

Voice Synthesis System Based on Recursive Functions Designed by Graphs

Rodolfo Romero Herrera¹, José Elias Romero Martínez²

¹ Instituto Politécnico Nacional, Escuela Superior de Cómputo,
Departamento de Ciencias e Ingeniería de Computación,
Ciudad de México, Mexico

² Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica
Ciudad de México, Mexico
rromero@ipn.mx, saile14398@gmail.com

Abstract. A recursive function based on the use of graphs is presented. The recursive equation is based on mathematical morphology; A written text divided into syllables is related to phonemes of spoken Spanish in Mexico. The algorithm is implemented in a PIC and an integrated circuit that stores the sounds in memory locations. Each space in memory becomes a code that is related to a phoneme. Thus, the recursive function receives as input a vector that is the structure of a word; and outputs the corresponding memory locations. Obtaining intelligibility of more than 84%. The development of the recursive function in graphs is proposed, which allows an easy description of the sequence because it is handled as a black box with input and outputs; and the equation that governs the system can even be a transfer function.

Keywords: Synthesis of voice, graph, recursion, mathematical linguistics.

1 Introduction

Recursive functions are observed and treated in different ways. In the case of artificial intelligence, the recognition of patterns is a case treated by many [1]; and of course, hardware designed with the appearance of FPGAs have a special meaning in the design; although it is complicated to implement, especially when it is not accessible to most programmers because of the specialization required [2]. A useful option is to visualize the systems as black boxes, considering their inputs and outputs, as well as their transfer function; however, some form of describing them is required. What generates matrices that allow identifying states and exits of the system that consequently takes us to recursive functions that can even depend on probabilities [3]. This is where the description of graph-based systems simplifies the design and application of even nature-based systems such as Genetic Algorithms [4]; or problems of analysis of public transport networks or interconnected social networks, etc. [5 to 7].

This establishes the design of systems based on graphs that can be directed or not, and that allows establishing a correspondence between even the combinational design [8], sequential or systems based on probabilities and the algebra of graph theory [9] [10]. In this way, in the present work, a voice reproduction system based on phonemes

is designed using the graph technique for the recursive function; and mathematical linguistics to describe the grammar of function [11].

2 Voice Processing

A digital signal processing project is presented that justifies the results of the methodology proposed in this work. The project consists of a phonological system of the Spanish language spoken in Mexico.

For the development of the project, the naturalness and intelligibility of the resulting voice and mainly the complexity of the processing was taken into account, because the use of a graph-based technology is tested [13]. Once proven its feasibility, it is feasible not only to use phonemes, it is also possible to use di-phonemes or the combination of both.

For the recording of the phoneme, the WAV format was used with a sampling frequency of 11025Hz, mono-stereo, at 16 bits [11]. The following steps were carried out with Matlab:

- Filtering the phoneme,
- Elimination of entrance noise,
- Delimitation of the phoneme,
- Phoneme selection.

The recording of all the simple combinations with a vowel and complex with existing vowel was performed, as well as separate vowels, and consonants located in coda position susceptible to being isolated and recorded, such as “*b, c, d, f, g, l, m, n, r*” and “*s*”. The syllables that have the consonants *p* and *t* in coda position are reviewed and recorded together with the attack (if it exists) and the nucleus [14].

Table 1. Phonemes in coda position.

Simple attack	Complex attack with l	Complex attack with r	Core
<i>b [a - u]</i>	<i>b [a - u]</i>	<i>br [a - u]</i>	<i>A</i>
<i>c [a - u]</i>	Does not apply	Does not apply	<i>E</i>
<i>d [a - u]</i>	Does not apply	<i>dr [a - u]</i>	<i>I</i>
<i>f [a - u]</i>	<i>fl [a - u]</i>	<i>fr [a - u]</i>	<i>O</i>
<i>g [a - u]</i>	<i>gl [a - u]</i>	<i>gr [a - u]</i>	<i>U</i>
<i>j [a - u]</i>	Does not apply	Does not apply	
<i>k [a - u]</i>	<i>kl [a - u]</i>	<i>kr [a - u]</i>	
<i>l [a - u]</i>	Does not apply	Does not apply	
<i>ll [a - u]</i>	Does not apply	Does not apply	
<i>m [a - u]</i>	Does not apply	Does not apply	
<i>n [a - u]</i>	Does not apply	Does not apply	
<i>ñ [a - u]</i>	Does not apply	Does not apply	
<i>P [a - u]</i>	<i>pl [a - u]</i>	<i>pr [a - u]</i>	
<i>r [a - u]</i>	Does not apply	Does not apply	
<i>rr [a - u]</i>	Does not apply	Does not apply	
<i>s [a - u]</i>	Does not apply	Does not apply	
<i>t [a - u]</i>	<i>tl [a - u]</i>	<i>tr [a - u]</i>	

Table 1 shows some phonemes in coda position, given the way in which they are generated, specifically p and t; it was not possible to isolate them in a way; so it is necessary to record the basic combinations of Spanish that involve these two elements in coda position [14].

3 Mathematical Morphology for the Voice Concatenation Function

We used a microcontroller that implements the equation of nature [14]. See equation (1) where S is a black box, e is an element, e * means that you can have as many elements as you want and S * indicates that the box can be called as many times as you want. If S is a speech synthesizer system to produce any word of Spanish, it can call itself as many times as the limit that is marked by a phrase.

$$S \rightarrow e^*S^* \tag{1}$$

Mathematical linguistics was used in the speech synthesizer. If it works by mathematical linguistics, it carries out the following actions:

El orador dice hola,
El orador dice Adios.

These sentences work with a structure of sentences seeking to find a general structure or syntactic rule from particular structures or canonical prayers. An equivalent sentence is:

El orador dice hola y dice Adios

Now fragments of the sentence are replaced by labels according to Table 2; where $o_1r_1o_2$ and $o_1r_1o_3$ in equation 2 are known as canonical sentences [16].

Table 2. Replacement of fragment by label.

Fragment	Label
<i>El orador</i>	o_1
<i>Dice</i>	r_1
<i>Hola</i>	o_2
<i>Y</i>	+
<i>Dice</i>	r_1
<i>Adios</i>	o_3

$$o_1 r_1 o_2 + o_1 r_1 o_3 \tag{2}$$

The algebraic factorization consists of finding the common factors in an expression. Applying the factorization to the canonical sentences of (2) we have to:

$$o_1 r_1 (o_2 + o_3) \tag{3}$$

In (3) the simplification of words is observed, so that memory is saved and it is possible to implement it as a recursive function. When a set of sentences contains an element that is repeated indefinitely, it is convenient to replace the entire set of sentences with a recursive grammar, for example:

$$S \rightarrow ab; aab; aaab; aaa\dots ab; \tag{4}$$

An associated recursive grammar is:

$$S \rightarrow aS | b, \tag{5}$$

where "|" represents the concatenation. Substituting S for aS or for b can have chains like the following:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaab. \tag{6}$$

This can be understood as a system or reusable component that is called likewise. Recursion is introduced when it is detected that a string "a" repeats monotonously. For example, in prayer:

$$Abcabcbcabcd, \tag{7}$$

where the string *abc* presents a monotonous behavior that ends in d.

Equation 4 and 6 can be represented with directed graphs. See figure 1.

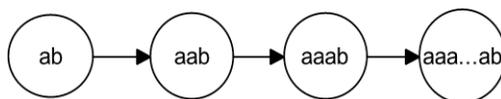


Fig. 1. Graph equation 6.

But as shown in (5) it is feasible to modify the graph in such a way that it can be implemented as a directed graph.

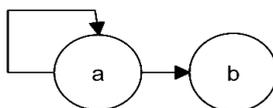


Fig. 2. Graph equation 5.

4 Voice Processing

Once defined the elements and the mathematical foundation of the system, you can establish the graphs that allow the design of the system, which will depend on the degree of abstraction that the designer determines convenient since this can be very specific or very generic. Figure 3 presents the graph for the synthesizer where each node represents a function.

- a) The solution of the irregularities of the Spanish language applied to all the phrases and substitution of graphemes by the symbols used in the representation of phonemes in the speech synthesizer.
- b) Separation in syllables of the sentence according to syllabication rules.
- c) Extraction of phonemes from the phrase.
- d) Concatenation of the samples of the audio files of the phonemes to be used.
- e) Reproduction of the samples of the resulting concatenation.

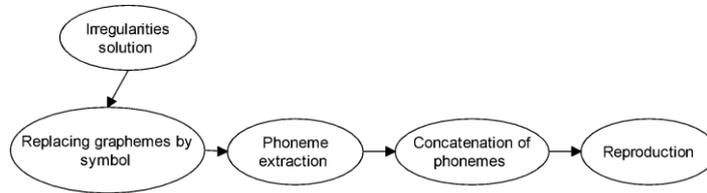


Fig. 3. Graph for components of the synthesizer.

The Conca function is responsible for concatenating the different files stored in a single object, which is responsible for reproducing the input message. Once the arrangement with the names of the audio files has been obtained, Conca opens each of these files and its samples are read.

Samples are stored in an object that contains the samples of all the files (each file contains a phoneme). When you finish writing this object, you start writing the samples of the next file, until you finish with the last one. Figure 4 presents the graph corresponding to Conca.

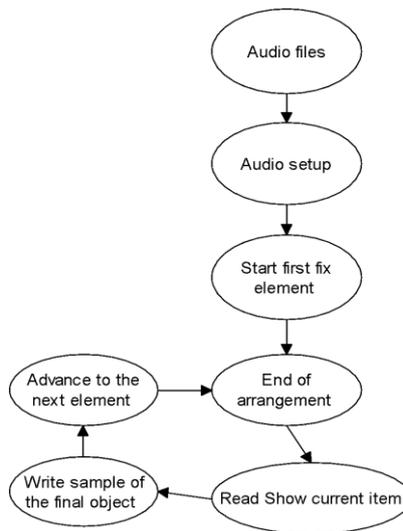


Fig. 4. Degree of Conca.

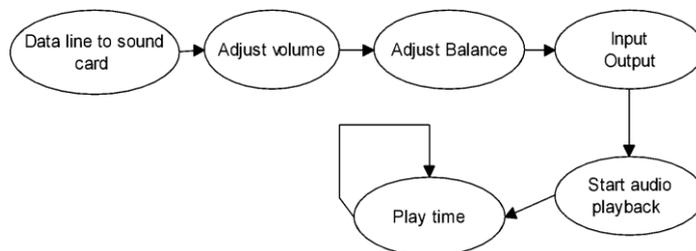


Fig. 5. Degree of the Play function.

Figure 5 shows the graph of the Play function. The created phrase is taken and reproduced. It is also possible to adjust the volume and balance of playback by two parameters in this module.

5 Results

The voice obtained at the output of the synthesizer has an intelligible average quality. Figure 6 (a) shows the shape of the signal from the word lomo recorded in a normal way. Figure 6 (b) presents the concatenation of phonemes. The decrease and increase of the signal between the combination of the phoneme “lo” and “mo” are clearly observed. This reduction creates a lack of fluidity in the resulting voice.

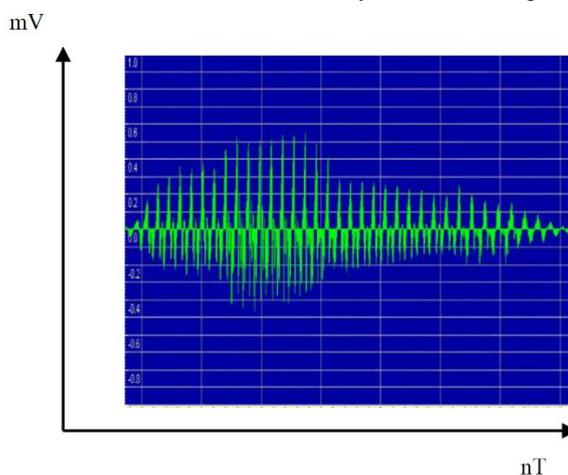


Fig. 6. (a) Word "lomo" recorded.

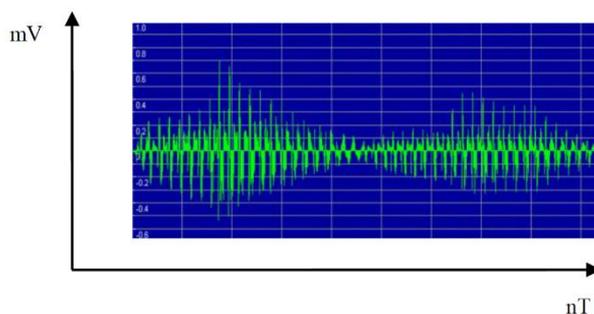


Fig. 6. (b) Word "lomo" synthesized.

The intelligibility test of the speech synthesizer was applied with a group of people to listen to generated words and phrases, without previously knowing what they were [17]. The results of these tests are shown in Tables 3 and 4, respectively. From these tests, it is deduced that only 15.66 did not understand the phrase; 50 male and 50 female individuals were taken to perform the tests.

Table 3. Intelligible phrases.

Words understood correctly:	253
Words that are incorrectly understood:	47
Intelligibility percentage:	84.34 %

For the second test, phrases were chosen that did not have something in common, but that did have coherence by themselves. Only 0.5% did not understand the phrase.

Table 4. Intelligibility in sentences without something in common.

Phrases correctly understood:	224
Phrases included incorrectly:	1
Intelligibility percentage:	99.55%

Figure 7 shows the physical system of voice reproduction, where the components are observed. The functions seen as black boxes were implemented in hardware through an AX PIC. The circuit only uses the PIC together with an audio storage memory where the different phonemes to be reproduced are placed [18].

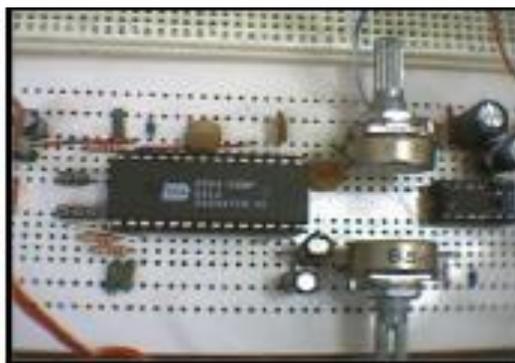


Fig. 7. Voice synthesizer system.

6 Conclusions

A proposal for modeling in graphs was presented. The proposed diagrams are based on the object-oriented design, which will allow us to obtain efficient graphics modeling. In the same way, we present how mathematical linguistics allows us to represent systems through the equation of nature; checking how a reusable system can be called or multiplied to develop complex tasks, as a recursive function does, proposing the construction of hardware systems supported by concatenation operations. The system applied to hardware development allows us to simplify resources and take advantage of the potential of devices such as microprocessors, etc. In such a way that a system in hardware can call itself so many times until it reaches the limit. Seen in this way, generic systems represented by a graph can be produced. Where an object can be a component or a reusable system and this, in turn, a recursive function in hardware.

Although no tests with di-phonemes were performed, the intelligibility level exceeded 99%. In future work, in order to avoid the robotic voice, it is considered to make an integration of phonemes and phonemes; once it has been demonstrated that the design using graphs gives advantages; Although no tests with di-phonemes were performed, the intelligibility level exceeded 99%. In future work, in order to avoid the robotic voice, it is considered to make an integration of phonemes and phonemes; once it has been demonstrated that the design using graphs gives advantages; since, although here it was not considered conditional probabilities or some other factors it is evident that when applying it results in greater benefits.

Acknowledgment. Acknowledgments to the IPN for the support received for the development of this Project.

References

1. Puerto Cuadros, E.G., Aguilar Castro, J.L.: Un algoritmo recursivo de reconocimiento de patrones. *Revista Técnica* 40(2), p. 95 (2017)
2. Pimentel, B.F.: *Synthesis of FPGA-Based Accelerators Implementing Recursive Algorithms*. ProQuest Dissertations Publishing (2009)
3. Medel Juárez, J.J., Zagaceta Álvarez, M.T.: Estimación-identificación como filtro digital integrado: descripción e implementación recursiva. *Revista Mexicana de Física* 56(1), 1–8 (2010)
4. Quintero, J.A.G., Toledo, E.S., Diaz, M.A.: Reconfiguración de sistemas de distribución mediante algoritmos genéticos basados en la teoría de Grafos. *Ingeniería Energética* 37(2), p. 115 (2016)
5. Cardozo, O.D., Góme, E.L., Parras, M.A.: Teoría de grafos y sistemas de información geográfica aplicados al transporte público de pasajeros en resistencia (Argentina). *Transporte y Territorio*, No. 1, pp. 89–111 (2009)
6. Mena Díaz, N.: *Redes sociales y Gestión de la Información: un enfoque desde la teoría de grafos* (2012)
7. Escalona, Y.E. *et al*: Sistema para la solución de problemas de cómputo basado en el Método de los Grafos Dicromáticos. *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología* 1(1), 37–46 (2014)
8. Avalos Gaytán, V., Rivera Ramírez, M., Schaeffer, E.: Agrupamiento local en grafos dirigidos (2009)
9. Haemers, W.H., Kharaghani, H., Meulenberg, M.A.: Divisible design graphs. *Journal of Combinatorial Theory, Series A*, 118(3), 978–992 (2011)
10. Booch, G.: *The Unified Modeling Language User Guide*. Addison-Wesley, USA (2000)
11. Matworks: Matlab: <https://la.mathworks.com/products/matlab.html>, USA (2019)
12. Ortega, A. et al: Graph Signal Processing: Overview, Challenges, and Applications. *Proceedings of the IEEE* 106(5), 808–828 (2018)
13. Voiers, W.D.: Diagnostic Evaluation of Speech Intelligibility. In: Hawley, M.E. (ed.): *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson, and Ross. Stroudsburg, PA. (1977)
14. Valero, L.A.: Curso audiovisual interactivo de codificación de voz. Departamento de electrónica y Tecnología de computadoras, Universidad de Granada, <http://ceres.ugr.es/~alumnos/luis/codif.htm>, España (2005)
15. Galindo Soria, F.: *Una ecuación de la Naturaleza*. Escuela Superior Cómputo, Instituto Politécnico Nacional (1998)

16. Marcus, S., Nicolau, E., Stati, S.: *Introducción en la lingüística matemática*. Editorial Teide/Barcelona (1966)
17. Voiers, W.D.: *Diagnostic Evaluation of Speech Intelligibility*. In: Hawley, M.E. (ed.): *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson, and Ross. Stroudsburg, PA. (1977)
18. Piacaxe, <http://www.picaxe.com/> (2019)