in Flanders, the northern part of Belgium. The grammatical rules and spelling are the same for both regions, but the pronunciation of the Dutch language differs clearly between them (comparable to the difference between British and American English)[1]. We will refer to the language as spoken in The Netherlands as 'Northern Dutch', and to the language as spoken in Flanders as the Flemish language.

A text-to-speech system (TTS system) is an application that converts a written text into a speech signal. The development of such systems has been a topic of research for many years but unfortunately only few open-source TTS projects, usable for research, are available. Regrettably, no open-source TTS system has yet been developed for Flemish. Nextens (Nextens, 2006) is an open-source TTS system for Northern Dutch, which we used as a starting point for developing our own TTS system for the Flemish language.

This paper starts with a short introduction to TTS systems and Nextens in section 2. There, we also introduce our strategy for changing the Nextens voice to a Flemish sounding voice. Since the difference in pronunciation is caused by discrepancies in phonetics, we decided to record a new diphone database. To assure compatibility with the database and to permit future enhancements we also designed a new back-end. This state of the art acoustic diphone synthesizer will be described in section 3., which will be the main part of this paper. The quality of the result is evaluated in section 4. and finally the conclusions are drawn in section 5.

[1]Note that in contrast to the variants of the English language, there is only one correct spelling for both variants of Dutch

2. Strategy for adapting Nextens to Flemish

In this section we give a short summary of the different modules and functionalities found in a TTS system like Nextens, after which we explain which modules need to be replaced in order to obtain a Flemish version of the Nextens system.

2.1. A text-to-speech system

Figure 1 illustrates the different modules found in most common TTS systems. Such a system can be split-up in two main parts. The text input is first handled by a linguistic front-end, which starts by *normalizing* the input text and converting it into a set of known *tokens* (e.g., abbreviations and numbers written down with numerals are converted to plain words). Then, a *part-of-speech tagging* will take place, which delivers information about the position of the nouns, the verbs, etc. in the sentence. Hereafter, a *syntactic parsing* provides data about the inter-word relationships. All this information is used to create an accurate *prosody model* for the speech. In this part of the TTS synthesis, this model will mostly be expressed by means of 'tone-and-break indices' (ToBi), which indicate the variations in speech rate and pitch going from word to word or from syllable to syllable.

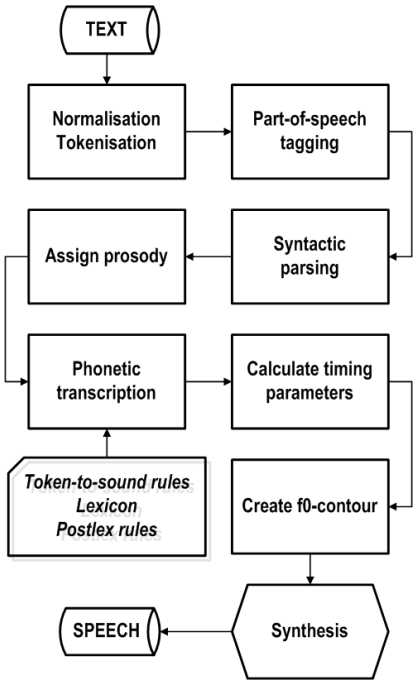

Figure 1: Overview of the different steps involved in the conversion of plain text into speech.

Obviously, the plain sentences have to be transcribed into their *phonetic transcriptions* before the corresponding speech can be generated. The system accomplishes this by using a phonetic *lexicon*, in combination with *token-to-sound* rules which are applied when the target word is missing in the lexicon (e.g., for names and foreign expressions). In the last stage, the *phonetic transcriptions* are modified by the *postlex-rules*, which are based on phone co-pronunciation properties, and the prosodic information is converted in more useful data that can be applied directly to the *synthesis* module of the TTS system. The *timing parameters* describe the optimal durations of the phonemes in the speech and the *f0-contour* indicates the most favorable fundamental frequency (or pitch) for the output signal at all time instances.

In the acoustic part of the TTS process, the prosodic and phonetic information is used as input for the synthesis module, which constructs the physical speech signal. In a concatenative system, the target speech is assembled by joining multiple sound records from a database in an appropriate way. Typical examples are the diphone systems which employ a database consisting of diphones, speech signals containing two consecutive phonemes (i.e., a diphone actually starts in a characteristic point of the first phoneme (e.g., in its most stable part) and ends in a characteristic point of the second phoneme). Nowadays, higher quality TTS synthesis can be accomplished with so-called 'non-uniform unit selection systems' which use much larger speech databases and can better account for contextual variations, for example. Nevertheless, diphone systems are still important for their possible application in small mobile devices.

2.2. Modification of the original system

The Nextens project provides the Dutch equivalent for all the front-end modules shown in Figure 1 and is equipped with an MBROLA synthesizer (MBROLA, 2006). In order to achieve a Flemish sounding output, a modification of the synthesizer will be necessary. In any case, a new database with speech recordings needs to be created and provided to the synthesis module. It is obvious that by registering a new diphone set by means of Flemish carrier words, a big step toward a Flemish TTS output is realized. Furthermore, one can opt to implement a new synthesizer in order to assure a maximum compatibility between the dataset and the used synthesis algorithms, which will undoubtedly have a positive influence on the output quality.

Since the Dutch grammatical rules and spelling apply for Northern Dutch as well as for Flemish, only minimal revision of the front-end will be necessary. The phonetic transcription of the input text is accomplished by using a language-dependent lexicon. After the lexicon-lookup, the phonetic transcriptions are handled by the 'postlex-rules'. These rules are also used to modify the transcriptions in order to attain the intended regional accent, hence a modification of some of these postlex-rules will be obligatory to facilitate the adjustment of the Nextens voice from Northern Dutch to Flemish. Note that by 'accent', we understand in this context the differences in pronunciation of the official Dutch language, which are comparable to the differences between spoken British English and American English. For the conversion of a TTS system to a real dialectic voice (where non-standard sounds, words and expressions are used), much more effort would be required (for example, changing the grapheme to phoneme conversion module would be needed, the lexicon would need major revisions, etc.).

A few examples of Northern Dutch postlex-rules that needed to be discarded for the Flemish language are shown in table 1.

Postlex-rule	Example
G-r $\rightarrow x - r$	begrip
G-l $\rightarrow x - l$	begluren
N-G $\rightarrow N - x$	mongool
l-G $\rightarrow l - x$	algebra
b-d $\rightarrow p - d$	abdij
b-n $\rightarrow p - n$	abnormaal
Z-w $\rightarrow S - w$	bourgeois

Table 1: Northern Dutch postlex-rules that were discarded to adapt the system to the Flemish speaking style.

3. A high-quality acoustic diphone synthesizer

A new diphone synthesizer has been implemented in order to achieve maximum compatibility with the new Flemish diphone database that we recorded (around 1800 recordings were included in the dataset). This section will explain how this synthesizer constructs an output speech signal by the concatenation of elements from the diphone database, followed by the assignment of the prosody defined by the

parameters delivered by the linguistic front-end. We believe that the strength of our synthesizer mainly resides in its high quality and low complexity that was achieved by using an overlap-add technique for both the segment concatenation and the prosodic modification, in accordance with the source filter interpretation of pitch synchronized overlap-add (PSOLA) (Moulines and Charpentier, 1990), as introduced in (Verhelst, 1991). As will be explained further in this section, according to this interpretation, the synthesizer can make use of the series of pitch markers that is defined for each diphone signal to fulfill the concatenation.

3.1. Pitchmarking

The pitch markers are a set of sample indices which indicate the local pitch periods in a speech signal. This implies that the distance between two consecutive pitch markers is in fact a local pitch measure for the signal. The prosody in our synthesizer will be assigned by using the pitch-synchronous overlap-add technique (PSOLA), which needs a series of good pitch markers to attain quality output. The quality of the synthesizer will thus greatly depend on the correctness of these markers. Therefore we designed an efficient and robust algorithm to accomplish this pitch epoch detection, as described in (Mattheyses et al., 2006) and summarized below.

Our algorithm is an extension of a previous technique (Lin and Jang, 2004) that is based on a dynamic programming approach applied to voiced segments. In our approach, we start by performing a frame-based voiced/unvoiced decision on the speech signal. This is necessary because unvoiced frames, due to their noise-like behavior, have to be treated differently than the voiced speech segments, which contain a clear periodicity.

In the voiced regions of the signal, the markers are systematically placed at signal peaks or at signal troughs. Note that the choice for peaks or troughs has to be the same for all the diphone signals in the database. This peak/trough decision is made corresponding to whichever minimizes an error measure between the local pitch values (obtained as the difference between consecutive pitch markers) and the global pitch contour obtained from an AMDF pitch detection algorithm.

If we assume that the markers are to be positioned at signal peaks, the algorithm continues by searching for the maximum sample present in the frame. By using the AMDF pitch measure in combination with this highest sample index, several search-regions can be defined. These correspond to those parts of the signal in which the other pitch markers in this frame can be assumed to be located. Next, a set of candidate markers is selected for each search region, based on two properties. The candidates have to represent a sample value which is as high as possible, while we also require that successive candidates are separated by a given minimum distance. This results in a set of possible markers per search region, each representing a different signal peak. In a final stage, each candidate is given a score based on its height and another score is associated with the transition between two candidates. The algorithm selects one candidate in each region as final pitch marker by maximizing the total score, summed over all selected candidates and transitions. For more details, the reader is referred to (Mattheyses et al., 2006). Figure 2 shows the first steps of the voiced pitch marking process. As illustrated in the last panel of the figure, selecting the highest candidate as the final marker would not always result in a consistent set of pitch markers, which explains the introduction of the transition scores.

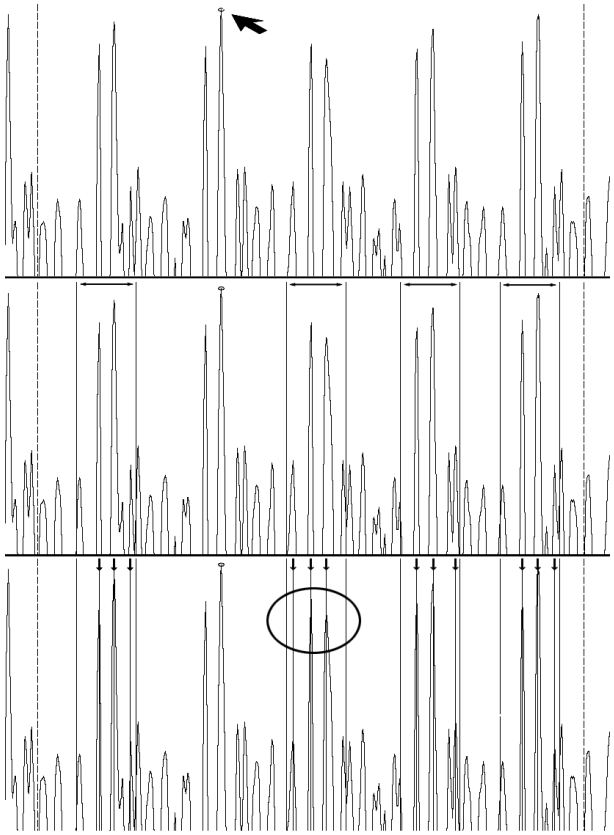

Figure 2: Finding pitch mark candidates in voiced speech. The largest peak of the signal is detected (upper panel) and search regions are defined lying at multiples of a global pitch measure from the largest peak (middle panel). The three largest peaks in each search region are the pitch mark candidates.

In contrast to many other pitch marking algorithms, which simply place the pitch marks in unvoiced signal regions at regular time intervals, we opted to position the unvoiced markers in a well-thought manner. We found this to be necessary as a frame could be classified as unvoiced, but still contain part of a voiced signal, include voiced/unvoiced transitions, etc. Such signals contain some residual periodicity, which should be indicated by the final set of pitch markers. Therefore, in the unvoiced regions of the speech signal, we determine the pitch markers by positioning them according to the neighboring voiced pitch markers. Figure 3 shows a detail of a speech signal and its trough-based pitch markers. It illustrates the correctness of the pitch marks for voiced parts of the signal as well as for unvoiced parts and for voiced/unvoiced transitions. As reported in (Mattheyses et al., 2006), the pitch marking

algorithm has been tested and evaluated and it provides a series of consistent markers, which are suitable for application in a TTS system. Note that, although not really necessary, one could also choose to hand-correct the pitch marks since pitch marking of a TTS database is done offline.

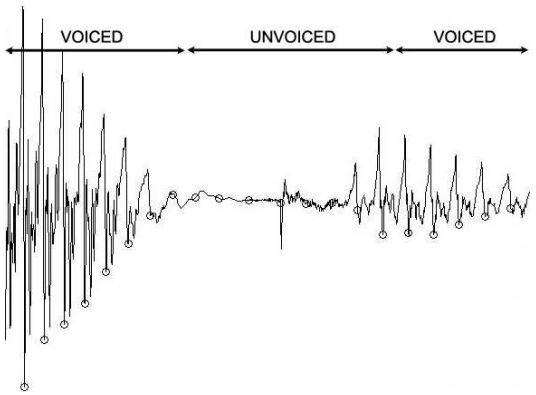

Figure 3: Open circles illustrate the result of automatic trough-based pitch marking in a transitory speech segment.

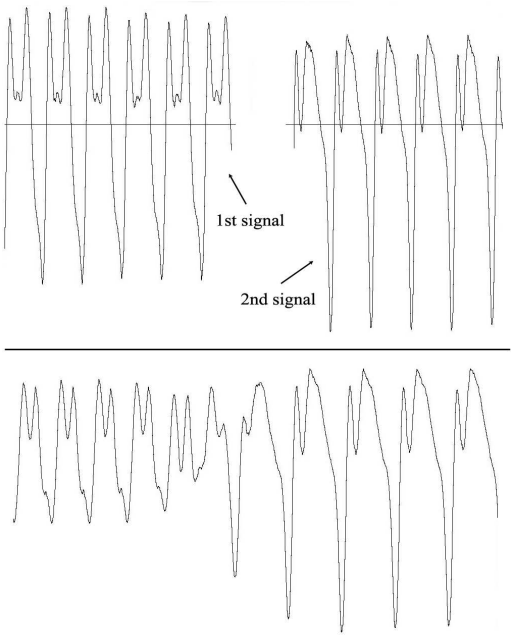

Figure 4: Illustration of typical problems that occur with straightforward non optimized diphone concatenation.

3.2. Segment concatenation

The acoustic synthesizer has to concatenate the diphone recordings in order to construct the desired speech signal. To achieve a fluent and intelligible speech, the diphones have to be concatenated in an appropriate way. Figure 4 illustrates the concatenation of the Dutch diphones 'b-o' and 'o-m'. It shows that there is a quite large dissimilarity between the two signals, although both represent the same phoneme 'o'. In an ideal diphone database, every phoneme would have been recorded at a same speech rate and having a same pitch value. It is obvious that in reality only an approximation of this ideal can be achieved. Therefore, the concatenation technique has to smooth the transition between the two signals over a certain time, otherwise these transitions will appear to be too abrupt and the concatenated speech would not sound very fluent, but chopped.

While joining two voiced speech signals, we have to make sure that the resultant signal shows a continuous periodicity. A shortcoming of many concatenation techniques is that they introduce anomalous pitch periods at the diphone transitions, which has a harmful influence on the output quality. In the second panel of figure 4, such a bad concatenation result of the 'o' phoneme is shown. As one can see, the transition between the two consecutive 'o' signals is not smooth and at the transition point abnormal pitch periods appeared.

Since we have a series of pitch markers for each diphone signal, we can exploit the benefits of the use of this pitch-information in joining the diphone segments. A diphone database contains information about the most optimal cut-points in the diphone recordings (this is referred to as the 'segmentation' of the database). This information is derived offline and obviously can not take into account exactly which two segments will be concatenated. By choosing a pitch marker as the diphone cut-point, we can assure that the periodicity of the speech signal will not be disrupted by the concatenation procedure. The most straightforward technique would be to select that pitch marker that is closest to the segmentation-cut-point as the 'cut-marker'. In order to further enhance the concatenation quality, we designed an optimization method which selects the best cut-marker according to the minimization of a MEL-scale spectral distance, as suggested in (Conkie and Isard, 1994). This technique selects for each join a pitch marker from the first and from the second diphone in such a way that the transition will occur where there is as much similarity between the two speech signals as possible.

Once the cut marks are determined, the actual concatenation problem is tackled by a pitch-synchronous window/overlap technique. First, a number of pitch periods (typically 5) is selected from the end cut-marker and from the beginning cut-marker of the first and second diphone, respectively. Then, the pitch of these two short segments is altered using the well known PSOLA technique, which will result in two signals having exactly the same pitch. The initial pitch value of these resulting signals is chosen equal to the pitch present in the original signal extracted from the first diphone. This pitch then varies smoothly along the length of the signals such that the final pitch value becomes equal to the pitch of the signal extracted from the second diphone. Finally, these two completely pitch synchronized signals are cross-faded using a hanning-function to complete the concatenation of both diphone recordings. By first assuring the pitch-synchronicity of both signals before applying the cross-fade, the introduction of irregular pitch periods is minimized and the periodicity is preserved as much as possible.

Figure 5 illustrates our concatenation method using the same diphones as in figure 4. To illustrate its robustness,

we used a first diphone recording that has a pitch value which is much higher than that of the second diphone, as one can see in the upper panel of the figure. The middle panel shows the pitch-alignment of the extracted pitch periods and the bottom panel shows the final concatenated 'o' phoneme. This last plot illustrates that in the concatenated speech signal the diphone transition is smoothed among a few pitch periods, which is necessary if a fluent output is to be obtained. In addition, the output does not suffer from irregular pitch periods.

The proposed concatenation technique delivers results of the same quality as some more complex concatenation methods found in the literature. The technique has been systematically judged against a spectral interpolation approach and it was concluded that the computationally more complex interpolation could not outperform the proposed overlap-add method. This can be explained by noting that the transition was actually realized as the result of three processes: the use of the pitch markers assures a maximum preservation of the periodicity, the pitch-synchronous overlap-add accomplishes the transition in pitch value from the first diphone to the second one, and finally the window/overlap operation creates the transition in waveform shapes between both diphones.

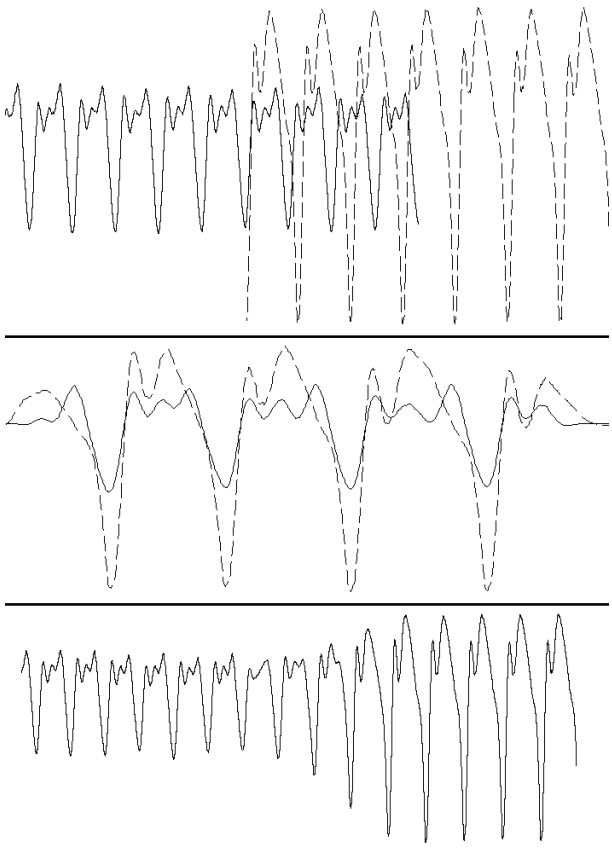

Figure 5: Pitch-synchronous concatenation. The upper panel illustrates the diphones to be concatenated, the middle panel illustrates the pitch-synchronized waveshapes, and the lower panel illustrates the result after cross-fading.

3.3. Adding prosody

At this point we need to apply the correct prosody to the concatenated signal. We opted to use the PSOLA technique to alter the timing and the pitch of the speech. During the concatenation process, the pitch markers of the synthesized speech signal can be computed from the diphone pitch markers. These will then be used as analysis-pitch markers for the PSOLA technique.

At the same time, each sample point that indicates a phoneme transition is memorized. By using these transition points the synthesizer calculates the length of each phoneme present in the concatenated signal. The Nextens front-end provides a set of timing parameters, indicating the optimal length of each phoneme in the final TTS output. Using these two sets of values, the amount of time-stretching that is necessary to provide the output speech with the correct timing properties is computed. Subsequently, the PSOLA algorithm will synthesize the output signal by using a time varying time-stretch value going from phoneme to phoneme. The synthesis-pitch markers used by the PSOLA operation determine the pitch of the final TTS output (Verhelst, 1991). Obviously, it suffices to calculate these pitch markers based on the pitch-parameters coming from the front-end (the 'f0-contour') to ensure that the imposed intonation curve is correctly assigned to the final speech signal. Note that only voiced parts of the speech require a pitch-shift. Since the PSOLA algorithm is constructing the output of a TTS system, we know at each point in time which phoneme corresponds to the current signal segment. This information can be used to decide whether a pitch-shift is desired or not.

4. Evaluation

In this section the performance of the TTS system will be discussed. First our system will be compared with the Nextens application and afterwards the overall TTS quality will be judged while possible explanations and solutions to improve the output quality will be stated. The evaluation is based on informal listening tests, conducted by people with experience in the field and by people without experience.

To achieve a Flemish TTS synthesis, our diphone synthesizer is provided with the prosodic parameters of the Nextens system. When the output of our Flemish application is judged against the original Nextens speech, we actually also compare our overlap-add synthesizer with the MBROLA synthesizer, which is resident in the Nextens system. It appears from our experiments that our synthetic voice definitely sounds as fluent as the MBROLA voice[2]. Both signals display very similar timing and pitch variations, which indicates that our acoustic synthesizer does apply the desired prosodic modifications in an accurate way. Due to the cut-point optimization, discussed in subsection 3.2., our voice is robust against small segmentation-errors of the diphone database and the pitch-synchronous concatenation technique makes it feasible for use with databases that contain inconsistent pitch levels. Further, the output of

[2]Note, however, that we could not compare our PSOLA synthesis method against the MBROLA synthesis technique using in both cases a same diphone database

our system sounds undoubtedly Flemish in contrast with to original Nextens voice which means that the main goal, the conversion of the language of the system, is achieved.

In general, the output of our Flemish TTS system is very intelligible. However, in most cases the speech possesses a sub-optimal prosody (coming from the Nextens system). The pitch variations are often to abrupt and sometimes syllable durations are to short. Especially this last imperfection can have a dreadful influence on the clarity of the output speech. We compared our TTS system with two commercial systems for Flemish, (Realspeak, 2006) and (Fluency, 2006). The first one is a system that uses a very large segment database instead of a small diphone database. As one would expect, the naturalness of its speech is much higher than with our system at the expense of a much larger footprint and computational load. These systems also achieve a higher output quality due to the presence of multiple instances of the same segment in the database. More appropriate is the comparison with the second commercial application, which is also a diphone system. The smoothness of this commercial system and the fluency of our TTS application are about the same. However, the output of the commercial system sounds more natural and is overall more intelligible than the output of our system. As mentioned before, a correct timing model is necessary to attain a highly intelligible output. The accuracy of the f0-contour has less influence on the clarity of the speech, although a precise intonation curve is needed to reach a natural sounding TTS output. The influence of a non-optimal prosodic model can be counteracted by lowering the speaking rate. However, this classic technique obviously has its limitations. To ensure enough naturalness, we suggest that the variations in the f0-contour are kept limited, since a more flat intonation will sound less disturbing than an incorrect one. It is also important to create an f0-contour with mean value around the original pitch present in the diphone recordings. This ensures that only minor pitch modifications are required from the PSOLA algorithm, which enhances the quality of the output speech. Another point of attention is the introduction of 'phrase breaks' in the speech signal. These are short pauses between two words, some of which, but not all, are determined by the punctuation. In contrast to the commercial systems, the Nextens front-end fails to predict these pauses accurately, as they are only placed according to punctuation (e.g., after a comma).

We performed some experiments in which we provided our synthesizer with a better set of prosodic parameters by manually measuring these values in the commercial TTS outputs. This resulted in speech signals of approximately the same quality as the commercial diphone system, which demonstrates the importance of the prosodic information in order to attain high-class output speech. Furthermore, we manually inserted the correct phrase breaks into the signal, which lead to an important enhancement of the clarity of the TTS output. This can be explained by noting that the extra pauses will slow down some parts of the speech and they will make it easier to distinguish between the different words in a sentence. These tests illustrated that a very good diphone TTS output is achievable by using our diphone synthesizer, provided that more optimal prosodic parameters are used in comparison to the prosody that the Nextens front-end can provide.

5. Conclusions

In this paper we discussed the conversion of a TTS system between two regional accents: Northern Dutch and Flemish. A new diphone synthesizer has been designed, which uses the PSOLA technique to impose the desired prosody on the output speech. The synthesizer also uses the PSOLA pitch markers to successfully maintain a maximum periodicity while concatenating the diphones. A cut-point optimization method proved useful to cope with small segmentation errors in the database. By combining the pitch-synchronous overlap-add technique with a simple cross-fade method, robust high quality concatenation was achieved.

The switch from Northern Dutch to Flemish was accomplished by providing a new set of diphones and a modification of some postlex-rules. Once the synthesizer produces fluent and intelligible speech, a revision of some of the linguistic modules of Nextens will be necessary in order to enhance the clarity and the naturalness of the output. The introduction of phrase breaks and the adjustment of the f0-contour can definitely contribute to achieve this goal. Our experiments have shown that high-class diphone synthesis is attainable by using our diphone synthesizer and a set of optimal prosodic parameters.

6. Acknowledgments

Parts of the research reported on in this paper where supported by the IWOIB project Link II - Voice Modification of the Brussels region, by the IWT projects SPACE (sbo/040102) and SMS4PA-II (O&O/040803), and by the research fund of the Vrije Universiteit Brussel.

7. References

A. Conkie in I. Isard. 1994. Optimal coupling of diphones. V: *Proc. SSW2 – 2nd ESCA/IEEE Workshop on Speech Synthesis*.

Fluency. 2006. http://www.fluency.nl/.

C. Y. Lin in J.S. Jang. 2004. A two phase pitch marking method for td-psola synthesis. *GETS Interbational transaction on Speech Science and Engineering*, 1(2):211–221.

Wesley Mattheyses, Werner Verhelst, in Piet Verhoeve. 2006. Robust pitch marking for prosodic modification of speech using td-psola. V: *Proc. IEEE Benelux Signal Processing Symposium, SPS-DARTS*.

MBROLA. 2006. http://tcts.fpms.ac.be/synthesis/mbrola.html.

E. Moulines in F. Charpentier. 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9:453–467.

Nextens. 2006. http://nextens.uvt.nl/.

Realspeak. 2006. http://www.nuance.com/realspeak/.

W. Verhelst. 1991. On the quality of speech produced by impulse driven linear systems. V: *Proc. International Conference on Acoustics, Speech and Signal Processing*, str. 501–504.