High quality Spanish restricted-domain TTS oriented to a weather forecast application

Francesc Alías, Ignasi Iriondo, Lluís Formiga, Xavier Gonzalvo, Carlos Monzo, Xavier Sevillano

Dep. of Communications and Signal Theory. Enginyeria i Arquitectura La Salle
Ramon Llull University, Barcelona, Spain
{falias,iriondo,llformiga,gonzalvo,cmonzo,xavis}@salleURL.edu

Abstract

A restricted domain text-to-speech system oriented to a weather forecast application is presented. This TTS system is embedded in a multimedia interactive service accessible from different media, such as TV, Internet and mobile devices. The requirements of this application give rise to several particularities in the design and implementation of the TTS system, which are discussed throughout this paper. Several tests have been conducted to analyze the TTS system in terms of performance and speech quality.

1. Introduction

This paper presents a restricted but unlimited corpus-based text-to-speech (RU-TTS) system [1, 2], which aims to obtain highly natural synthetic speech. By one hand, it is restricted as it is oriented to a weather forecast application, following a phrase-splicing approach [3]. By the other hand, it is unlimited as the speech corpus contains all the diphones (extended with some triphones) of Castilian Spanish. Thus, although the design and contents of the speech corpus are totally application-oriented, the system is capable of synthesizing *any* input text. Moreover, this system follows our previous works on multi-domain TTS [4, 5], as the corpus is structured to speed up the synthesis process, although no automatic text classification is conducted.

The RU-TTS system presented in this paper has been developed within the framework of a research project called *Virtual Characters*. The project members are the Catalan Broadcasting Corporation (CCRTV), the Interactive Technology Group of the Pompeu Fabra University (ITG) and the Department of Communications and Signal Theory of Enginyeria i Arquitectura La Salle (Ramon Llull University). The main goal of this project is the creation of an environment to allow the automatic generation of audiovisual products for different media: TV, Internet and mobile devices. These products are based on animated virtual characters with synthetic voices. As a result of this project, multimedia directors and designers are provided with a set of tools to create characters, give them movement and expression, put them into real or synthetic scenarios, furnish them with synthetic speech capabilities, etc.

Figure 1 depicts the block diagram of the Virtual Characters project, which is constituted by three main systems: *i)* the Script Generator, which defines the scheduling of the scene, the character movements and expressions, the text to be synthesized and the additional animations (developed by CCRTV and ITG); *ii)* the Text-to-Speech Generator, which synthesizes the message and incorporates the timing for event and lip synchronization; and *iii)* the 3D Scene Generator, which generates the final video adapted to the selected output device (developed by ITG).

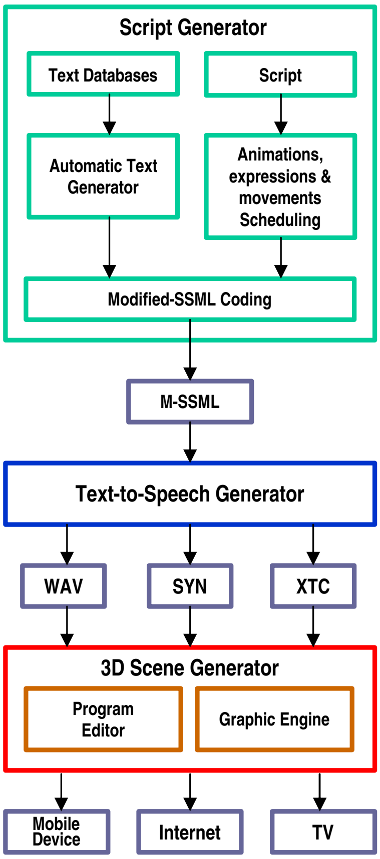

Figure 1: Block diagram of the Virtual Characters project.

As a first step towards this challenging aim, an application called *Virtual Weatherman* has been developed. A human, but cartoon-looking, virtual speaker embedded in a 3D scenario offers the user the weather forecast for a selected location, currently chosen among Spanish cities or European capitals.

In this context, the developed RU-TTS system has to be compliant with the application architecture and specifications (see section 2). To meet these requirements, two data flow interfaces are appended to the TTS system (see section 3). Moreover, the classic concatenative corpus-based synthesis approach is tuned to achieve highly natural synthetic speech (see section 4). Finally, the suggested RU-TTS system is evaluated in terms of objective and subjective performance (see section 5).

2. TTS system description

Figure 2 shows the schematic diagram of the designed RU-TTS system. It is constituted by: *i)* an input interface consisting of an XML parser, which processes the input modified SSML tagged text (see section 3.1), *ii*) a TTS engine, *iii*) a Castilian Spanish speech corpus, and *iv*) an output interface, which provides the time code for scene synchronization. As a result, the system generates three output data: a WAV file containing the synthetic speech signal, a XTC (XML Time Code) file, which includes timing information for events synchronization (see section 3.2), and a SYN file, which contains the lip-synch information (see section 3.3).

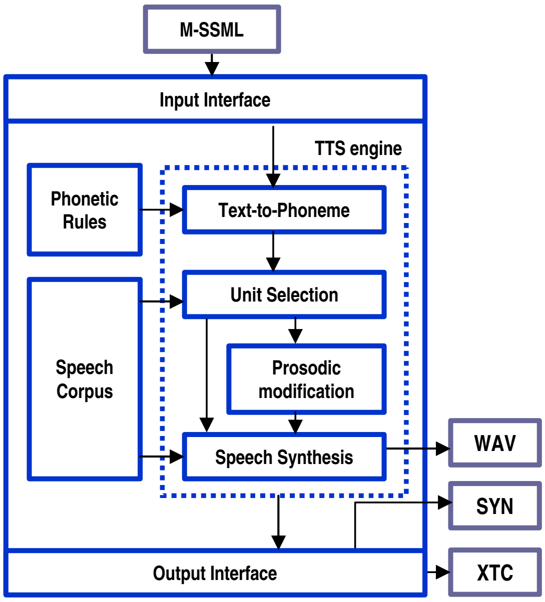

Figure 2: TTS diagram for the Virtual Weatherman application.

The speech corpus, with a total duration of 2.5h, is divided into three independent subcorpus: *welcome*, *forecast* and *farewell*. The unit selection process is only conducted in the corresponding corpus section, speeding up the unit search. In order to obtain highly natural synthetic speech, all the phrases used by the Automatic Text Generator to build the message (key components) (see figure 1) have been recorded with rich prosodic variability. Moreover, all the Castilian Spanish diphones (extended with some triphones) have also been recorded to synthesize *unseen* words (e.g. city names) or unit transitions.

The TTS conversion begins with a classic rule-based text-to-phoneme transcriber. Secondly, the unit selection module looks for the optimal set of units (basic search units are diphones plus some triphones) (see section 4.1). After unit search is conducted, the prosody of the selected units is retrieved as target prosody, adjusting the sharp transitions at concatenation points (see section 4.2). Finally, the speech signal is generated by means of a pitch synchronous process (see section 4.3).

3. Input and Output Interfaces

This section deals with the description of the input and output interfaces which allow the TTS system to interact with the Script Generator and 3D Scene Generator, respectively (see figures 1 and 2).

3.1. Modified SSML input

In order to satisfy the application requirements, a mark-up language inspired in SSML [6] has been designed to exchange data between the three main systems of figure 1. This modified SSML language is referred as M-SSML. The M-SSML document is composed of two parts: the header and the blocks. The header contains the definition of the global video variables specifying the parameters of the 3D Scene Generator: the main actor, the city name or the temperature signs, among others. The blocks define the contents of the different video components. Each block is composed of one or more <speak> elements. These elements include several SSML attributes and some extra tokens to extend the functionality of the SSML data description. One of them is the attribute *corpus*, which indicates the subcorpus where the selection has to be conducted. The rest of extra tokens are used to synchronize the video events, like camera shots or actor gestures. Following the SSML standard, *pitch*, *speed* and *volume* attributes can be adjusted within the <prosody> element. By means of this element, the prosodic information of the units can be adjusted in a relative fashion (e.g. <prosody pitch="high" speed="slow" volume="high">).

3.2. Event Synchronization Output

As the TTS system is embedded in a multimedia application, it is necessary to synchronize video events and synthetic speech. The synchronization is provided by a time code contained in a XTC file. This file is an enlarged version of the input M-SSML file, including timing information for audiovisual events synchronization. The attributes *begin* and *end* are added to the <speak> elements and to all those elements below its level in the XML hierarchy (e.g. embedded video events). Moreover, the XTC file incorporates the full path of each audio file and its corresponding lip-synch file as additional <speak> attributes. Notice that each <speak> element is associated to a couple of WAV and SYN files. Moreover, each input M-SSML file is related to one XTC file (see table 1).

Table 1: *Short example of a XTC file contents.*

```
<speak actor = “main_actor” voice_id=“SPANISH”
cache=“true” corpus=“welcome”
wav_file=“f:/out/ORAL_TV_DAY_BARCELONA01.wav”
lypsync_file=“f:/out/ORAL_TV_DAY_BARCELONA01.syn
begin=“0.000” end=“4.941”>
```

3.3. Lip synchronization output

The virtual character has to synchronize its lip movements with the speech signal. To avoid jamming the XTC file with excessive data, lip-synch timings are stored in a separate file, namely the SYN file. It incorporates SAMPA phonetic information in addition to the phoneme duration, which follows the same tag structure (*begin* / *end*) employed in the XTC file (see table 2).

Table 2: *Short example of a SYN file contents.*

```
< lip_sync>
<phoneme id=“o”  begin=”0.142” end=”0.254” />
<phoneme id=“l”  begin=”0.254” end=”0.294” />
<phoneme id=“a”  begin=”0.294” end=”0.526” />
</ lip_sync>
```

4. TTS engine

The RU-TTS system has been implemented by means of a phrase splicing approach [3]. Thus, its main goal is to find the longest segment of the recorded key components, retrieving and adjusting their prosody at synthesis time. The principal idea is to take advantage of the actual prosody of the units in order to preserve the recorded speaking style richness. This is only possible due to the full manual revision of unit boundaries and pitch marks, making corpus labelling completely reliable.

4.1. Unit Selection Module

The design and implementation of the unit selection module have been optimized to find the set of units with the minimum number of concatenation points (i.e. trying to retrieve full key components). The basic units (diphones or triphones with stress information) and the desired intonational type are the only data used for conducting a dynamic programming based full search (i.e. no clustering or pruning). As a consequence, a simple binary concatenation cost ($C^c = 0$, for consecutive units or $C^c = 1$, otherwise) has been employed. Moreover, a simple target cost is considered in terms of unit intonational agreement (see table 3).

Table 3: *Intonational target cost matrix. DEC, INT and EXC denote declarative, interrogative and exclamatory sentences.*

C^t	**Target**		
Selection	**DEC**	**INT**	**EXC**
DEC	0	0.05	0.05
INT	0.1	0	0.1
EXC	0.05	0.1	0

Experimentally, it is observed that often several sets of units achieve the same cost function value. This is due to: *i)* the recording of each key component using different speaking styles to enrich the variability of the corpus, and *ii)* the simplicity of the cost function design. Thus, in order to avoid picking always the same speech segments, the selected set of units are chosen randomly. Hence, successive identical queries are generated with different speaking styles.

4.2. Prosodic Adjustment

After selecting the longest speech segments from the corpus, their prosody (pitch, duration and energy) is retrieved as the target of the speech synthesis module (copy-prosody strategy). Unless indicated in the M-SSML file, the prosody of the units is maintained to preserve the recorded speaking style. However, the pitch curve needs to be adjusted to avoid pitch discontinuities at concatenation points or to face intonational variations. The pitch adjustment module (PAM) is described as follows.

4.2.1. Pitch interpolation at concatenation points

In order to avoid direct concatenation between boundary units with very different pitch values, a progressive smoothing of the pitch curve is carried out by means of a simple yet effective iterative procedure. This pitch adjustment is conducted on *n* units around the joint (*n* is experimentally adjusted to 3). The new pitch value is iteratively evolved (see figure 3) until the difference between the left an the right slopes of the pitch curve in that point is lower than an empirically determined threshold:

$$|m_l - m_r| < \Delta m_{max} \tag{1}$$

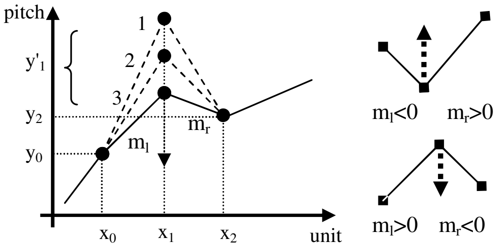

Figure 3: Iterative pitch curve peak smoothing at concatenation point. The adjustment depends on the point concavity.

The new ordinate value of the pitch curve is fixed depending on the concavity of each *n*-point, increasing (see eq. 2) or decreasing (see eq. 3) its value until the discontinuity is smoothed.

$$y_1^{'} < \frac{\Delta m_{max} + y_0 + y_2}{2} \tag{2}$$

$$y_1^{'} > \frac{-\Delta m_{max} + y_0 + y_2}{2} \tag{3}$$

4.2.2. Pitch modification due to intonational variation

When the target intonation of a speech segment differs from its own, the pitch curve also needs to be modified. Up to now, in the Virtual Weatherman application, this process must be conducted in two different situations: *i)* the intonational variation at the end of a sentence (e.g. conversion from declarative to interrogative and vice versa), and *ii)* the intonational variation due to the insertion or deletion of a pause (a comma, generally). From the analysis of the speech corpus it was concluded that the most significant pitch variations due to intonational modifications start at the last stressed vowel before the joint ($p(x)$ in eq. 4). This pitch value is used as a reference for computing the pitch value of the following units ($p(i)$ where $i = x+1, x+2, \ldots$). Equation 4 illustrates this process, which depends on the desired intonational variation. The pitch value is decremented or incremented with respect to $p(x)$, the value of the weighting factor $\beta \in [0, 1)$ and the distance from the last stressed vowel $(i - x)$.

$$p(i) = \begin{cases} p(x) \cdot (1+\beta)^{(i-x)} & \text{if } p(i) > p(x) \\ p(x) \cdot \Big(1 - \beta(i-x)\Big) & \text{if } p(i) < p(x) \end{cases} \tag{4}$$

After analyzing 800 sentences, splitted in two halves of interrogatives and declaratives, this relative pitch change factor has been adjusted to $\beta = 0.3$ for end-sentence intonational variation. Notice that, in the present application, the interrogatives only correspond to Yes/No questions. As for the pitch modification due to insertion/deletion of a pause, the factor β is 0.2, since intonational changes are less significant in this case. Figure 4 depicts the original pitch curve obtained from the actual pitch values of the selected units overlapped with the resulting curve after pitch adjustment.

4.3. Speech Synthesis module

Once the set of units and their corresponding target prosody is obtained, the synthetic speech signal is fully regenerated following a modified pitch-synchronous overlap and add algorithm [7]. The signal duration is adjusted interpolating the retrieved

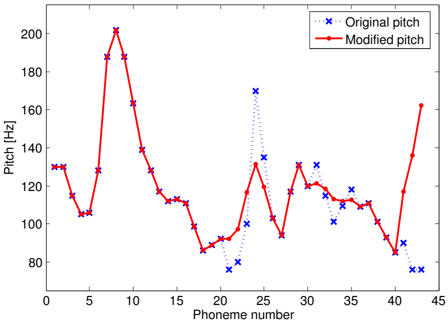

Figure 4: Adjustment and smoothing of the pitch curve at a point of concatenation (unit 23), for pause insertion (unit 31) and due to conversion from declarative to interrogative sentence (beginning at unit 40).

frames ($N2M$ algorithm [7]) and the energy discontinuities are globally minimized by means of a frame-to-frame smoothing process.

5. Experiments

CCRTV provided 900 weather reports for testing the RU-TTS system, containing the weather forecast for 175 different cities. Firstly, the system is analyzed in terms of objective performance, and secondly, after informally ratifying the highly natural synthetic speech achieved within the application, its dependence with respect to the prosodic adjustment is also analyzed.

5.1. Objective system performance

As the RU-TTS system described in this paper is a short-term real application, a performance test has been conducted on 900 synthesized reports. In average, each report lasted 39.1 ± 5.7 sec, and it was synthesized in 16.3 ± 2.75 sec. The test has been carried out over a Windows PC (PIV 3GHz - 1GB RAM) using the Visual .NET 2003 compiler. In terms of an objective speech quality measure, the number of units (diphones and triphones) per report was 43.02 ± 5.23 and the average number of concatenations (ANC) was only 0.55 ± 0.15 per sentence (each report contained 11.35 sentences in average).

In addition, if the unit selection process is conducted on the corresponding subcorpus (*welcome*, *forecast* or *farewell*), an average reduction of 40% of the execution time is achieved when compared to full corpus search, like in [4]. However, the same ANC (i.e. the same speech quality) is obtained in this case due to the totally application-oriented corpus design.

5.2. Subjective test

A preference test was developed in order to evaluate the PAM performance, a critical module for achieving highly natural speech. This test was composed of 10 pairs of audio files, each containing a sentence. One member of the pair was generated with the PAM on and the other with the PAM off. The pairs were randomly presented to 14 listeners, who were asked to choose between each pair according to their preference in terms of naturalness. The analysis of the results (see figure 5) yields a 76% preference for sentences generated with the PAM on. The results are very significant in terms of the analysis of variance (ANOVA) $(F(2,39) = 259.13, p < 0.000)$.

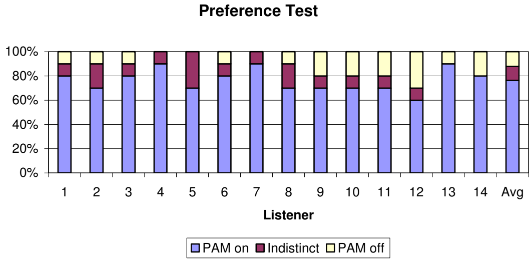

Figure 5: Preference test of 14 users judging 10 sentence pairs.

6. Conclusions

The restricted but unlimited domain TTS system described in this paper achieves highly natural speech and good performance in terms of computational requirements within the application. However, the speech quality decreases notably whenever the target sentences are not composed of the recorded set of key components. Future work is oriented towards improving this issue, conducting more exhaustive evaluation tests and implementing new strategies (e.g. clustering) to reduce the computational load of the synthesis process as full unit search is currently being conducted.

7. Acknowledgements

This work was partly supported by the Catalan Broadcasting Corporation (CCRTV). The authors are also grateful to the ITG of the Pompeu Fabra University for their valuable suggestions and discussions during the interfaces definition process.

8. References

[1] A. W. Black and K. Lenzo, “Limited Domain Synthesis,” in *ICSLP*, Beijing, China, 2000.

[2] A. Schweitzer, N. Braunschweiler, T. Klankert, B. Säuberlich, and B. Möbius, “Restricted unlimited domain synthesis,” in *EuroSpeech*, Geneve, 2003, pp. 1321–1324.

[3] R. E. Donovan, A. Ittycheriah, M. Franz, B. Ramabhadran, E. Eide, M. Viswanathan, R. Bakis, W. Hamza, M. Picheny, P. Gleason, T. Rutherfoord, P. Cox, D. Green, E. Janke, S. Revelin, C. Waast, B. Zeller, C. Guenther, and J. Kunzmann, “Current Status of the IBM Trainable Speech Synthesis System,” in *The 4th ISCA Tutorial and Research Workshop on Speech Sintesis*, Perthshire, Scotland, 2001.

[4] F. Alías, I. Iriondo, and P. Barnola, “Multi-domain text classification for unit selection Text-to-Speech Synthesis,” in *The 15th International Congress of Phonetic Sciences (ICPhS)*, Barcelona, 2003, pp. 2341–2344.

[5] X. Sevillano, F. Alías, and J. Socoró, “ICA-Based Hierarchical Text Classification for Multi-domain Text-to-Speech Synthesis,” in *ICASSP*, vol. 5, Montreal, 2004, pp. 697–700.

[6] W3C, “Speech synthesis markup language, version 1.0.” http://www.w3.org/TR/speech-synthesis.

[7] I. Iriondo, F. Alías, J. Sanchis, and J. Melenchón, “A Hybrid Method Oriented to Concatenative Text-to-Speech Synthesis,” in *EuroSpeech*, Geneve, 2003, pp. 2953–2958.