The development of algorithms for alleviating the problem of discontinuity in speech synthesis from text written in Albanian

Adnan Maxhuni[1], Agni Dika[1], Avni Rexhepi[1] and Dren Imeraj[1]

[1]Faculty of Electrical and Computer Engineering, University of Prishtina
Prishtina, 10000, Kosovo

Abstract

Efforts to generate speech from written text are different, ranging from mechanical ones of centuries ago, to software solutions of recent decades, known as TTS (Text-to-Speech). The ultimate goal of a speech synthesizer from written text is 'ability' to read any text [2]. Reading should be understandable and natural. A speech synthesizer contains "the part" for the linguistic analysis (NLP - Natural Language Processing) and the digital signal processing (DSP - Digital Signal Processing)[4].

Albanian language as a separate language in Indo-European family of languages, with its own specifics in writing and reading differs from other languages and models for these languages cannot be used for conversion of Albanian texts. We acquired the Concatenation Synthesis model of building the conversion system of text to speech. This technique is based on preliminary recording of the text, from which acoustic segments are cut and stored in a database. Then, we create application which concatenates these acoustic segments, to "create" words and sentences [4].

During concatenation of sequences, due to differences in their amplitudes emerge problem of disconnection. In this paper we present some algorithms for minimizing the effect of discontinuity along concatenation of acoustic segments of the synthesized speech [9].

Keywords: *Sequence, Acoustic Segments, Algorithm, Amplitude.*

1. Introduction

Simple assembly of two acoustic sequences in order to make words causes disconnection/discontinuity that make the generated speech not very clear, especially unnatural. Voice generated in this way is known as "metallic" and intermitted. For these reasons we have developed a set of algorithms in order to minimize these problems. There is no single algorithm which would solve all these problems. Here we address the problem of amplitude differences, caused during reunion of sequences stored in database [10].

The first case is when there is a break after the end of the sequence (sequence at the end of the word). In these cases, the sequence should be reduced gradually. Change in amplitude should be linear and there should be a certain period of transition [7].

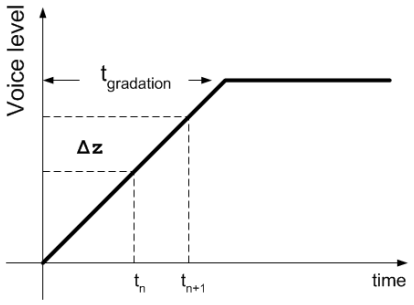
The other case is the connection of two sequences (in the middle of the word). In most cases, the amplitudes of the sequences do not match. This problem is mitigated in two ways: through the amplitude equalization or through gradual amplitude reduction to a certain level [6]..

2. The effect of discontinuity during union of acoustic segments

Algorithm for mitigating the effect of discontinuity during union with gradually changing the amplitude of the phoneme, can be realized with multiplying the sequence samples with a constant. In the case of increasing the amplitude from zero to its maximum value, or the reduction from its full value to zero, this "constant value" takes values from 0 to 1, respectively, from 1 to 0. Since the period of change and the starting/ending point are taken as parameters, the expression for the amplitude change should be depending on them [6].

2.1 Algorithm for gradual increase and decrease of the amplitude

Because it is required that this change has to be linear (Fig.1), we must first set a constant Δz (linearity constant) which shows the amplitude change in the unit of time (between samples) [1].



$$\Delta z = \frac{1}{t_{gradation}}$$

Fig.1 Gradual amplitude increase

Pseudo code for amplitude rise/increase from zero to its full value (max. value):

```
dZ = 1 / t_gradation
gradation = 0
t = t_start
WHILE (t < t_start + t_gradation)
     samples[t]=samples[t] * gradation
     gradation = gradation + dZ
     t = t + 1
REPEAT
```

After execution of this algorithm, the values of the `samples[]` vector will be modulated so as to gradually reach maximum amplitude value. Values `t_start` and `t_gradation` are input values of the algorithm. Similarly is done gradual amplitude decrease.

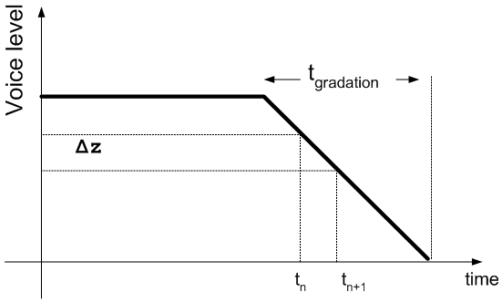


Fig.2 Gradual amplitude decrease

Pseudo code of the algorithm for amplitude decrease to the zero value (Fig. 2), is as follows:

```
dZ = 1 / t_gradation
gradation = 1
WHILE (t < t_start + t_gradation)
      samples[t]=samples[t]*gradation
      gradation = gradation - dZ
      t = t + 1
REPEAT
```

Abovementioned algorithms can be applied in two ways:
The first way is to apply the algorithm after sequence construction. In this case, the sequence is constructed by copying samples from the database of sequences and then algorithm for amplitude changing is applied. This way is convenient when change of the amplitude must be applied in the middle of the sequence copied from the base of sequences [9].

The second way is the application of change during sequence copying. This method reduces the execution time of the algorithm, as for to apply change, it uses the sequence copying cycle.
Depending on the use case, one or the other method may prove more effective [9].

To test the algorithm we take the sequence for letter 'e' in the word 'teksti'. In this word, after the letter 'e' should be a pause and then continues the rest. Because the sequences of the letter 'e' does not end gradually, then the algorithm for gradual reduction of amplitude must be applied [10].

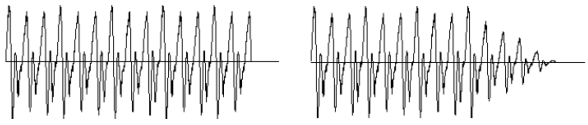


Fig.3 Sequence before and after the application of the algorithm for amplitude decrease/reduction

From Fig. 3 it can be seen that all points of the sequence (not only the amplitudes) are gradually reduced. Indeed gradual extinction of the sequence is the goal of this algorithm. Also, the starting point of the amplitude change can be noticed. The starting point of changing the amplitude is given as a parameter. Here, the amplitude decrease started in 2000 samples before the end of the sequence [3].

In algorithms for copying the sequence and overlapping, is used vector `samples[]`. This vector contains samples for the sequence. Since the algorithms defined in this section apply PCM, each sample shows the amplitude of sequence in a particular moment. In the PCM technique, samples are equally distant from each other, because there is a frequency of sampling [5].

Each sample is a numeric value that represents the current value of the amplitude. If in a sequence, is used PCM technique with 16 bits per sample, the amplitude values can be in the range [-32 768; 32 767]. By changing values (magnitudes) of the samples, is done the change in the form of sequence.

2.2 Algorithms for equalization of amplitudes

During the union (link) of the sequences repeatedly (e.g. transition from 'e' to 'ek'), usually appears the problem of differences in amplitudes.

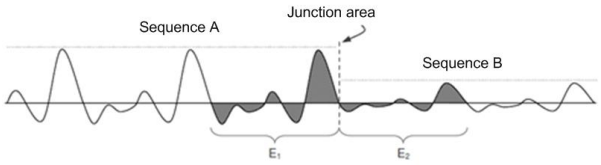


Fig.4 Problem of difference in amplitudes

Since recordings are made at different times, it is natural that the amplitudes differ from sequence to sequence, although their wave shape is almost identical (Fig. 4). Rapid changes in amplitude create harmful effects on comprehensibility and naturality of speech.

To eliminate this difference of amplitudes, we tried to equalize the energy of periods of the sequence B (E2) with energy of periods of the sequence A (E1).

As the wave forms of the two sequences are very similar, but differ in amplitude, then the sequence A can be describes with a.f(x), while the sequence B with function bf(x). In this way, equation for energy of a discrete signal for sequences A and B will be [1]:

$$E_1 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, af[n]\,|^2$$

$$E_2 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, bf[n]\,|^2 \qquad \text{.....(1)}$$

Equations in (1) can be further extended:

$$E_1 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, af[n]\,|^2 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, a\,|^2 |\, f[n]\,|^2 =$$

$$= \frac{1}{N}\sum_{n=n_0}^{n_0+N} a^2 \,|\, f[n]\,|^2 = a^2 \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, f[n]\,|^2$$

$$E_2 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, bf[n]\,|^2 = \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, b\,|^2 |\, f[n]\,|^2 =$$

$$= \frac{1}{N}\sum_{n=n_0}^{n_0+N} b^2 \,|\, f[n]\,|^2 = b^2 \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, f[n]\,|^2 \qquad \text{.....(2)}$$

If from the last equation (2), we find the ratio of the energies of sequences A and B, we'll have:

$$\frac{E_1}{E_2} = \frac{a^2 \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, f[n]\,|^2}{b^2 \frac{1}{N}\sum_{n=n_0}^{n_0+N} |\, f[n]\,|^2} = \frac{a^2}{b^2}$$

$$\frac{a}{b} = \sqrt{\frac{E_1}{E_2}} \qquad \text{.....(3)}$$

If we multiply each sample of the sequence B with $\sqrt{E_1 / E_2}$, than sequence B ($b \cdot f(x)$) will become:

$\sqrt{\frac{E_1}{E_2}}bf(x)$, that from (3) can be written as:

$$\sqrt{\frac{E_1}{E_2}}bf(x) = \frac{a}{b}bf(x) = af(x) \qquad \text{.....(4)}$$

So, if each sample of sequence B is multiplied with $\sqrt{E_1 / E_2}$, sequence B will have amplitudes equal with sequence A.

To implement the amplitude equalization algorithm it is necessary to know some parameters related sequences. The most important parameter is the length of the period, which must be precise in order to make accurate normalization [5].

According to the algorithm, initially must be calculated energies of the two periods to be joined. Then, each sequence of the second sample should be multiplied by the square root of the quotient of energies. Input parameters of the algorithm for normalization are:

Point_Junction - the point where the two sequences join together
Length_Period1 - the length of the last period of the first sequence
Length_Period2 - the length of the first period of the second sequence
Length_Sequence2 - total length of the second sequence
Below is presented the algorithm that does the equalization of amplitudes after join of the sequences:

```
E1 = 0
t = Point_Junction
WHILE t > PointStart - Length_Period1
      E1 = E1 + samples[t]^2
      t = t - 1
REPEAT
      E1 = E1/Length_Period1
      E2 = 0
      t = Point_Junction
```

```
WHILE t < PointStart + Length_Period2
      E2 = E2 + samples[t]^2
      t = t + 1
REPEAT
      E2 = E2/Length_Period2
      FactorNormalization = sqrt(E1/E2)
      t = Point_Junction
WHILE t < GjatesiaSekuenca2
      samples[t]=samples[t]*FactorNorma
      lization
      t = t + 1
REPEAT
```

After applying the above algorithm, the amplitudes of the second sequence will be equal to those of the first sequence. In this way are eliminated the effects of the immediate backdrop of amplitude at the point of junction. Applying the above algorithm has the effect of reducing the amplitude of the first sequence from the original level to the level of the amplitude of the second sequence (Fig. 5).

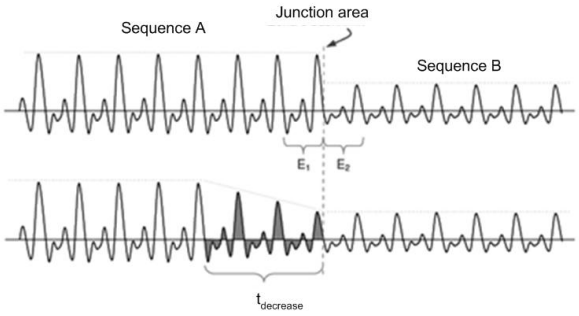


Fig.5 Application of the algorithm for decreasing the amplitude at the junction area

2.3 Algorithm for gradual decrease of the amplitude

Equalization of amplitudes through algorithm for gradual decrease of amplitude, in some cases it is not possible, because in the sequential sequence after the join, cannot be changed all values of the amplitudes. In such cases is applied the algorithm which gradually decreases the amplitudes of the first sequence, so that at the point of junction, the amplitude is approximately as the amplitude of the second sequence. The decrease must be realized during the time tdecrease.

If we analyze the requirement of the latest algorithm, we'll see that there are similarities with the algorithm for the gradual increase of amplitude. The difference between these two algorithms is the level of the last amplitude. While in the algorithm for gradual increase of the amplitude the last level is zero, in the algorithm for equalization of amplitudes after joining of sequences, as can be seen in the last figure, should be as the level of the first amplitude of the following sequence.

Requirement is that the amplitude drops linearly from A0 (amplitude of sequence A before application of the algorithm) to A1 (amplitude at the beginning of the sequence B) within the time tdecrease. In this case Δz will have the value:

$$\Delta z = \frac{1}{t_{decrease}} \quad \text{.....(5)}$$

If variable ndryshimi with initial value 1 decreases for Δz from pbashkimi – trenja till pbashkimi, at the point pbashkimi will have zero value. Using variable `gradation` we can write:

$$r = \frac{\sqrt{E_2} \cdot gradation + \sqrt{E_1} \cdot (1 - gradation)}{\sqrt{E_2}} \quad \text{.....(6)}$$

If each sample of the first sequence from $p_{junction}$ – $t_{decrease}$ till $p_{junction}$ is multiplied with r, there will be gradual decrease during the interval.

Algorithm for equalization of amplitudes after joining of the sequences uses division of the square roots of energies, which means that it is necessary to calculate the energies of the periods. After calculating the energies, is calculated ratio r given by equation (6) and each sample of the first sequence from $p_{junction}$ – $t_{decrease}$ to $p_{junction}$ is multiplied by this value (r), (pseudocode below).

```
E1 = 0
t = Point_Junction
WHILE t > Point_Start - Length_Period1
            E1 = E1 + samples[t]^2
            t = t - 1
REPEAT
E1 = E1/Length_Period1
E2 = 0
t = Point_Junction
WHILE t < Point_Start - Length_Period2
      E2 = E2 + samples[t]^2
      t = t + 1
REPEAT
E2 = E2/Length_Period2
gradation = 1
dZ = 1/t_decrease
r = (sqrt(E2)*gradation + sqrt(E1)*(1-
gradation))/sqrt(E2)
t  = Point_Junction - t_decrease
WHILE t < Point_Start
      samples[t] = samples[t] * r
            r = r + dZ
            t = t + 1
REPEAT
```

3. Conclusions

Concatenation Synthesis is the most used technique for speech synthesis from written text. Acoustic segments are created by cutting dyphones from a reading text. These acoustic segments stored in a database. Through software applications, with the union (junction) of these acoustic segments is synthesis the speech. The union of acoustic sequences to form words in speech synthesis from written text occur problems, which causes no meaningful and non-natural speech. This is caused by the change in the amplitude level of the sequences. The algorithms presented in this paper do minimize these effects by improving the quality of the generated speech. These algorithms have to do with icreasing or decreasing the amplitude of the sequence at the beginning or end of the word. We also present an algorithm for equalization of the two sequences amplitude in the union area.

References

[1] Abramowitz, Milton; Stegun, Irene A., "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables", Dover, 1965.

[2] Dutoit, Thierry, "An introduction to text-to-speech synthesis", Kluwer Academic Publishers, 2001.

[3] Hayes, M. Horace, "Digital Signal Processing",McGraw Hill, 1999.

[4] Jurafsky, Daniel & Martin, James H., "Speech and Language Processing",Prentice Hall, 2009.

[5] Kaiser, J. F., "System Analysis by Digital Computer", Wiley, 1966.

[6] Lemmetty, Sami., "Review of Speech Synthesis Technology", 1999.

[7] Agni Dika, Adnan Maxhuni, Avni Rexhepi, "The principles of designing of algorithm for speech synthesis from texts written in Albanian language", IJCSI, May 2012.

[8] Avni Rexhepi, Agni Dika, Adnan Maxhuni, "Conversion of numbers to text and to speech in Albanian", WSEAS - World Scientific and Engineering Academy and Society, Greece, 2012.

[9] MBROLA. Multilingual Speech Synthesizer http://tcts.fpms.ac.be/ synthesis/mbrola.html, 2008.

[10] Agni Dika, Mentor Hamiti, "Options for Converting Albanian Written Texts into Speech", BCI, 2009.

Adnan Maxhuni Master of Computer Science (2005) at the University of Prishtina; Teaching and Research Assistant at the University of Prishtina.

Agni Dika PhD degree in Computer Sciences (1989) at the University of Zagreb, Croatia; Full Professor at the University of Prishtina, Faculty of Electrical and Computer Engineering.

Avni Rexhepi PhD degree in Computer Science (2013) at the University of Prishtina; Teaching and Research Assistant at the University of Prishtina.

Dren Imeraj BSc in Computer Science (2012) at the University of Prishtina.