University of Sarajevo
Faculty of Electrical Engineering Sarajevo

Static Linking of Phonemes to Polygonal 3D Model's Facial Expressions


Prof. Zikrija Avdagic


Dr Selma Rizvic


Dr Samim Konjicija


Mr Adnan Nuhic

Project timetable

- 2003-2007/Jan-Jul,Sarajevo

Theoretical research :

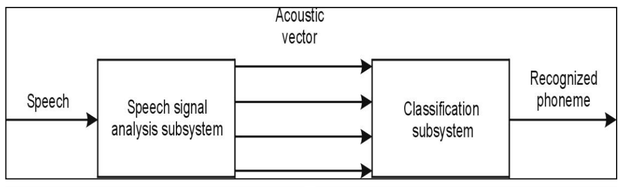
- Field: Artificial Intelligence
- Field: Computer Graphics
- Field: Signal Processing

Practical implementation :

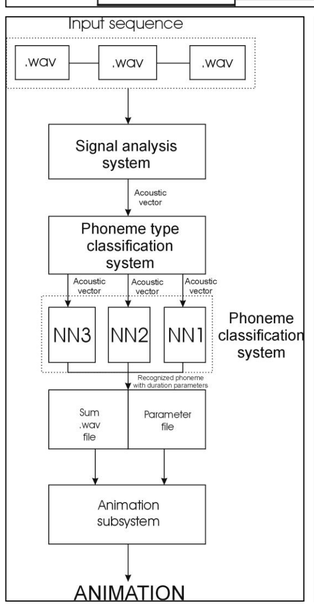
- MATLAB-Neural Network Toolbox
- Software Maya

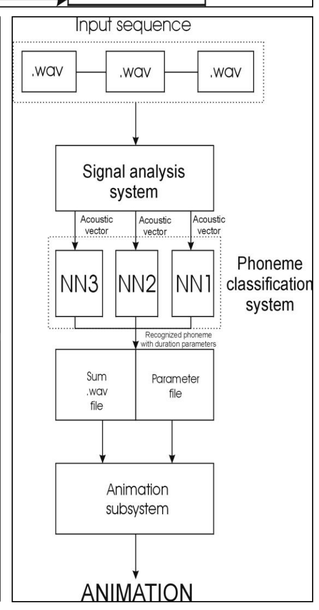
Project description

Facial animation consists of two parts – speech animation and animation of emotions. Using the prerecorded soundtrack of the speech, we can create speech animation and fine tune it by adding emotional expressions.
Linguists have defined that the human speech consists of a particular number of phonemes e.g. the smallest contrastive units in the sound system of a language. The number and kinds of phonemes are different for every language. In our research we experimented with the Bosnian language phonemes. Viseme is a generic facial image that can be used to describe a particular phoneme. We created 13 visemes for 30 Bosnian language phonemes.
In last two decades facial animation has been intensively explored and various approaches and techniques were introduced in order to achieve the most accurate and natural look of the animated character. Three basic approaches are concatenative, parameterized and muscle based approach.
Our research is based on parameterized approach introduced by Parke, where the 3D polygonal mesh of the model is animated using the data from the parameter file, such as phonemes and their durations.
The project is organised as follows: in sections 2-5 we introduce the training and phoneme recognition using LVQ neural networks and generating the parameter file. Section 6 describes mapping of the phonemes from the parameter file to 3D model's facial expressions using the MaxScript control script. The final section is a general conclusion about our present work and some details of future work for the automated speech animation of Bosnian language phonemes.

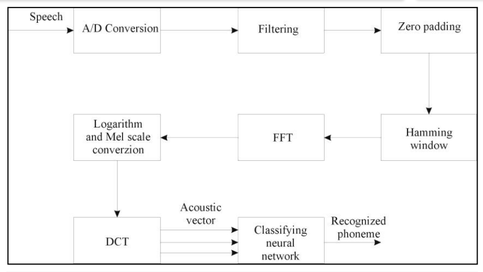


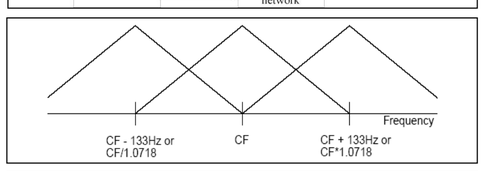
Speech is a signal mathematically represented as a function of time. It can be viewed as continuous time signal or discrete time signal.
In most of the cases the goal of signal processing is obtaining certain characteristics of output signal. In speech signal analysis the goal is interpretation of signal and extracting characteristics and information from input signal. Typically digital processing is used here (such as filtering, parameter estimation etc.), followed by feature recognition. The outputs of such a system are symbols (such as phonemes or whole words).

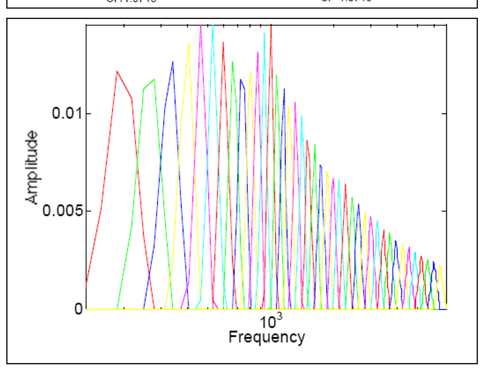




It is known that human auditory system uses a form of Fourier analysis for speech signal processing. This serves as motivation for using spectral analysis for analyzing speech signal.
During speech signal analysis, frequency resolution of human ear should be taken into account. There is no reason to analyze sounds which contain no valuable information for us. Thus, we will use mel scale. Mel scale is experimentally formed scale (analytical form was created later) by asking listeners to determinate pitches of sound which are equally spaced one from another. As a result, mel scale is approximately linear below 1000Hz and logarithmic above. This means that human ear perceives better sounds with lower pitch.







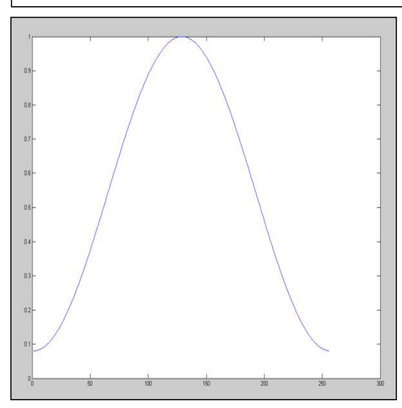
To perform these steps, mel based filter bank is used. Filter bank attempts to decompose signal into a discrete set of spectral features that contain information similar to those presented to the higher levels of processing in human auditory system.
The filter bank is constructed using 13 linearly-spaced filters (133.33Hz between center frequencies,) followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency.)

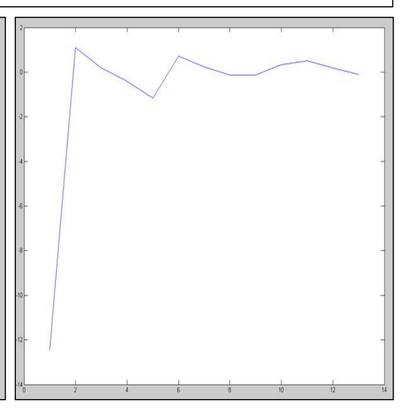
Before a sample goes through filter bank, it is first passed through preemphasize filter described by following equation:

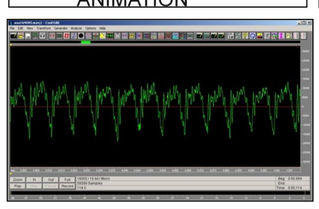
$$H_{pre}(z) = 1 + a_{pre} z^{-1}$$

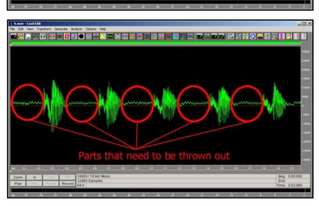
Typical values for a_{pre} are [-1.0, -0.4]. This filter is used for boosting a spectrum of signal by 20db by decade (which is approximately an order of magnitude in frequency). Next step is to window the signal using Hamming window, described by relation:

$$w_h(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N-1}\right)$$
$$0 \le n \le N-1$$



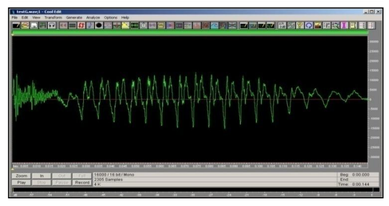




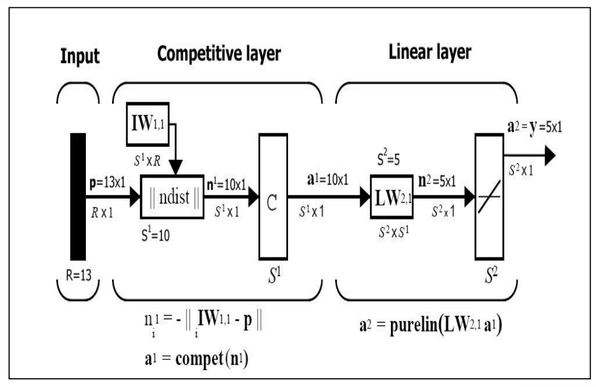


Using a computer system with a sound card and microphone, we have recorded phonemes, performing A/D conversion. Since frequency range of human speech lies in range of 300Hz-8kHz (although there are sounds that go up to 10kHz), by Nyquist, we used sampling frequency of 16kHz. Recording was done using PCM modulation (Pulse Coded Modulation). Since dynamic range of human ear is 120 dB, 20 bits would be required for forming digital word. But, most of the speech lies in range of 70 dB, which corresponds to representation in 12 bits. Thus, since we had to use representation that is power of 2, 16 bits were used, which corresponds to 96dB in dynamic range.

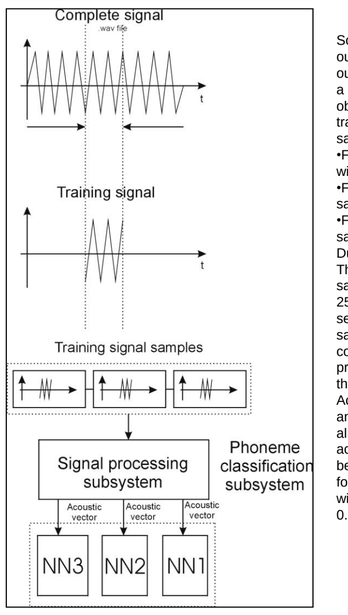
Training samples are used for training a neural network. Typical length of these samples is between 1,5s and 3,5s although only a small part of sample is used for actual training. Speech is recorded in two different ways. First way is by continuous pronunciation of one phoneme.



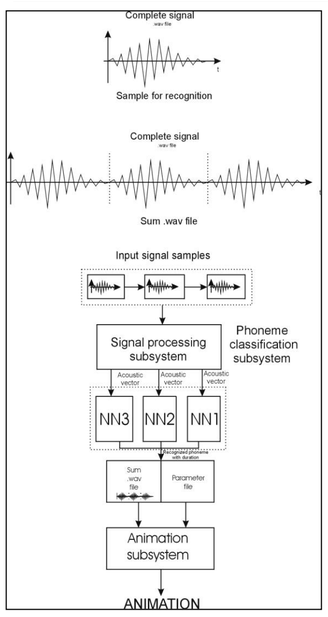
As it can be seen from the picture, during repeated pronunciation there are parts in signal which contain no useful information. Those parts are usually on the beginning and at the end of the file, and in between pronunciation.
By carefully listening some phonemes, such as K or G, we can establish that even some parts of phoneme do not play important role in creating a sound information. Sometimes these parts even cause problems during recognition, so they need to be cut out of the signal. Thus, we have gained continual pronunciation of consonant that is fully functional during training .
Testing samples are recorded as single pronunciation of one phoneme. Part of the signal that carries no information before and after phoneme is cut out.
For signal processing described in this chapter CoolEdit96 was used.



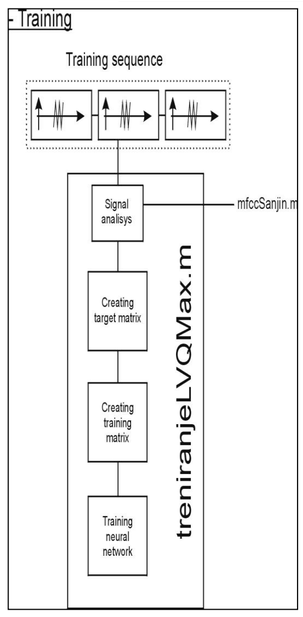
Self-organizing networks are one of the most fascinating topics in the neural network field. Such networks can learn to detect regularities and correlations in their input and adapt their future responses to that input accordingly. The neurons of competitive networks learn to recognize groups of similar input vectors. They are called competitive since neurons in layers of these network compete for the right to produce an output. Neuron that creates an output is called „the winner". Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors.
Learning Vector Quantization (LVQ) is a method for training of a competitive layer in a supervised manner. This means that during training of the network (that is during adjusting network weights) there is "a teacher" who explicitly gives target values for each input pattern. Thus network knows exactly which value to produce on net output for given input value.

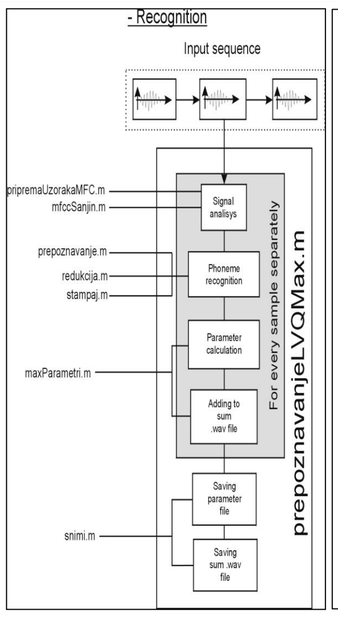


So, before we can start recognizing with our network we must train it. For training of our network, we used samples obtained in a manner described previously. For obtaining acoustic vector with which we train our network, we do not use complete sample, but only small part of it:
•For vowel classification network a sample with length of 20 ms;
•For semivowel classification network a sample with length of 500 ms;
•For consonant classification network a sample with length of 500 ms;
During analysis, window of 10 ms is used. Thus, for vowels we gain 25 training samples for one phoneme (because of 25% of overlap between each window), for semivowels and consonants 62 training samples per phoneme. Network receives column vectors with 13 values (described previously). Training samples are given to the signal analysis subsystem in sequence. Acoustic vectors obtained by signal analysis are stored in matrix. Once when all samples are analyzed, and resulting acoustic vectors are in the matrix, we can begin the training. This process is repeated for each neural network. Training was done within 250 iterations with learning rate of 0.01.

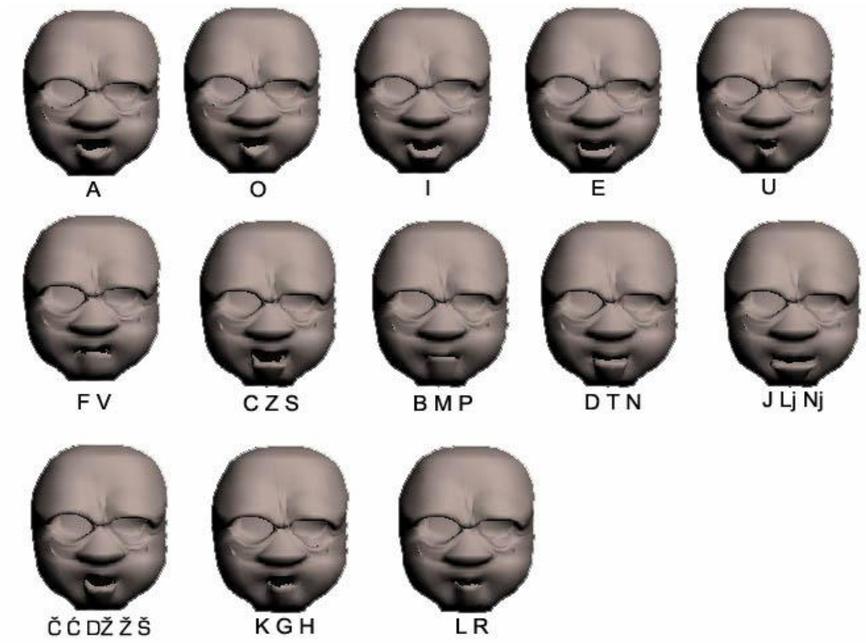


For recognition we used complete sample of recorded phoneme. Patterns that need to be recognized are given to the network in sequence. During signal analysis of given sample, as with the training samples, window of 10 ms was used with overlap of 25%. Matrix of recognized phonemes within one sample is forwarded to reduction function and then to printing function for output on screen. After recognition, calculation of phoneme length is performed. Result is in frames. One frame is 1/25 of a second, according to PAL standard. Length is then stored in a matrix along with other phoneme lengths in consecutive order. After all sound files given as input for recognition has been processed in this manner, matrix is saved as a textual parameter file. Separate .wav files are woven into one sum .wav file. Recognition part of the system is realized within prepoznavanje LVQMax.m file. Function of signal analysis used during recognition is realized in mfccSanjin.m and pripremaUzorakaMFC.m files. All other procedures and functions used are standard Matlab toolbox for neural networks.

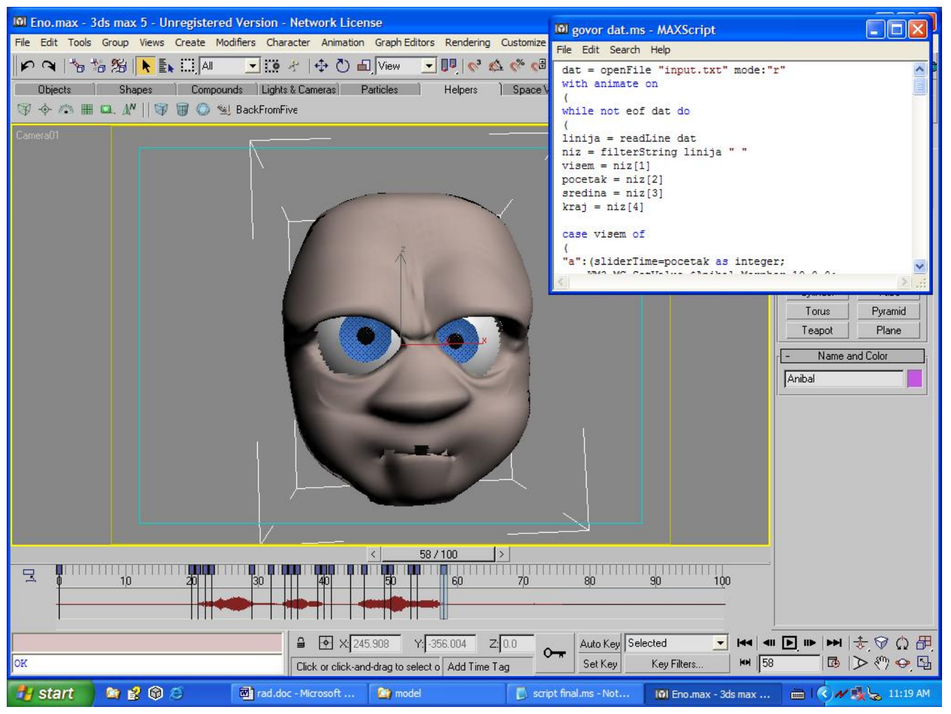




Training of the system is realized in treniranjeLVQMax.m file. Function of signal analysis used during recognition is realized in mfccSanjin.m file. Recognition of phonemes is realized in prepoznavanje.m, redukcija.m and stampaj.m. Their function is:
• File Prepoznavanje – performs recognition of given sample with already trained network;
•Reduction – result of each recognized phoneme is placed into a result matrix. Result matrix contains consecutive values of each recognized phoneme. Function reduction performs counting of counting of each occurrence of recognized phoneme within given sample. Final result, that is a recognized phoneme, is a phoneme with largest number of occurrences. Output from this procedure is a 1x1 matrix that contains recognized phoneme;
•Command Stampaj – performs printing of recognized phoneme on the screen. Recognized phoneme is in fact stored in resulting 1x1 matrix as a class number within neural network. Thus this procedure performs decoding of a class into a phoneme character.
Parameter calculation and creation of sum .wav file is performed within maxParametri.m file. Recording of textual parameter file and sum .wav file is performed with snimi.m file.



This paper described the present stage of our research in automatic animation of Bosnian language phonemes. We recognized phonemes using LVQ neural network in order to generate a parameter file for the automatic animation system based on 3ds max.



Morpher modifier is a data structure designed to handle particular number of channels and their parameters. Assigning this modifier to an object allows the animator to make the animation by morphing a basic object from one copy to another in time. After visemes are created we assign them to Morpher modifier channels as morph targets. Our algorithm provides automatic key frame generation for these visemes using MaxScript scripting program .
MaxScript is the built-in scripting language for 3ds max. It's an object oriented programming language working with classes, objects and methods.
MaxScript provides a set of methods working with Morpher modifier. Using these methods we can add or delete Morpher modifier channels, check their range, usage etc.
The algorithm has the following phases:
•initialization, file opening
•viseme check
•key frame creation in corresponding Morpher modifier channel
•end of file check
•final animation rendering
MaxScript control script uses a parameter file with phonemes and their start and end frames as an input file. Each line of the file is parsed. By a case expression is determined which phoneme appeared. The corresponding viseme channel of Morpher modifier is set to 100% at start time and to 0% at end time. By performing this operation in animate context, key frames are created.
After reaching end of file, the animation could be rendered by the script using previously defined render parameters such as camera name, output file size, interval in frames and output file name .
The animator is now able to fine tune the animation and add the emotions visemes to the model.

Using that parameter file as an input parameter of the MaxScript control script we performed the hardest work for an animator, creating the keyframes of Morpher modifier channels containing visemes that are corresponding to the phonemes of the prerecorded soundtrack. After this process, the animator is able to fine tune the animation and add the facial expressions of the emotions.
Our further work will be dedicated to phoneme classification using neural networks. At first we should classify the phonemes by three basic phoneme types in Bosnian language and after we will perform the classification of particular phonemes inside that type.
The following phase of our research will contain experiments in Bosnian language phoneme recognition in real time, that should be the basis for various applications in computer animation, computer assisted language learning, development of virtual environment avatars etc.