AUTOMATIC LIP SYNCHRONIZATION BY SPEECH SIGNAL ANALYSIS

AUTOMATSKA SINKRONIZACIJA USANA POMOĆU ANALIZE GOVORNOG SIGNALA

MASTER THESIS
MAGISTARSKI RAD

Zagreb, 2005.
Automatic Lip Synchronization by Speech Signal Analysis

Master thesis written at the Department of
Telecommunications, Faculty of Electrical
Engineering and Computing,
University of Zagreb.

Mentor: Doc. dr. sc. Igor S. Pandžić
This thesis has 74 pages.
Thesis no.: 03-Ac-17/2002-z

Magistarski rad je izrađen na Zavodu za
Telekomunikacije Fakulteta elektrotehnike
i računarstva, Sveučilišta u Zagrebu.

Mentor: Doc. dr. sc. Igor S. Pandžić
Magistarski rad ima 74 stranice.
Rad br.: 03-Ac-17/2002-z
Thesis Evaluation Committee:

1. Dr. Ignac Lovrek, Professor, FER
2. Dr. Igor S. Pandžić, Assistant Professor, FER
3. Dr. Robert Manger, Professor, PMF – Department of Mathematics, Zagreb

Thesis Defense Committee:

1. Dr. Maja Matijašević, Professor, FER - President
2. Dr. Igor S. Pandžić, Assistant Professor, FER - Mentor
3. Dr. Robert Manger, Professor, PMF – Department of Mathematics, Zagreb

Date of thesis defense: October 21\textsuperscript{st}, 2005.
Contents

CONTENTS .................................................................................................................. IV

LIST OF FIGURES .................................................................................................... VII

LIST OF TABLES ...................................................................................................... IX

1 INTRODUCTION .......................................................................................... 1

1.1 IDEA .......................................................................................................... 1

1.2 OBJECTIVES ............................................................................................. 2

1.3 THESIS OUTLINE ....................................................................................... 2

2 BACKGROUND ............................................................................................... 4

2.1 DIGITAL SPEECH PROCESSING ................................................................. 7

2.1.1 MFCC ..................................................................................................... 7

2.1.2 FLDT ..................................................................................................... 9

2.2 AUDIO TO VISUAL MAPPING .................................................................... 9

2.2.1 NN ....................................................................................................... 9

2.2.2 GMM .................................................................................................. 12

2.2.3 VQ ..................................................................................................... 13

2.2.4 HMM .................................................................................................. 13

2.3 FACE ANIMATION IN MPEG-4 ................................................................. 14

2.4 EXISTING TECHNIQUES .......................................................................... 16

2.4.1 NN based techniques ....................................................................... 17

2.4.2 HMM based techniques ................................................................... 17

2.4.3 Other techniques ............................................................................... 18
3 THE PROPOSED LIP SYNCHRONIZATION ALGORITHM .......... 19

3.1 BASIC SYSTEM FOR LIP SYNC ................................................................. 19
  3.1.1 Phoneme database ........................................................................ 20
  3.1.2 Audio to Visual Mapping ................................................................. 20
  3.1.3 MPEG-4 Face Animation ................................................................. 28

3.2 IMPROVED SYSTEM FOR LIP SYNC ....................................................... 28
  3.2.1 Adjustments for characteristics of Croatian language ..................... 29
  3.2.2 Viseme based training of NNs ......................................................... 31
  3.2.3 GA and NNs in our approach .......................................................... 33
  3.2.4 Synchronization .............................................................................. 35

4 IMPLEMENTATION .................................................................................. 36

4.1 LIP SYNC MODULE ............................................................................. 36

4.2 DEMO APPLICATION ............................................................................ 39

5 SYSTEM VALIDATION ............................................................................. 42

5.1 TESTING METHODS ............................................................................. 42
  5.1.1 Neural network simulation .............................................................. 43
  5.1.2 Lip Sync Test Application ............................................................... 44
  5.1.3 Subjective testing using video generated from animation............... 47

5.2 TESTING RESULTS ................................................................................ 48
  5.2.1 Validation of recognition with neural networks ............................... 48
  5.2.2 Testing results in different conditions .............................................. 54

5.3 DISCUSSION .......................................................................................... 56

6 APPLICATION SCENARIOS .................................................................... 59

6.1 VIRTUAL CHARACTER AS A WEB GUIDE ........................................... 61

6.2 VIRTUAL PRESENTATION ..................................................................... 62
List of Figures

Figure 1: A basic idea of lip sync ................................................................. 4
Figure 2: A procedure for MFCC extraction ................................................. 9
Figure 3: An abstract neuron ..................................................................... 10
Figure 4: Feature points ........................................................................... 14
Figure 5: FAPUs ......................................................................................... 16
Figure 6: Scope variation of the first and the second coefficient of the phonemes /a and /e before and after FLDT ................................................................. 21
Figure 7: Scope variation of the first and the second coefficient of the phonemes /a and /b before and after FLDT ................................................................. 22
Figure 8: Scope variation of the first and the second coefficient of the phonemes /h and /f before and after FLDT ................................................................. 23
Figure 9: an audio preprocessing used ....................................................... 24
Figure 10: The MFCCs before FLDT of the phonemes /a and /b ...................... 24
Figure 11: The MFCCs after FLDT of the phonemes /a and /b ...................... 25
Figure 12: 75% overlapping of the frames. The frame length is 256 samples [10].. 25
Figure 13: A two layer neural network ....................................................... 26
Figure 14: Activation functions for the neurons .......................................... 27
Figure 15: A comparison of the phoneme and viseme based training considering the time needed for one training cycle ......................................................... 32
Figure 16: Overview of the demo application .............................................. 40
Figure 17: Menu in lip sync from audio file mode ......................................... 40
Figure 18: GUI of our application ............................................................... 41
Figure 19: Methods used for validation of the lip sync system ...................... 42
Figure 20: A basic idea of the Lip Sync Test Application .............................. 44
Figure 21: Wave and spectral form of the ground truth sentence ............... 45
Figure 22: Percentage of correctly recognized visemes in the online and offline mode obtained by Lip Sync Test Application ......................................................... 53
Figure 23: Number of correctly recognized visemes in the online and offline mode compared with total number of visemes that appear in the ground truth sentence ................................................................. 53

Figure 24: Scores achieved for the first and the second question in the subjective testing......................................................................................................................... 55

Figure 25: An overview of the face animation system using lip sync for creation of the animation contest..................................................................................................... 60

Figure 26: A virtual human as a web guide ................................................................. 61

Figure 27: A symbolic view of the virtual presenter use................................................ 63

Figure 28: Scenario for personalized multimedia messaging on the mobile phones 65

Figure 29: A symbolic view of the synthetic video conference calls ......................... 66
List of Tables

Table 1: Visemes and related English phonemes ...................................................... 6
Table 2: FAP groups ................................................................................................ 15
Table 3: Primary facial expressions as defined for FAP2 ....................................... 15
Table 4: Croatian phonemes and the words used to extract the phonemes .......... 29
Table 5: Croatian phonemes and related MPEG-4 viseme groups .................... 31
Table 6: Results of initial experiments for GA parameter determination .......... 34
Table 7: An overview of the Lip Sync module characteristics ............................. 37
Table 8: Questions in subjective test Questionnaire ............................................. 48
Table 9: Validation results of the initial system .................................................... 49
Table 10: Validation results of viseme based NN training .................................. 50
Table 11: Validation results obtained by neural networks configured with genetic algorithms ........................................................................................................ 50
Table 12: Characteristics and result of final neural networks ............................. 51
Table 13: Validation results obtained by neural networks configured with genetic algorithms and with extended database used .............................................. 52
Table 14: Answering form for subjective test ....................................................... 55
Table 15: General impression of seen videos of several tested subjects ............. 56
1 Introduction

A human speech is bimodal in its nature [1]. A speech that is perceived by a person depends not only on the acoustic information, but also on the visual information such as lip movements or facial expressions. In noisy environments, a visual component of a speech can compensate for a possible loss in speech signal. This combination of the auditory and visual speech recognition is more accurate than only auditory or only visual. Use of multiple sources generally enhances a speech perception and understanding. Consequently, there has been a large amount of research on incorporating bimodality of a speech into the human-computer interaction interfaces. A speech-driven face animation is one of the research topics in this area.

The goal is to animate the face of a speaking avatar (i.e. a synthetic 3D human face) in such a way that it realistically pronounces the given text, which is based only on the speech input. Especially important component of facial animation is the movement of lips and the tongue during speech. For a realistic result, lip movements must be perfectly synchronized with the audio.

1.1 Idea

The goal of this master thesis work is to propose and implement a system for automatic lip synchronization by speech signal analysis that will be suitable for real-time and offline applications. A particular sub-goal is to optimise the system for specificity of the Croatian language.

The system must analyse an audio signal containing speech and classify it into lip shape categories (visemes) in order to synchronize the lips of a computer generated face with the speech. The animation is already implemented, so the work done in this thesis is focused on the signal processing of an audio signal and the implementation of audio to visual mapping and synchronization. It is very important that a facial
animation and a sound are synchronized. However, in the real time use, some time delay must be accepted, since a speech has to be spoken before it can be classified.

A basic lip sync system, developed by Linköping University, is used as a starting point for this work [10].

1.2 Objectives

The objectives of this master thesis are listed below:

- To investigate models for the automatic lip synchronization by speech signal analysis
- To create detailed report about existing systems
- To construct and to implement an automatic lip sync system which will be suitable for real time applications as well as for offline production
- The system will be adjusted to characteristics of the Croatian language
- The testing of the system will include several different parameters such as the quality of the input signal (noise, background speech) or different languages (Croatian, English, German and Swedish)
- To create detailed report about the behaviour of the system in the different conditions

1.3 Thesis outline

This thesis is organized into following chapters:

- Chapter 1 (Introduction)
- Chapter 2 (Background) - gives the background information and the idea behind this master thesis work. Existing techniques and basic methods are presented here as well
- *Chapter 3* (The Proposed Lip Synchronization Algorithm) describes the basic lip synchronization system and improvements made in it

- *Chapter 4* (Implementation) contains the short description of the implementation of our system and its possibilities

- *Chapter 5* (System validation) presents achieved results and reports about the behaviour of the system in different conditions

- *Chapter 6* (Application Scenarios) gives an overview of possible scenarios for the use of our lip sync application

- *Conclusion*
2 Background

Lip synchronization is the determination of the motion of the mouth and tongue during speech [2]. A speech sound is produced by the vibration of the vocal cords in the case of voiced sounds and air turbulence in the case of whispered sounds [3]. A vocal tract, which consists of the throat, mouth, tongue, teeth, lips and nasal cavity, additionally models the produced sound. Vowels are created by the relatively free passage of the breath through the larynx and oral cavity, while consonants are produced by a partial or complete obstruction of the air stream by any of various constrictions of the speech organs. Intonation characteristics are pitch, amplitude and voiced/whispered quality and they are dependent on the sound source, while the vocal tract determines the phoneme.

A phoneme is the basic unit of the acoustic speech. A visual representation of the phoneme is called viseme. There are many sounds that are visually ambiguous when pronounced. Therefore, there is a many-to-one mapping between phonemes and visemes. To make lip sync possible, position of the mouth and tongue must be related to characteristics of the speech signal. Positions of the mouth and tongue are functions of the phoneme and are independent of intonation characteristics of speech.

![Figure 1: A basic idea of lip sync](image)

A basic idea of lip synchronization is shown on Figure 1. The process of the automatic lip sync consists of two main parts. The first one, audio to visual mapping, or more specific speech to lip shape mapping, is key issue in bimodal speech processing. In this first phase speech is analysed and classified into viseme
categories. In the second part, calculated visemes are used for the animation of virtual character's face. The animation is not the topic of the interest in this work as it is already implemented in the Visage Technologies [4] software on which our application is based, so it is briefly described.

The problem of converting a speech signal to the lip shape information can be solved on several different levels, depending on the speech analysis that is being used [5]. These levels are:

- Front end (signal level)
- Acoustic model (phoneme level)
- Language model (word level)

Each of the three levels can be applied within the speech-driven face animation system. However, the choice will depend on a specific application, considering characteristics of the individual solution. In addition, a balance between time needed for the signal processing and the quality to be achieved must be found.

A signal level concentrates on a physical relationship between the shape of the vocal tract and the sound that is produced. The speech signal is segmented into frames. A mapping is then performed from acoustic to visual feature, frame by frame. This method uses a large set of audio-visual parameters to train the mapping. There are many algorithms that can be modified to perform such mapping – Vector Quantization (VQ), the Neural Networks (NN), the Gaussian Mixture Model (GMM), etc.

At the second level, speech is observed as a linguistic entity. The speech is first segmented into a sequence of phonemes. Mapping is then found for each phoneme in the speech signal using a lookup table, which contains one visual feature set for each phoneme. The standard set of visemes is specified in MPEG-4 and contains 15 static visemes that can be easily distinguished [21] (Table 1).

The language model is more concerned about context cues in the speech signals. A speech recogniser must be first used for segmenting the speech into words. Then a Hidden Markov Model (HMM) can be created to represent the acoustic state transition in the word. In the next step, one of the methods used in the first level, can
be applied for each state in this model to perform mapping from audio to visual parameters, frame by frame. Because the mapping is modelled inside individual words, better results can be achieved using this solution.

<table>
<thead>
<tr>
<th>viseme</th>
<th>phonemes</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
<td>na</td>
</tr>
<tr>
<td>1</td>
<td>p, b, m</td>
<td>put, bed, mill</td>
</tr>
<tr>
<td>2</td>
<td>f, v</td>
<td>far, voice</td>
</tr>
<tr>
<td>3</td>
<td>T, D</td>
<td>think, that</td>
</tr>
<tr>
<td>4</td>
<td>t, d</td>
<td>tip, doll</td>
</tr>
<tr>
<td>5</td>
<td>k, g</td>
<td>call, gas</td>
</tr>
<tr>
<td>6</td>
<td>tS, dZ, S</td>
<td>chair, join, she</td>
</tr>
<tr>
<td>7</td>
<td>s, z</td>
<td>sir, zeal</td>
</tr>
<tr>
<td>8</td>
<td>n, l</td>
<td>lot, not</td>
</tr>
<tr>
<td>9</td>
<td>r</td>
<td>red</td>
</tr>
<tr>
<td>10</td>
<td>A:</td>
<td>car</td>
</tr>
<tr>
<td>11</td>
<td>e</td>
<td>bed</td>
</tr>
<tr>
<td>12</td>
<td>I</td>
<td>tip</td>
</tr>
<tr>
<td>13</td>
<td>Q</td>
<td>top</td>
</tr>
<tr>
<td>14</td>
<td>U</td>
<td>book</td>
</tr>
</tbody>
</table>

The latter two approaches are providing more precise speech analysis. Acoustic speech signal is explored together with the context, so that co-articulations (co-articulation is a process by which one sound effects production of the neighbouring sounds) are incorporated. However, higher input signal level requires a more complex system. At the same time, because the motion of the lips, tongue and mouth can be found from the speech signal without previous recognition of phonemes or spoken words, these methods produce a certain amount of computation overhead. Another problem with phoneme level approach is the definition of different phonemes in different languages, so that there is no standard phoneme set [6]. Additionally, speaker’s gender, dialect or co-articulation could be an obstacle for obtaining a correct segmentation of the phonemes for individual’s speech.
On the other hand, an approach based on the low level acoustic signals is simple, language independent and suitable for the real-time implementation, what is not the case in acoustic model where a speech engine have to be incorporated in the system in order to obtain a phoneme sequence for a given speech.

An overview of the existing techniques used for lip synchronization is given at the end of this chapter. But first, a brief introduction on the digital speech processing and the methods used in this work is given. As well, basic techniques used for the audio to visual mapping and the basics of the MPEG-4 compliant facial animation will be discussed here. The purpose of these lines is to give the reader an opportunity to acquire knowledge needed to fully understand the chapters that follow.

2.1 Digital speech processing

Digital speech processing, as any signal processing, involves first obtaining representation of the signal and then the application of some higher level transformation in order to put signal in suitable form [7]. The last step is extraction and utilization of information needed for the specific application.

Automatic lip sync can be assumed as some kind of the speech recognition, which is one typical speech communications application [9]. In this example, information in the signal is the class of the phoneme. Signal transformation based on the suitable speech representation must enable classification of the speech into phonemes. This is necessary step in the process of mapping the speech to lip movements.

The techniques used in this work are briefly described, since this is not the area of the research.

2.1.1 MFCC

The Mel-Frequency Cepstrum Coefficients (MFCC) [8] is an audio feature extraction technique which extracts parameters from the speech similar to ones that are used by humans for hearing speech, while at the same time, deemphasizes all other
information. As MFCCs take into consideration the characteristics of the human auditory system, they are commonly used in the automatic speech recognition systems [10]. Additionally, these coefficients are robust and reliable to variations according to speakers and recording conditions.

The speech signal is first divided into time frames consisting of an arbitrary number of samples. In most systems overlapping of the frames is used to smooth transition from frame to frame. Each time frame is then windowed with Hamming window to eliminate discontinuities at the edges. The filter coefficients $w(i)$ of a Hamming window of length $n$ are computed according to the formula:

$$w(i) = 0.54 - 0.46 \cos\left(\frac{2\pi i}{n}\right)$$

After the windowing, Fast Fourier Transformation (FFT) is calculated for each frame to extract frequency components of a signal in the time-domain. For signal $x(n)$, the spectrum of $N$ samples is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-2\pi nk/N}$$

The logarithmic Mel-Scaled filter bank is applied to the fourier transformed frame. The mel-scale is a perceptually based subjective scale. This scale is approximately linear up to 1kHz, and logarithmic at greater frequencies:

$$m = 1127 \cdot \ln(1 + \frac{f}{700})$$

The last step is to calculate Discrete Cosine Transformation (DCT) of the outputs from the filter bank. DCT is defined as:

$$X(k) = w(k) \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{\pi (2n+1)k}{2N}\right), k = 0, \ldots, N - 1$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0 \\ \frac{2}{\sqrt{N}}, & k = 1, 2, \ldots, N - 1 \end{cases}$$
DCT ranges coefficients according to significance, whereby the 0th coefficient is excluded since it is unreliable.

The overall procedure of MFCC extraction is shown on Figure 2.

![Figure 2: A procedure for MFCC extraction](image)

### 2.1.2 FLDT

Fisher linear discriminate transformation (FLDT) is a classification method that reshapes the scatter of a data set to maximize class separability [11]. If there is no separation between classes before FLDT, transformation will not enhance separability, where as if there is only slight distinction between classes, the FLDT will separate them satisfactory [10].

### 2.2 Audio to visual mapping

Many approaches have been proposed in an attempt to solve the problem of extracting the mouth shape information from the speech signal. A background of the most used techniques is given.

#### 2.2.1 NN

Artificial neural networks (ANN) are simulation of the information processing capabilities of nervous systems [12]. Just as humans apply knowledge gathered from the past experience to new problems, a neural network uses previously solved examples to build system capable for solving new problems.

The brain is composed of a huge number of interconnected neurons. Analogously, the neural network is built-up of many artificial neurons, called perceptrons.
Figure 3 shows the structure of the perceptron with $n$ inputs. Usually, each input $x_i$ has associated weight, which means that the incoming information is multiplied by the corresponding weight $w_i$. The transmitted information is integrated at the neuron by summing weighted input signals. The primitive (activation) function is then evaluated in order to limit and determine the output of the neuron. Input to the activation function is usually modified by applying external bias.

Three elements are particularly important in any model of artificial neural networks [12]:

- The structure of the nodes (activation function)
- The topology of the network (number of hidden layers, number of nodes in each layer and connectivity)
- The learning algorithm used to find the weights of the network

A major unanswered question in NN research is how to set the series of configuration parameters best so as to maximize the network's performance. Finding a suitable network for a specific problem has proved to be difficult.

The most widely used neural network is a multilayer feedforward network which projects an input layer of nodes onto output layer through a number of hidden layers.
In such networks, a backpropagation algorithm is usually used as training algorithm for adjusting weights [13].

Neural networks are widely used for mapping between the acoustic speech and the appropriate visual speech movements [10]. In the training phase, input patterns and output patterns are presented to the network. Suitable learning algorithm, as well as the number of hidden layers and the number of nodes per layer should be determined. Besides, a single network can be trained to reproduce all the visual parameters as well as many networks can be trained so that each network estimates a single visual parameter.

Training of NNs will be described in more details in the next chapter.

Many parameters, such as weights, topology, learning algorithm, training data, transfer function and others can be controlled in the neural network [14]. As training neural network is an optimisation process where the error function of a network is minimized, GA can be used to search optimal combination of parameters [12].

**2.2.1.1 GA**

Genetic algorithms (GA) are a method for solving optimisation or search problems inspired by biological processes of inheritance, mutation, natural selection and genetic crossover. A conventional GA consists of three essential elements [12]:

- A coding of the optimisation problem
- A mutation operator
- A set of information-exchange operators

The coding of the optimisation problem produces discrimination of the values, which represent possible solutions to the problem being solved. GA starts with a randomly selected population of elements. Each of these elements, called a genome, is formed by a string of values (called genes).

All of the genomes are evaluated by a fitness function which determines how well the solution that gene codes solves the problem. Once the fitness of each member of the population is known, elements are sorted with those having better fitness at the top. New solutions are generated by combining genomes selected according the
fitness at chosen crossover point. After performing crossover, one of the positions in an offspring can be mutated. The mutation operator determines the probability with which the data structures are modified. These new solutions also get evaluated based on their ability to solve the problem. From the original solutions combined with the ones just created, a certain number of genomes are kept in the population, discarding the ones with the lowest fitness. The process is continued for a specified number of generations, until a particular fitness value is achieved, or until some other predetermined stopping criterion (maximum number of generations or consistency of fitness) is reached.

GAs might be used to help design neural networks by determining [15], [17], [18]:

- **Weights.** Algorithms for setting the weights by learning from presented input/output examples with given fixed topology often get stuck in local minima. GAs avoid this by considering many points in the search space simultaneously.

- **Topology.** Determining a good/optimal topology is even more difficult – most often, an appropriate structure is created by intuition and time consuming trial and error.

- **A suitable learning rule**

However, they are generally not used in all three problems at a time, since they are computationally very expensive. Therefore introducing parallelism in GAs is important task [16].

### 2.2.2 GMM

Gaussian mixture model is a type of density model which comprises a number of component Gaussian functions. These component functions are combined to result in a multi-modal density.

Gaussian mixture can be used to model the probability distribution of the audio-visual vectors [5]. Joint feature vector is composed from collected training data. Then the best estimation of the visual parameters is derived directly from the
composed vector. The Expectation-Maximization (EM) algorithm is used for fitting a mixture model to a set of training data. The algorithm operates in two steps. The first one estimates the distribution given the data and current value of the parameters, while the second one finds the new parameter set that maximizes the probability distribution [10]. The result is a mixture model with the probability density function for each class.

After GMM is trained, the model is used to map an audio feature to the visual feature. The Gaussian mixture approach produces smoother results than vector quantization.

### 2.2.3 VQ

Vector quantization is lossy data compression method which maps k-dimensional vectors into a finite set of vectors. It can be used as a classification method in audio to visual conversion.

First, the audio features (training data) are classified into one of a number of classes [23]. For each acoustic class, the corresponding visual representatives are averaged. Each class is then mapped into a corresponding visual output. Therefore, each input acoustic feature would be classified using an optimal acoustic vector quantizer and then mapped to the corresponding visual output. VQ is computationally efficient, but it does not produce a continuous mapping.

### 2.2.4 HMM

A Hidden Markov model [19] is a statistical model where the system being modelled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters, from the observable parameters, based on this assumption. The extracted model parameters can then be used to perform further analysis, for example for speech recognition applications. Moreover, HMM can be used for the audio to visual parameter conversion.
The HMM based method maps from an input speech signal to lip parameters through HMM states. The states must be inferred from the signal and this is done by different algorithms, such as Viterbi or Baum-Welch algorithm [20]. At each state, the visual parameters can be estimated to given acoustic parameters.

2.3 Face Animation in MPEG-4

A face animation (FA) is supported in MPEG-4 standard [21]. MPEG-4 FA specifies a face model in its neutral state, a number of feature points (FPs) and a set of Facial Animation Parameters (FAPs). There are 84 FPs on the neutral face serving as reference points for defining FAPs (Figure 4).

![Figure 4: Feature points](image)

The 68 FAPs are categorized into 10 groups (Table 2). Each FAP corresponds to a particular facial action deforming a face model in its neutral state. FAPs in groups 2-10 are low-level parameters and they directly move FP of a face for given amplitude. The first group contains high-level parameters, visemes and expressions. A viseme is visual representative of a phoneme. Only 15 static visemes are included in the standard set (Table 1), including the neutral face. The expression parameter defines the 6 primary facial expressions (Table 3). Parameters enable mixing of two visemes or facial expressions by using weighting factor.
Table 2: FAP groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of FAPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: visemes and expressions</td>
<td>2</td>
</tr>
<tr>
<td>2: jaw, chin, inner lowerlip, cornerlips, midlip</td>
<td>16</td>
</tr>
<tr>
<td>3: eyeballs, pupils, eyelids</td>
<td>12</td>
</tr>
<tr>
<td>4: eyebrow</td>
<td>8</td>
</tr>
<tr>
<td>5: cheeks</td>
<td>4</td>
</tr>
<tr>
<td>6: tongue</td>
<td>5</td>
</tr>
<tr>
<td>7: head rotation</td>
<td>3</td>
</tr>
<tr>
<td>8: outer lip positions</td>
<td>10</td>
</tr>
<tr>
<td>9: nose</td>
<td>4</td>
</tr>
<tr>
<td>10: ears</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3: Primary facial expressions as defined for FAP2

<table>
<thead>
<tr>
<th>expression_select</th>
<th>expression name</th>
<th>textual description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>1</td>
<td>joy</td>
<td>The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.</td>
</tr>
<tr>
<td>2</td>
<td>sadness</td>
<td>The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.</td>
</tr>
<tr>
<td>3</td>
<td>anger</td>
<td>The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.</td>
</tr>
<tr>
<td>4</td>
<td>fear</td>
<td>The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.</td>
</tr>
<tr>
<td>5</td>
<td>disgust</td>
<td>The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.</td>
</tr>
<tr>
<td>6</td>
<td>surprise</td>
<td>The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.</td>
</tr>
</tbody>
</table>
In order to define FAPs for arbitrary face models, MPEG-4 defines Facial Animation Parameter Units (FAPUs) that scale FAPs for any face model. FAPUs are defined as distances between key feature points (Figure 5).

![Figure 5: FAPUs](image)

### 2.4 Existing techniques

Various solutions have been proposed in the field of the automatic lip synchronization. A basic difference between them is the level on which the speech signal is analysed. There out arise some other characteristics of the system, such as applicability on real time systems, language independence, co-articulation, computation complexity/accuracy etc.

For example, systems that use Hidden Markov Model take into consideration the audio contextual information, which is very important for modelling mouth co-articulation during the speech. That is not the case with the vector quantization and the Gaussian mixture model. Neural networks can be trained for audio to visual mapping so that they take into account the audio contextual information (e.g. time-delay neural networks – TDNN [32]). TDNN is more computationally efficient than HMM, but requires a large number of hidden units, which results in high computational complexity during training phase. Many approaches use a combination of the different techniques.
What follows is short description of some of approaches, divided by the basic technique used.

2.4.1 NN based techniques

Hong et al. [23], [24] use NNs together with Motion Units (MUs). MUs are visual representation of the facial deformation and are obtained from the video. When audio-visual database is built, three layer perceptrons are trained to estimate MU parameters (MUPs) from audio. A similar technique is presented in [22] by the same authors. It uses Gaussian mixture model and multilayer neural network to perform auditory-visual speech recognition in the real time.

Massaro [27] trains three layer feed-forward ANN with a huge number of hidden units and as input takes parameters from the previous and the next frame, in addition to the current time frame.

Frank et al. [31] as well use 3-layer-feedforward neural net with back-propagation learning rule, but for the training of the net use manually created animations.

2.4.2 HMM based techniques

Huang et al. [25] combine HMMs with sequence searching in order to animate lip movements from the speech in the real time. Acoustic feature vectors are calculated from the input voice and then the minimum distance between the vectors and the vocal data of the sequences in the base is found. If the distance is larger than a threshold, the face is synthesized by HMM-based method. Otherwise, the face in the corresponding sequence is exported.

Huang et al. [5] implemented a real-time audio to visual mapping using Hidden Markov Model together with Gaussian mixture model.

Brand [20] introduces method for speech-based full facial animation (lip sync, upper-face expressions) from a video. The video is analysed once, for training, and then facial HMM is used to construct the vocal HMM. Similar system is proposed by
Vanroose [28]. Face expressions of all possible visemes and all viseme pair transitions are learned from 3D face dynamics Audio is then segmented and each part is assigned to the HMM state. Finally, Viterbi algorithm is used to determine the most likely state transition sequence.

Tamura *et al.* [30] propose a technique based on an algorithm for parameter generation from HMM with dynamic features for synthesizing synchronized lip movements from auditory input speech signal. Generated parameter sequence contains information of both static and dynamic features of several phonemes before and after the current one, so the synthetic lip motion becomes smooth and realistic, but it is not applicable on the real time systems.

### 2.4.3 Other techniques

Lewis [29] describes a lip sync approach based on a linear prediction. In this approach speech is effectively deconvolved into the sound source and vocal tract filtering components.

Kshirsagar *et al.* [26] train three-layer neural network to classify coefficients derived with the linear predictive (LP) analysis into the vowels. Also, the average energy in the speech signal is used to modulate vowel-vowel and vowel-consonant lip-shape transition and zero crossing rate is used to detect fricatives. The same authors in [34] use Principal Components Analysis (PCA) of facial capture data extracted using a tracking system to form a vector space representation. The similar approach is used in [35]. Kalberer *et al.* learn 3D shape statistics from faces with a few markers. A 3D reconstruction of a speaking face is produced for each video frame and 3D shape statistics are extracted and PCA is used to reduce the dimension of the space.
3 The Proposed Lip Synchronization Algorithm

Our system for automatic lip synchronization is suitable for real-time and offline applications. It is speaker independent and multilingual. Visual representation of phoneme, viseme, defined in MPEG-4 FA, is used for face synthesis. Database used for viseme classification is not audio-visual but auditory only.

The initial version of this lip sync system has been implemented by A. Axelsson and E. Björhall as part of their master thesis of Linköping University [10] and in the collaboration with Visage Technologies AB, Linköping, Sweden [4].

As part of this thesis work, significant improvements to the basic system were introduced: adoption for specificity of the Croatian language, optimisation in the process of viseme classification, automatic design of neural networks with genetic algorithms, use of simple synchronization tricks which generally improve visual impression and implementation of advanced features into lip synchronization application.

These modifications done on the initial system are described later on, but first basic lip synchronization system, available as a starting point, is discussed.

3.1 Basic system for lip sync

Neural networks are used to classify the speech into a sequence of phonemes in this system. In order to obtain training data for the NNs, a training set with phonemes was collected. Input in NNs are MFCCs calculated from the training data and output is a specific phoneme class. Visemes used for MPEG-4 compliant facial animation production are obtained from phonemes using a lookup table.
Implementation of the method is available in both, real time and offline mode. The program reads speech from pre-recorded audio files and continuously performs spectral analysis of the speech. Suitable visemes are shown on the screen.

Next sections present the components of the system.

### 3.1.1 Phoneme database

As a training data, a set of phonemes is collected. For each Swedish phoneme, nine test subjects’ (six male and three female) with different age and accent recorded three different words containing the specific phoneme as the first letter. The whole words are recorded instead of a single phoneme because it is hard to pronounce only one phoneme. Each phoneme is extracted from each of the three words corresponding to the phoneme. The phonemes are manually cut out from the words and stored as wav files. This gives 27 versions of each phoneme in the database. Recordings of the 7 subjects are used for training and the rest for validation. The words are recorded in a noise free environment and with top quality equipment with sampling frequency of 16 kHz with 16 bits accuracy and saved into wav files.

Since there are 37 Swedish phonemes plus recordings of silence in the database, there are 38 subsets of training data available for the neural network.

### 3.1.2 Audio to Visual Mapping

In order to perform audio to visual mapping, the speech is first segmented into the frames. Then, preprocessing and classification into the phoneme classes is performed on every frame of the input speech.

#### 3.1.2.1 Speech Analysis

MFCC representation of the speech is chosen as first step in preprocessing the speech. Additionally, FLDT is done on MFCC vectors to separate classes more. Figure 6, Figure 7 and Figure 8 show the scatter between phonemes /a and /e, /a and /b and /h and /f respectively before and after FLDT.
Figure 6: Scope variation of the first and the second coefficient of the phonemes /a and /e before and after FLDT
Figure 7: Scope variation of the first and the second coefficient of the phonemes /a and /b before and after FLDT
Figure 8: Scope variation of the first and the second coefficient of the phonemes /h and /f before and after FLDT
As it can be seen on the figures, the separation between /a and /e has improved, while the separation between /h and /f remained the same – still poor. Separation between /a and /b also remained almost unchanged, but it was satisfactory even before FLDT.

The overall procedure of the coefficients calculation is shown on Figure 9.

![Figure 9: an audio preprocessing used](image1)

In order to use MFCCs on the speech signal, frame length and the dimension of the MFCC vectors must be determined. The frame length must be chosen, so that the frame contains enough information [10]. The choice is frame length of 256 samples and 12 dimensional MFCC vector (Figure 10). The coefficients in MFCC vectors are ranked according to significance and the information content diminishes towards the end of the vector.

![Figure 10: The MFCCs before FLDT of the phonemes /a and /b](image2)
In the following FLDT, no reduction of dimensions is made (Figure 11). Overlapping of the frames is used to smooth transition from frame to frame (Figure 12), where 75 percent of the samples in the current frame are reused. The phoneme database, 38 subsets of 12-dimensional MFCC vectors, is now used as a training set in order to train neural networks.

![12-dimensional Fisher](image)

**Figure 11:** The MFCCs after FLDT of the phonemes /a and /b

![75% overlapping of the frames](image)

**Figure 12:** 75% overlapping of the frames. The frame length is 256 samples [10]

### 3.1.2.2 Training NNs

In order to be able to separate inputs from each other, the network needs to be trained to learn from its environment. The training process is repeated a number of times. In every iteration, the vectors in the training set are processed through the network and
the corresponding outputs are compared with the desired outputs, followed by adjustments of the weights [10].

In this approach, multilayer feedforward networks are used to map the speech to lip movements. Figure 13 shows a two layer neural network with 12 inputs, 1 hidden layer consisting of 6 neurons and a single neuron output layer. The network takes a 12-dimensional vector as input and produces a single value as output.

![A two layer neural network](image)

Figure 13: A two layer neural network

The training algorithm used for adjusting the weights in this work is the Levenberg-Marquardt backpropagation algorithm [44]. It is gradient based method, extremely fast for small networks (few hundreds of parameters), which tries to avoid local minima by using repeated trainings and randomly initialized weight values [33]. Two different kinds of activation functions are used (Figure 14). For the neurons in the first two layers, the \textit{tansig} function is used and for the output layer, the \textit{logsig}
function is used as it produces outputs within the range of [0,1], which is desired here.

\[
\tan \, \text{sig}(x) = \frac{2}{1 + e^{-2x}}
\]

\[
\log \, \text{sig}(x) = \frac{1}{1 + e^{-x}}
\]

Figure 14: Activation functions for the neurons

The 12-dimensional MFCC vectors are used as inputs to 38 different networks. For each phoneme class, a NN with 12 inputs, a number of hidden nodes and 1 output is trained. The number of hidden layers and the number of nodes per each layer is determined for each network by running a training session several times on the data and studying the classification results. The network for each subset is expected to give 1 as output when the corresponding phoneme is present at the input and 0 otherwise.
3.1.2.3 Lip Shape generator

For every frame of the speech that is classified in one of the phoneme classes, the corresponding viseme need to be determined and sent to the animated face model. Since the visemes defined in MPEG-4 are not developed for a specific language, Swedish phonemes are divided into the viseme class that best describes the phoneme. Because of the network imperfection, output lies within the interval [0,1] and the network that produces the largest output is picked as the correct phoneme [10]. Errors may occur, and an incorrect phoneme can be identified as the correct one. Choosing a viseme in each frame, could cause a sudden discontinuous facial expression. To avoid this, the outputs from neural network for four consecutive frames (1024 samples) are analysed. The viseme class with the largest output sum is chosen as a correct viseme. This results in some time delay from input to output.

3.1.3 MPEG-4 Face Animation

Once the required information is extracted from the speech and the proper visemes are identified, any parameterised face model can be animated. In this work, MPEG-4 standard is used for generating facial animation. More details on the MPEG-4 standard (i.e. MPEG-4 compatible 3D faces, MPEG-4 FA player) can be found in [21].

3.2 Improved system for lip sync

Basic lip sync system performs automatic lip synchronization by analysing the speech signal in the real time and offline mode. However, many limitations of this system are known [10]. Changes done on the initial system try to solve some of the problems.

The system is trained to classify Swedish phonemes. When tested with other languages, as well as with Croatian, the system gave satisfactory results. But for fine-
tuning of the animation, phonemes specific for certain language might be added in the database.

Training of neural networks is the hard part. In the initial system, one neural network for every phoneme class is created. For every network suitable number of hidden nodes needed to be determined. This is laborious work based on the trial and error method. By decreasing number of networks to be trained and by introducing automatic procedure for determination of the network configuration, the problem of training NNs is simplified.

Importance of the synchronization of the lip movements with the speech is unmistakable. It is achieved mostly by the correct classification of the speech into viseme classes. But, the visual perception quality of the face animation can be enhanced using some tricks.

3.2.1 Adjustments for characteristics of Croatian language

The database needed to be extended with those phonemes that appear in the Croatian language but not in the Swedish, since original database is made of the Swedish phonemes only. In order to remain consistent, the procedure for constructing database did not change. That means that 27 versions of each additional phoneme were recorded by nine different speakers.

In the Croatian language there are 31 phonemes. Eleven of them (/ie/, /ɛ/, /iɛ/, /iɛ/, /dʒ/, /d/, /dj/, /nj/, /š/, /z/ and /ž/) do not exist in Swedish, or they differ a little bit. Table 4 shows all Croatian phonemes. The rows with additionally recorded phonemes and associated words used for phoneme extraction are marked.

Table 4: Croatian phonemes and the words used to extract the phonemes

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>/a/</td>
<td>ako</td>
</tr>
<tr>
<td>/e/</td>
<td>evo</td>
</tr>
<tr>
<td>/i/</td>
<td>ili</td>
</tr>
<tr>
<td>/o/</td>
<td>ono</td>
</tr>
</tbody>
</table>
Next, it was to decide which phoneme would belong to which viseme class defined in the MPEG-4 standard. Only the phonemes that exist in the English language are
already divided into viseme classes, as it is shown in Table 1. For the phonemes that exist in Croatian but not in English, appropriate viseme classes needed to be assigned. The result shown in Table 5 is achieved by simply choosing the viseme class that seemed to match the best. The same thing was done in [10] for the Swedish phonemes.

Table 5: Croatian phonemes and related MPEG-4 viseme groups

<table>
<thead>
<tr>
<th>viseme</th>
<th>phonemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>p, b, m</td>
</tr>
<tr>
<td>2</td>
<td>f, v</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>t, d, g, c</td>
</tr>
<tr>
<td>5</td>
<td>k, h</td>
</tr>
<tr>
<td>6</td>
<td>č, dž, š, ž, č, d</td>
</tr>
<tr>
<td>7</td>
<td>s, z</td>
</tr>
<tr>
<td>8</td>
<td>n, l, nj, lj</td>
</tr>
<tr>
<td>9</td>
<td>r</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
</tr>
<tr>
<td>11</td>
<td>e</td>
</tr>
<tr>
<td>12</td>
<td>i, ie, j</td>
</tr>
<tr>
<td>13</td>
<td>o</td>
</tr>
<tr>
<td>14</td>
<td>u</td>
</tr>
</tbody>
</table>

### 3.2.2 Viseme based training of NNs

In our lip sync system, the visemes are used for the speech animation of the synthetic face model. Final goal of the speech analysis is to get visemes. Phonemes that are visual ambiguous, do not need to be separated, since it does not influence animation. We took advantage of this idea and decided not to classify speech into phonemes first and then to map them onto visemes. Instead, we choose more natural approach. The phonemes are first manually mapped onto MPEG-4 visemes which are then used as the main classification target.
Such procedure resulted in 15 neural networks to be trained, each for every viseme class. The reduction in number of neural networks is significant, and training of every network consumes extra time. The benefit of the decreased number of networks is noticeable in the saved time and enhanced simplicity, as each neural network must be created, trained and its weights and biases need to be saved for later use. The experimental trainings of NNs with different number of networks, both based on visemes and phonemes, have confirmed this consideration. Figure 15 demonstrates the difference in amount of time needed for training neural networks to classify speech into lip movements. Although the results are shown only for one training cycle, the benefit is obvious. However, this is not the final number, since to get satisfactory results numerous of iterations must be done. Additionally, pre and post processing of neural networks (preparations for training, saving weighs etc.) consumes some time which also should be added into calculations. Therefore, the total time saved is far bigger than the one showed.

\[
t_{\text{total}} = t \cdot n,
\]

\(n\) - number of networks
\(t\) - time needed for training of one neural network (with ~20 hidden nodes)
\(t_{\text{total}}\) - time needed for training of all networks

<table>
<thead>
<tr>
<th>Phoneme based training</th>
<th>Viseme based training</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n = \sim 38)</td>
<td>(n = 15)</td>
</tr>
<tr>
<td>(t = \sim 10) min</td>
<td>(t = \sim 10) min</td>
</tr>
<tr>
<td>(t_{\text{total}} = 6 - 7) h</td>
<td>(t_{\text{total}} = 2 - 3) h</td>
</tr>
</tbody>
</table>

\(\Delta t = \sim 4\) h (for every training cycle)

*Results obtained on the avarage PC
P4, 2.4 GHz, 768 MB

Figure 15: A comparison of the phoneme and viseme based training considering the time needed for one training cycle

According to the changes in the way of training the NNs, the database is reorganized so that now contains 15 groups, each corresponding to one MPEG-4 viseme. Each viseme class, depending on the number of phonemes that contains, is represented by certain number of samples in the database - it ranges from zero (viseme class 3 at the
moment does not contain phonemes) to 216 (viseme class 6 contains eight phonemes), what makes average more than 80 samples per viseme group.

Another benefit of viseme based training of NNs is that extending the database with new phonemes demands now less time since new networks do not need to be created, but only the training process need to be repeated with new coefficients.

### 3.2.3 GA and NNs in our approach

Since the configuration of neural network need to be optimised for a specific application, we had to find suitable network for our lip sync application. As determining a good or optimal topology is even the most difficult task in design of NN, we tried to solve this problem with genetic algorithms.

For each viseme class, a NN with 12 inputs (as we have 12-dimensional MFCC vectors), a number of hidden nodes and 1 output is trained. The number of hidden layers and the number of nodes per each layer should have been determined for each of 15 networks. This is laborious and time consuming work since the training session must be run until the result is satisfactory. In order to avoid time consuming trial and error method, we have introduced simple genetic algorithm to help find suitable topology for our NNs.

In our example, given the learning rule (*Levenberg-Marquardt*), we used GA for training a backpropagation feedforward network to determine near optimal network topology, including the number of hidden layers and the number of units within each layer.

We use simple genetic algorithm [45], where number of genes specify the number of hidden layers ($n$). Gene maximum and minimum values are defined in the range from zero to $m$, determining the number of nodes per layer. If a value of the single gene is set to zero, the number of hidden layers is decreased, so practically it ranges from zero to $n$. Other parameters that have to be specified are population size, maximum number of generation and mutation rate.
We have done several initial experiments in order to determine the GA parameters. Table 6 shows two initial GA configurations, achieved results and the time needed. As a starting configuration for genetic algorithm following parameters were chosen: two hidden layers \( n = 2 \), maximum of 30 nodes per layer \( m = 30 \), population size \( 30 \) and maximum iterations \( 50 \). In the next iteration number of hidden nodes decreased \( n = 1 \), as well as population size \( 20 \) and maximum number of iterations \( 10 \). First experiment gave better results for the tested classes. However, if we compare the time spent and achieved result, the benefit is not significant, since the solution with a huge number of hidden nodes is rather time consuming, while the results are not dramatically better. Therefore, we have chosen the solution with the smaller number of nodes.

Table 6: Results of initial experiments for GA parameter determination

<table>
<thead>
<tr>
<th>GA parameters</th>
<th>Result</th>
<th>Time</th>
<th>Result</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class 1</td>
<td></td>
<td>Class 14</td>
<td></td>
</tr>
<tr>
<td>Maximum Iterations: 50</td>
<td>65</td>
<td>~20h</td>
<td>46</td>
<td>~25h</td>
</tr>
<tr>
<td>Max Generations With No Change: 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population Size: 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Genes: 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene Maximum Value: 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene Minimum Value: 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mutation Rate: 60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Iterations: 10</td>
<td>63</td>
<td>~6h</td>
<td>45</td>
<td>~6h</td>
</tr>
<tr>
<td>Max Generations With No Change: 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population Size: 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Genes: 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene Maximum Value: 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gene Minimum Value: 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mutation Rate: 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Such configuration of GA seems suitable since larger network increases computation time, but does not give much better results.

By using genetic algorithms, the process of designing neural network is automated. Once the problem to be solved is coded and GA parameters are determined, the whole process is automated. Although it is still a time consuming work, much time is saved since all work is done automatically by computer.
3.2.4 Synchronization

Many elements affect a visual perception quality of the face animation. The most important one is the accuracy of the viseme determination. Of course, the classification is not always correct, and mistakes are made. But all visemes do not have the same influence on the face animation perception [36].

For example, if the speech frame to be analysed contains silence, and the system classifies it in some other viseme group then zero, the observer will notice mistake. In the case, when similar visemes are mixed, such as viseme group 4 and 5 (/t/ and /k/), the mistakes are not so evident. Generally, it is important that the mouth remains closed if they are supposed to be closed and vice versa. Viseme groups zero (silence) and one (/p/, /b/, /m/) are the ones with higher relevance since in this case the lips are closed and by opening them synchronization looses on the quality.

Another trick arises from the nature of a speech production. Before any sound can be heard, a vocal tract (throat, mouth, tongue, teeth, lips and nasal cavity) starts to move. Ideally, the movements of the lips need to be modelled afore the speech [36]. In the lip sync, the face animation is generated from the speech signal and the speech comes first, since some time delay must exist. But adding artificial delay to the speech, better results are achieved. In the case of real time applications this is not sometimes possible.
4 Implementation

A final task was to implement an application capable of performing automatic lip synchronization in the real time as well as offline mode. The implementation is accomplished by integrating Matlab [42] functions with C/C++ using Microsoft Visual C++ 6.0. A simple genetic algorithm used for determining the neural network topology is Black Box GA [45].

Matlab functions are used for the classification of the speech in the visemes, as described in the previous chapter. A speech processing toolbox for Matlab, Voice box [43], is used for the MFCCs calculation. A database construction and creation of 15 neural networks had to be done only once. In the training process, network's biases and weights are extracted and saved for later use. Together with Fisher matrix (obtained by calculating FLDT), biases and weights matrix are loaded in the application.

The integration is possible by the use of Matlab Compiler, which can be used as a plug-in in Visual Studio and generates C++ code from m-functions. The functions can now be used in the similar way as in Matlab, with the limitation that arrays and matrixes used by the functions must be of the type MwArray.

A Lip Sync module is developed, based on the Visage Technologies software [4], which comprises the functionality needed for the automatic lip synchronization system. Additionally, Windows application demonstrating added functionalities is implemented.

4.1 Lip Sync module

Lip Sync module implements the lip syncing functionality, i.e. by analysing the speech signal it produces the corresponding MPEG-4 visemes synchronized to the speech. Table 7 demonstrates basic characteristics of the Lip Sync module.
Table 7: An overview of the Lip Sync module characteristics

<table>
<thead>
<tr>
<th>Mode</th>
<th>Input Audio</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Microphone</td>
<td>FBA file Rendering WAV file Overlapping</td>
</tr>
<tr>
<td>Real time</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

It has two main operational modes: *real time* and *offline*. In the real time mode, the analysis of the audio signal is fast enough, and the delay low enough to drive face animation on-the-fly; however, the quality of the analysis is slightly lower in order to achieve such a low delay. In offline mode, the analysis algorithm takes into account more information of the audio signal, resulting in higher quality of the result, but with a longer delay. This mode is suitable for off line productions where the result of lip synchronization is written into a file for later reproduction. When microphone is used as input, the synchronization mode is always real time.

There are two input modes for the audio signal: *file* or *microphone*. For file input, PCM format (16 kHz sample rate, 16 bit accuracy) is supported. In the microphone input mode, the audio signal is captured from the microphone and processed on-the-fly. Optionally, the captured audio can be written into a file. Loading and saving waves as well as reading and writing waves are implemented using routines from Microsoft Wave library. If input audio file is used, the lip sync will stop at the end of the file. However, if microphone input is used, the lip synchronization is performed until the stop function is called.

The generated visemes can be written into an MPEG-4 FBA file. Additionally, the application is capable of capturing the lip sync data in real time, as it appears, and react to it, for example by rendering an animated face on the screen.

The combination of these parameters, as it Table 7 shows, makes this module extremely versatile, so it can be used in many different ways. The most typical ways are:

- produce animation from an audio file and store it into an FBA file
- produce animation from microphone and store it into an FBA file
- produce animation from microphone and render it in real time on the screen
- produce animation from an audio file and enhance it at the same time with additional facial motions (eye blinks, head movement etc.)

Lip Sync module functions in the following way. First, the speech obtained from the microphone or audio file is segmented into frames of 256 samples. When a frame is prepared, data is stored and calculations are performed during preparation of the next frame. These calculations consist of MFCC extraction and simulation of 15 networks. The outputs are added to outputs from the previous frame. Every fourth frame, the viseme class that has the largest sum of output values from NNs is chosen as the correct one [10].

In the offline mode, 75 percent overlapping between frames is introduced. Since computational time is unlimited, more complex calculations are performed. On the other hand, in the real time mode time delay between the sound and the animated lip movements plays significant role. The total time delay consists of the time needed to perform calculations and the time length of the frames taken into account for viseme determination. As we have 16 kHz sampled frames of 256 samples, the time needed for playing one frame is 16 ms. Consequently, a calculation time must be less than 16 ms. In the real time mode, when overlapping is not used, additional time delay is 64 ms, since we analyse 4 frames before deciding about correct viseme. That makes a total time delay less than 80 ms, what is short enough not to lose a real time impression.

\[
\begin{align*}
 t_{\text{delay}} &= t_{\text{calc}} + 4 \cdot t_{\text{frame}} \\
 t_{\text{calc}} &< t_{\text{frame}} \\
 t_{\text{frame}} &= \frac{n_{\text{samp}}}{f} \\
 n_{\text{samp}} &= 256 \\
 f &= 16 \text{ kHz} \\
 t_{\text{frame}} &= 16 \text{ ms} \\
 t_{\text{delay}} &< t_{\text{frame}} + 4 \cdot t_{\text{frame}} \\
 t_{\text{delay}} &< 80 \text{ ms} \\
 t_{\text{delay}} &- \text{ total time delay} \\
 t_{\text{calc}} &- \text{ time needed for calculations} \\
 t_{\text{frame}} &- \text{ time length of one frame} \\
 n_{\text{samp}} &- \text{ number of samples in the frame} \\
 f &- \text{ frame rate}
\end{align*}
\]
4.2 Demo Application

This example application demonstrates capabilities of Lip Sync module, as follows:

- generating face animation (lip synchronization) from only auditory input speech signal, coming from a file or from a microphone
- real-time or offline lip sync
- on-the-fly animation rendering or writing to a file (fba)

Weather the speech is coming from pre-recorded audio files or the microphone, spectral analysis of the speech is continuously performed. Correct visemes are chosen by neural networks at every frame. Network's biases and weights, extracted and saved in the training process, are loaded in the application together with Fisher matrix. So, for this demo to run properly, following files are needed: \textit{WeightsBiases.mat} and \textit{FisherMatrix.mat}.

To use demo application, animatable face model must be present as well as a base animation track \textit{headmotion.fba}. In general, the code works as follows. A base animation track is loaded and played by a FAPlayer. This track contains just simple head motion. In case that we want to render animation on the screen, we add the visemes generated by the Lip Sync module. To achieve better results, appropriate interpolation between visemes is performed, visemes are set at every frame of animation and animated face is rendered. The coarticulation model is a simple one using linear interpolation between visemes.

Figure 16 demonstrates overview of the demo lip sync application. \textit{Load Face Model} loads animatable face model. Choosing \textit{Play} starts lip synchronization from the audio file. In a dialog that appears user should choose between the real time and offline mode. Also, generated animation can be rendered on the screen, saved in the fba file, or both, depending on the choice. \textit{Record} starts lip synchronization from the microphone. It is always performed in the real time mode. In a dialog that appears user should choose what to do with the generated animation (render it on the screen, save it in the fba file, or both). \textit{Stop} stops lip syncing operation when microphone is used.
Automatic Lip Synchronization by Speech Signal Analysis

Possible application progress
1 - 2 - 3 - 5 exor 6 - 7 or 8
1 - 2 - 4 - 6 - 7 or 8 - 9

Figure 16: Overview of the demo application

Screenshots in Figure 17 and Figure 18 show GUI (Graphical User Interface) of demo application.

Figure 17: Menu in lip sync from audio file mode
Figure 18: GUI of our application
5 System validation

As a result of this work, Lip Sync module is designed. Due to its simplicity of use, Lip Sync module is easy to integrate with the variety of applications. In our demo application, the lips of the animated face model are synchronized with incoming speech. The source of the speech can be audio file or microphone, while the lip synchronization process is performed in the real time or offline mode, as described before. Although some time delay between the sound and the animated lip movements in the real time mode exists, the achieved visual impression is satisfactory. The accuracy of the recognition is also considered to be good.

The results obtained in the different testing conditions, will be demonstrated later in this chapter, but first, testing methods used are described.

5.1 Testing Methods

A validation of our lip synchronization system consists of two parts (Figure 19).

![Diagram showing testing methods](image)

Figure 19: Methods used for validation of the lip sync system

The first one is performed in the process of neural networks generation in order to obtain validation results of the classification with the NNs. In this part, two methods were used. The first one is based on the functions available in Matlab, while the
second one compares results of viseme recognition obtained with our Lip Sync module with the ground truth, as it will be described later.

Once the suitable neural networks are found, the effects of various factors, such as background noise, language or personality are tested on our lip sync system. In this second part we have used a subjective test, as it is based on the opinions of different persons on videos made from the animations generated in different conditions.

5.1.1 Neural network simulation

To perform some analysis of the network response, the generated neural networks are simulated with the Matlab function `sim`. The function `sim` takes the network input, and the network object, and returns the network output [44]. The network object is the neural network we want to analyse and the network input is the data we put through the network to get output. Since our database consists of recordings made by nine persons where seven of them are used in the training process, the rest of recordings are used as validation data.

The following piece of code simulates neural network `net_1`, trained for recognizing visemes in the first class, with `sim` to see if it can separate first group from the rest:

```matlab
% Neural network simulation

% Simulate the network with 'sim' function
e_0 = sum(sim(net_1,cep_coeff_0_valid))/length(cep_coeff_0_valid);
e_1 = sum(sim(net_1,cep_coeff_1_valid))/length(cep_coeff_1_valid);
e_2 = sum(sim(net_1,cep_coeff_2_valid))/length(cep_coeff_2_valid);
e_4 = sum(sim(net_1,cep_coeff_4_valid))/length(cep_coeff_4_valid);
e_5 = sum(sim(net_1,cep_coeff_5_valid))/length(cep_coeff_5_valid);
e_6 = sum(sim(net_1,cep_coeff_6_valid))/length(cep_coeff_6_valid);
e_7 = sum(sim(net_1,cep_coeff_7_valid))/length(cep_coeff_7_valid);
e_8 = sum(sim(net_1,cep_coeff_8_valid))/length(cep_coeff_8_valid);
e_9 = sum(sim(net_1,cep_coeff_9_valid))/length(cep_coeff_9_valid);
e_10 = sum(sim(net_1,cep_coeff_10_valid))/length(cep_coeff_10_valid);
e_11 = sum(sim(net_1,cep_coeff_11_valid))/length(cep_coeff_11_valid);
e_12 = sum(sim(net_1,cep_coeff_12_valid))/length(cep_coeff_12_valid);
e_13 = sum(sim(net_1,cep_coeff_13_valid))/length(cep_coeff_13_valid);
e_14 = sum(sim(net_1,cep_coeff_14_valid))/length(cep_coeff_14_valid);
res_v = [e_0, e_1, e_2, e_4, e_5, e_6, e_7, e_8, e_9, e_10, e_11, e_12, e_13, e_14];
e_p = round((res_v(2) / sum(res_v))*100);
```
The neural network net_1 is not just simulated with the data from the first group, but the calculation is repeated with the whole set of validation data. Outputs of every viseme group are summed and validation results in percent of the neural networks calculated. If the correct viseme group is recognized with 80% accuracy or more, the evaluated neural network was considered trained enough and no more effort was put to make results better. Otherwise, the new network with changed parameters is created, in order to obtain better result.

5.1.2 Lip Sync Test Application

Lip Sync Test Application is developed in order to evaluate generated NNs in more natural environment. In this method, the whole sentence is used as testing data and
not just isolated phonemes as in NN simulation method. The idea is to compare the visemes calculated by the Lip Sync module with the ground truth data. The comparison is made by the simple algorithm that had to be developed for that purpose. Figure 20 shows a basic idea of the Lip Sync Test Application.

Having familiar text, together with the basic knowledge of acoustic shapes and co-articulation principles, it is possible to segment acoustic signal into components in which sequence of phonemes can be recognized [37]. However, some phonemes have clear borders, but there are some which can be segmented only roughly and there are no generally applicable rules. Therefore, it is hard to segment correctly some acoustic signal into phonemes and the accuracy of the segmentation extremely affects results of this testing method.

Figure 21: Wave and spectral form of the ground truth sentence
As a ground truth we use a single Croatian sentence recorded by the microphone in the silent room by the person which did not participate in the database creation (Figure 21). The sentence is then segmented into phonemes by manually marking the timestamps for the beginning and the end of every phoneme. Each phoneme is represented with its associate viseme for later comparison. The following sentence is used (translated on English – *I am Reana, the first virtual person to talk Croatian*):

Ja sam Reana, prva virtualna osoba koja govori hrvatski jezik.

For the same sentence we use our Lip Sync module to calculate visemes. The visemes are calculated for four consecutive frames meaning that one viseme contains 1024 samples.

Next some alignment must be performed since ground truth viseme and calculated one differ in the number of samples it encloses as well as the timestamps for the beginning and the end. Therefore we have developed algorithm that normalizes ground truth visemes on calculated ones.

The pseudo code of the algorithm is as follows:

1. for every viseme calculated by lip sync module
2. find beginning sample - lcalcv
3. find ending sample - rcalcv
4. go through all ground truth visemes
5. find beginning sample - lgtv
6. find ending sample - rgtv
7. if ground truth viseme can be found for place of calculated viseme bounded with the beginning and the end sample
   // if (\((\text{lcalcv} \geq \text{lgtv}) \&\& (\text{rcalcv} \leq \text{rgtv})\) ||
   // \((\text{lcalcv} \leq \text{lgtv}) \&\& (\text{rcalcv} \geq \text{rgtv})\) ||
   // \((\text{lcalcv} > \text{lgtv}) \&\& (\text{rcalcv} > \text{rgtv}) \&\&
   // 2*(\text{rgtv}-\text{lcalcv})>(\text{rcalcv}-\text{lcalcv})\) ||
   // \((\text{lcalcv} < \text{lgtv}) \&\& (\text{rcalcv} < \text{rgtv}) \&\&
   // 2*(\text{rcalcv}-\text{lgtv})>(\text{rcalcv}-\text{lcalcv}))
8. than ground truth viseme is associated to that place of calculated viseme in new array of ground truth visemes
9. until last ground truth viseme is reached or wanted ground truth viseme is found
10. if place of calculated viseme can not be filled with satisfactory ground truth viseme that place in new array of ground truth visemes is marked as questionable

Applying algorithm on the original array of the ground truth visemes, the new array of the ground truth visemes is created. The contest remained the same, but being divided into 1024 samples long blocks, the organization changed to fit the one of the calculated visemes. In doing so, the number of visemes in both cases becomes equal and the comparison simple.

In order to associate some ground truth viseme with the certain block, it must contain more than a half of its length, in our case 512 samples. If neither of the ground truth visemes satisfies that condition, that block is marked as questionable and is not taken into account when testing.

5.1.3 Subjective testing using video generated from animation

Numerical results provide very good base for the validation of the system. However, a visual impression still remains important indicator of our lip synchronization system quality, since the observers are ones to whom our virtual characters talk. Therefore, a subjective test is conducted, similar as in [47].

We synthesized a facial animation of the face model using our Lip Sync module, i.e. demo application. The input is audio file and the output animation file. Both files are than used to create videos using 3ds max [38] software. Several videos are generated using different audio files and presented to 25 subjects.

Audio files have been created in order to differ in the characteristics that can influence the speech perception and recognition, such as background noise, language etc.
Table 8: Questions in subjective test Questionnaire

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 – To what extent do the generated lip movements follow the speech?</td>
<td></td>
</tr>
<tr>
<td>Q2 – Did you notice any disturbing or discontinuous lip movements during the playback of the video?</td>
<td></td>
</tr>
<tr>
<td>Q3 – In short, describe your general impression of the videos you have just seen (with particular focus on achieved naturalness, quality and accuracy of animation…).</td>
<td></td>
</tr>
</tbody>
</table>

For every video, subjects are asked to answer previously prepared questions (Table 8). In questions 1 and 2, the score was graded on a scale of 5 to 1. The higher score correspond to more positive answers. Question 3 is about general impression of all seen videos.

5.2 Testing results

In this chapter results obtained during the process of optimising the outputs of the neural networks are given. At the end, the final system is tested including different parameters and the achieved results are presented.

In order to compare results, we are starting with the original database and initial neural network configuration. Next, changes in the network design as described before are made. Once the most appropriate networks are found, we test generated neural networks with the expanded database and present detailed validation results together with the ones obtained in different testing conditions.

5.2.1 Validation of recognition with neural networks

In the first phase, i.e. while searching the suitable neural networks, we used neural network simulation method for validation of the obtained results. Original database, the one without Croatian phonemes, is used in this phase. Validations results are
expressed by the percent the neural networks recognize specific viseme class. Ideally, neural network trained to recognize viseme class 0, would recognize it with 100% probability and others with 0%. Results shown in the tables are based on the evaluation of the four frames at the time and with overlapping used.

**Testing with original database**

The recognition in the initial lip sync system is phoneme based and thereby it is done by 38 neural networks, as the phonemes are classified first and then represented with the visemes. Table 9 shows validation results of the recognition with neural networks.

Table 9: Validation results of the initial system

<table>
<thead>
<tr>
<th>Exp. Cl.</th>
<th>Recognized Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>71 7 1 - 4 1 1 0 0 0 1 8 1 0 4</td>
</tr>
<tr>
<td>1</td>
<td>9 53 8 - 6 9 4 0 6 1 0 0 1 0 3</td>
</tr>
<tr>
<td>2</td>
<td>10 10 45 - 4 11 7 0 0 4 0 0 2 2 3</td>
</tr>
<tr>
<td>3</td>
<td>- - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>4</td>
<td>5 12 7 - 61 4 5 2 0 0 1 0 0 2 0</td>
</tr>
<tr>
<td>5</td>
<td>1 10 3 - 13 44 6 1 5 2 4 3 3 2</td>
</tr>
<tr>
<td>6</td>
<td>0 5 4 - 5 12 55 2 2 1 2 2 2 3 5</td>
</tr>
<tr>
<td>7</td>
<td>1 0 5 - 13 3 1 70 0 0 0 0 7 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 4 5 - 5 2 0 0 46 1 0 1 1 4 3 19</td>
</tr>
<tr>
<td>9</td>
<td>2 6 6 - 3 17 0 1 8 36 1 6 4 0 10</td>
</tr>
<tr>
<td>10</td>
<td>0 0 0 - 0 0 0 0 0 0 81 4 0 13 2</td>
</tr>
<tr>
<td>11</td>
<td>0 1 1 - 1 4 1 0 0 1 79 4 1 3</td>
</tr>
<tr>
<td>12</td>
<td>1 3 4 - 9 6 2 2 3 1 1 11 42 6 11</td>
</tr>
<tr>
<td>13</td>
<td>0 0 1 - 0 1 0 0 0 7 1 1 85 5</td>
</tr>
<tr>
<td>14</td>
<td>0 4 1 - 2 3 1 0 4 2 2 10 9 2 62</td>
</tr>
</tbody>
</table>

Table 10 contains results obtained in viseme based NN training with 15 neural networks configured manually. In order to have results good enough the training process for most of the viseme classes is repeated many times with different configuration parameters.
Table 10: Validation results of viseme based NN training

<table>
<thead>
<tr>
<th>Exp. Cl.</th>
<th>Recognized Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>91 2 4 1 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 41 8 11 5 0 22 5 3 5 0 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>6 9 59 6 6 9 1 1 1 0 0 0 1 0 0</td>
</tr>
<tr>
<td>3</td>
<td>- - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>4</td>
<td>13 15 10 - 49 4 1 3 1 1 0 0 1 1 0</td>
</tr>
<tr>
<td>5</td>
<td>1 13 7 18 28 9 1 2 17 0 1 2 1 1</td>
</tr>
<tr>
<td>6</td>
<td>0 0 2 - 2 4 91 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>7</td>
<td>0 0 4 - 10 2 2 82 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 10 4 - 2 2 0 44 0 0 0 20 2 16</td>
</tr>
<tr>
<td>9</td>
<td>0 4 6 - 1 9 0 0 14 48 1 2 7 0 7</td>
</tr>
<tr>
<td>10</td>
<td>0 1 1 - 2 1 6 1 0 1 67 7 0 7 5</td>
</tr>
<tr>
<td>11</td>
<td>0 0 0 - 0 4 0 0 2 0 4 73 10 2 4</td>
</tr>
<tr>
<td>12</td>
<td>0 0 10 - 12 2 0 0 2 0 0 4 61 2 8</td>
</tr>
<tr>
<td>13</td>
<td>0 1 1 - 1 3 0 0 3 0 7 9 1 70 4</td>
</tr>
<tr>
<td>14</td>
<td>0 8 2 - 1 4 0 0 5 10 1 16 21 0 31</td>
</tr>
</tbody>
</table>

The final configuration for each of 15 neural networks is set up by genetic algorithms. Table 11 shows achieved results.

Table 11: Validation results obtained by neural networks configured with genetic algorithms

<table>
<thead>
<tr>
<th>Exp. Cl.</th>
<th>Recognized Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>94 1 1 - 1 2 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>2 67 8 - 8 4 4 0 2 6 0 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>3 7 56 - 5 9 8 3 2 5 0 0 0 1 0</td>
</tr>
<tr>
<td>3</td>
<td>- - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>4</td>
<td>8 11 10 - 58 5 1 3 1 1 0 1 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 12 3 - 21 34 9 2 3 10 0 3 1 0 2</td>
</tr>
<tr>
<td>6</td>
<td>0 0 2 - 2 2 89 0 0 4 0 0 0 0 0</td>
</tr>
<tr>
<td>7</td>
<td>0 0 4 - 7 2 1 86 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 8 5 - 3 5 0 0 57 3 0 0 6 0 14</td>
</tr>
<tr>
<td>9</td>
<td>0 3 3 - 1 18 0 0 4 67 0 1 1 0 1</td>
</tr>
<tr>
<td>10</td>
<td>0 0 0 - 0 0 0 0 0 96 0 0 2 2 0</td>
</tr>
</tbody>
</table>

The final configuration for each of 15 neural networks is set up by genetic algorithms. Table 11 shows achieved results.

Table 11: Validation results obtained by neural networks configured with genetic algorithms
Testing with extended database

Having neural networks configured optimally on the initial database, the training is repeated but with the extended database. For the networks whose output differed a lot from the output obtained in the training with original database, neural network parameters were fine tuned and training repeated until satisfactory results were obtained. The final configuration of all neural networks can be seen in Table 12 (all neural networks are two layer networks).

Table 12: Characteristics and result of final neural networks

<table>
<thead>
<tr>
<th>Viseme class</th>
<th>Number of hidden nodes</th>
<th>Number of training iterations</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18</td>
<td>32</td>
<td>90</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>32</td>
<td>63</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>26</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>39</td>
<td>68</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>29</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>25</td>
<td>86</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>27</td>
<td>86</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>33</td>
<td>54</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>24</td>
<td>68</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>25</td>
<td>94</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>27</td>
<td>80</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>26</td>
<td>65</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>27</td>
<td>77</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>33</td>
<td>45</td>
</tr>
</tbody>
</table>
Generated neural networks are validated using simulation and Lip Sync Test Application. The following results are obtained:

Simulation of neural networks gave results as Table 13 shows (overlapping was used).

Table 13: Validation results obtained by neural networks configured with genetic algorithms and with extended database used

<table>
<thead>
<tr>
<th>Exp. Cl.</th>
<th>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>90 1 3 - 1 1 2 0 0 2 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>2 63 6 - 8 3 15 0 2 2 0 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>6 5 57 - 4 10 6 1 1 4 0 0 4 1 1</td>
</tr>
<tr>
<td>3</td>
<td>- - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>4</td>
<td>5 6 8 - 69 5 0 4 0 0 0 0 0 3 0</td>
</tr>
<tr>
<td>5</td>
<td>1 9 7 21 37 6 1 4 11 0 1 2 0 1</td>
</tr>
<tr>
<td>6</td>
<td>0 1 1 - 1 5 86 0 0 4 1 0 1 0 0</td>
</tr>
<tr>
<td>7</td>
<td>0 0 4 - 7 2 1 86 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 4 6 - 4 2 0 0 55 2 2 0 9 0 17</td>
</tr>
<tr>
<td>9</td>
<td>0 1 3 - 1 6 0 0 5 68 1 6 3 0 5</td>
</tr>
<tr>
<td>10</td>
<td>0 0 0 - 0 0 0 0 0 0 94 2 0 5 0</td>
</tr>
<tr>
<td>11</td>
<td>0 0 0 - 1 3 0 0 0 3 0 81 6 0 0 0</td>
</tr>
<tr>
<td>12</td>
<td>0 0 6 - 11 2 0 0 2 0 0 6 64 0 9</td>
</tr>
<tr>
<td>13</td>
<td>0 0 3 - 2 0 0 0 2 0 8 1 1 77 7</td>
</tr>
<tr>
<td>14</td>
<td>0 2 2 - 1 1 1 0 3 12 1 13 20 0 46</td>
</tr>
</tbody>
</table>

The output of the Lip Sync Test Application is the rate of the correct visemes and the total number of them in the ground truth sentence calculated by the Lip Sync module. Total number of visemes in our ground truth sentence is 71. Our test application marked seven of them as questionable, following the rules described in the chapter 5.1.2. In the case when the online mode was used 26 visemes out of 64 were recognized correctly, while in the offline mode 29 of them.

Besides overall correctness of recognized phonemes, correctness per viseme class is measured. Figure 22 shows results obtained in the offline and online mode in the form of percentage of correctly recognized visemes, while Figure 23 shows results as...
number of correctly recognized visemes compared with the total number of visemes that appeared in certain viseme class.

Figure 22: Percentage of correctly recognized visemes in the online and offline mode obtained by Lip Sync Test Application

Figure 23: Number of correctly recognized visemes in the online and offline mode compared with total number of visemes that appear in the ground truth sentence
Ground truth sentence did not contain visemes from all viseme groups. Those groups are viseme classes three, six and eight (on the graph displayed with zero – actually not displayed). The third column on the Figure 23 (labeled total) denote the total number of visemes for each viseme group that appeared in the ground truth sentence. Every viseme group is represented with different number of samples. The choice was accidental.

5.2.2 Testing results in different conditions

As the final step in our system validation, a subjective test is conducted. The goal was to test our system in different conditions, with emphasis on different languages. We created eight videos from eight different audio and animation files. The characteristics of them are following:

1. **Croatian 1** – a short sentence in Croatian recorded in the silent room and the animation created in the offline mode.
2. **Croatian 2** – a short sentence in Croatian recorded in the silent room and the animation created in the online mode.
3. **English** – a short sentence in English recorded in the silent room and the animation created in the offline mode.
4. **German** – a short sentence in German recorded in the silent room and the animation created in the online mode.
5. **Swedish** – a short sentence in Swedish recorded in the silent room and the animation created in the offline mode.
6. **Mobile** – a short sentence in Croatian recorded on the mobile phone, later transformed in the suitable format and animation created in the offline mode.
7. **Noisy** – a speech in Croatian recorded in the noisy room (telephone rings, other people talking, music…) and animation created in the online mode.
8. **Lab** – a speech in Croatian recorded in the professional studio with the top quality equipment and the animation created in the offline mode.
Next, 25 subjects, generally not connected with the topic, were asked to fill in the following questionnaire (Table 14):

Table 14: Answering form for subjective test

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Croatian 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Croatian 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 English</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 German</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Swedish</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Mobile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Noisy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Lab</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Q1, Q2 and Q3 denote the questions 1, 2 and 3 shown in the Table 8. Note that higher scores for Q1 and Q2 correspond to more positive answer (score was graded on scale of 5 to 1). Figure 24 summarizes the average score for the first two questions.

Figure 24: Scores achieved for the first and the second question in the subjective testing
Question 3 contains a general impression of all seen videos. Some interesting remarks that come out of answers on the Q3 can be seen in the Table 15:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The lips sometime move although they are not supposed to move, because there is a noise in the audio file (like convulsion).</td>
</tr>
<tr>
<td>2</td>
<td>If there is only short break between the words (or before new sentence), the lips does not stop moving.</td>
</tr>
<tr>
<td>3</td>
<td>It is noticeable that some phonemes are not correctly animated either because they are not correctly recognized or because the form of the animated lips does not seem to perfectly follow the form of a real speaker.</td>
</tr>
<tr>
<td>4</td>
<td>Animation accurately follows the speech in the sense of timing.</td>
</tr>
<tr>
<td>5</td>
<td>Generally, if subjects didn't know the language spoken in the video, animation seemed to them more accurate and natural.</td>
</tr>
<tr>
<td>6</td>
<td>On the face only lips were animated. Adding some other movements (eyebrows, head or eye blinking) would improve a lot general impression no matter of the lip synchronization accuracy.</td>
</tr>
</tbody>
</table>

### 5.3 Discussion

In the previous chapter, we presented various results from different stages of our work. What follows is concise comment on them.

Results obtained with NN simulation method for the final neural networks are far away from the ideal one. The range of percentage of recognition varies from 37 (viseme class 5) to 94 (viseme class 10), with average of 70 percent. As mentioned before, we would be satisfied with 80 and more accuracy, what is not the case with most of the classes. Generally, the improvement of recognition from the initial system (Table 9) to the final one (Table 13) is visible in almost all viseme classes. Only in classes 5, 13 and 14 better results did not achieved.

It is interesting to compare the results of NN simulation and Lip Sync Test method. For some viseme classes the Lip Sync Test results are as expected considering the ones of NN simulation method, while for others they are completely opposing. For
example, viseme class 0 (silence) is very good recognized no matter of the method – 90 percent with NN simulation and 80 and 100 percent (10 total appearances) with Lip Sync Test Application in online and offline mode respectively. On the other hand, viseme class 10 achieved very good results in NN simulation (94 percent) while in our testing application the results came out to be poor - only 9.1 percent (1 out of 11 visemes) in both modes. This can be explained as consequence of not precise segmentation, i.e. the border between phonemes was not determined accurate, either because of human mistake or because there is not really clear border. Another problem of our Lip Sync Test application is short ground truth sentence. The number of visemes in it is too small for any general conclusion. Moreover the distribution of visemes is not uniformly distributed over viseme groups (some viseme groups do not appear at all in the ground truth sentence).

The serious shortcoming of our system is low recognition of viseme class 1, containing phonemes /m, /p and /b (63 percent with NN simulation and zero – 0/3 with Lip Sync Test). As mentioned before, this class is important since the lips are closed while pronouncing those phonemes. Having wrong recognition and with opening of the lips, the synchronization is definitely disrupted. Even one tested subject mentioned in the answer three in the subjective test that some phonemes are not correctly recognized, specifying phoneme /m as example.

However, the results of subjective testing proved that visual impression is satisfactory. According to achieved results, subjects generally positively evaluated the generated animations. Average grade for the first question, related with the connectivity of created animation with the original speech, is 3.69 and for the second question, more concerned with sudden disturbing movements, 3.86 (with maximum grade 5). Three out of eight videos for the subjective test are generated in the online mode and the rest in the offline. Animations generated in the real time were graded with scores, high almost as for offline animation. Average grade for the online mode is 3.72, and for the offline mode 3.81. In the real time animations, it is visible in some videos that speech precedes the animation.

Having eight videos for testing we wanted to include as much as possible recording conditions into testing. But this task is only partially fulfilled since we did not take
into consideration all possible combination of parameters. Perfect test would have the same person speaking in all conditions on every tested language and several of such speakers would be needed to perform the testing. However, even with such reduced testing some general remarks can be given.

As we expected, the video generated in the laboratory conditions with professional speaker and equipment achieved the best results. As well, the results for the video in noisy environment were consistent with our expectations. Rather poor animation was created because different sounds that were present in the room were not recognized as noise. The system was trained with Swedish and Croatian phonemes so the above average results for those videos are not surprise. However, some tested subjects liked the video generated in German the best. There are several possible explanations. One is, as mentioned in the Table 15, not knowing the language. The other one, more probable, is similarity with Swedish. Video made in English did not achieve good results. Reason for that could be the characteristic of this language that words are not pronounced clear and separate from each other but with tendency of putting it all together (in contrary with German, which is more articulate). When creating animation with the audio file recorded on the mobile phone, we did not know what to expect. However, the results came out rather good. Lower quality caused by 8 bit recording available on the mobile phone did not affect much the accuracy of the animation.

Although the subjects were asked to evaluate only animation of the lips, many of them commented on the general animation of the face. It was hard to concentrate only on the lips if eyes or head do not move at all. So, many comments in Q3 were generally about unnatural face.
6 Application Scenarios

The human face is an extremely important communication channel. The face can express lots of information, such as emotions, intension or general condition of the person. In noisy environments, lip movements can compensate for a possible loss in speech signal. Moreover, the visual component of speech plays a key role for hearing impaired people.

Besides the communication functions, the human face is primary element in human recognition. Composed of a complex structure of bones and muscles, it is extremely flexible and capable for various movements and face expressions. Such anatomical complexity accompanied by human sensibility on discontinuities in simulated face movements, makes face animation one of the most difficult and challenging research areas in the computer animation.

Virtual humans are graphical simulations of real or imaginary persons capable of human-like behaviour, most importantly talking and gesturing [39]. When integrated into application, virtual human representing a real human, brings life and personality, improves realism and in general provides more natural interface. For a realistic result, lip movements must be perfectly synchronized with the audio. Other than lip sync, realistic face animation includes face expressions and emotions. However, this is not in the scope of this work.

Lip sync systems may find applications from film production and advertising to games, teleconferencing, messaging, news delivery and in advanced user interfaces (i.e. for education and commerce). Although such systems already exist, most often they do not work in the real time, what might be essential for some applications.
Typically, face animation system consist of three main parts [40] (Figure 25):

- Face model production
- Face animation production
- Specific platform delivery

Making a face model and preparing it for animation is typically time consuming, but it need to be done only once for specific usage. Face animation, in our case generated by the lip synchronization process, produces standard MPEG-4 FBA bitstreams, with bit rates that can be as low as 0.5 kbit/sec if only viseme-based speech animation is used. The delivery is based on the very small and portable Face Animation Player core which can easily be ported on top of any software environment supporting 3D graphics [40]. The player is essentially an MPEG-4 FBA decoder. When the MPEG-4 Face Animation Parameters (FAPs) are decoded, the player applies them to a face model. Due to its simplicity and low requirements, the face animation player is easy to implement on a variety of platforms (Java applet implementation, based on the Shout3D rendering engine, PC standalone version based on OpenGL, 3ds max plug-in, implementation on a Symbian platform for mobile devices – prototype ) [40]. Therefore, virtual characters can be used on different platforms and systems, such as 3D animation tools, PCs, games consoles, Web or mobile phones [40]. However, use
of the lip sync systems on the mobile devices is quite challenging concerning high requests that are put on the terminal.

What follows is description of several lip sync application scenarios. Although variety of similar approaches exists, here are given only the most representative ones, considering the way of use and creation. So, first a typical offline and real time lip sync application is described. Then, use of such applications on the mobile devices is explored, accompanied by two examples.

### 6.1 Virtual character as a web guide

Talking virtual character can be integrated into a Web site to provide services as a virtual guide. Appearing on the site, virtual character presents Web site’s information in an attractive way by walking visitors through the content. The virtual guide presents a service to the visitors giving brief explanations. A user can navigate the site and be talked to or can find out needed information from the virtual guide using interactive maps or get help while filling in a form.

![Figure 26: A virtual human as a web guide](image)

There are three steps in creating Web content needed for virtual guide, as shown on the Figure 26.
- Making the virtual guide
- Web page creation
- Visual speech synthesis

The virtual guide’s face is animatable face model which can be driven by Facial Animation Player implemented as Java applet [46]. Typically, the guide needs to be prepared only once for specific service. The second step is preparing the actual content. After the information to be presented is determined, the presentation form has to be chosen – plain descriptions, interactive graphics or forms. Depending on the chosen method, interactive graphics, forms or images must be prepared, as well as suitable speech for the virtual guide. No matter if the virtual guide is only welcoming visitors or acting in some other way, a speech for each item must be prepared. Suitable speaker is recorded reading the desired text and these recordings are used as input in the lip sync system in order to produce animation.

Since the virtual guide uses natural voice and interacts with users, Web site becomes even more pleasant and interesting, improving realism in human-machine communication.

The whole contents is generated in the offline process. For the end-user delivery, only standard Web browser is needed. Bandwidth and CPU requirements are modest which means that Web site containing the virtual guide is accessible to practically anyone who can access the Web.

6.2 Virtual presentation

Many people gather on the big fairs or shows. Presentations advertising some products or speeches on various themes are being held. In all those situations virtual characters might replace humans making them more attractive to the broad audience. Important is to create adequate talking virtual character for the assigned role which audience will like. In doing so, the things that are already said many times, could become again interesting. Additionally, real speaker do not have to stand on the stage in front of the audience, but can sit in comfortable chair somewhere in the
background, reading the prepared text and not worrying about appearance (Figure 27).

![Figure 27: A symbolic view of the virtual presenter use](image)

A virtual character is driven by speech in the real time. The speaker talks in the microphone. At the same time lip sync process is performed and the animation is projected on the big screen. Very few requirements are set for such application. Only interesting virtual character should be created, but this is on the artists and the marketing team. Considering that all is done in the real time, interaction with the audience is also possible.

### 6.3 Use of lip sync with applications for mobile devices

With the expansion of various mobile devices such as internet-enabled cellular phones or wireless handheld devices (PDAs), new applications for virtual humans’ technologies are opening. On the other hand, for the mobile device users, applications with interfaces using speech are especially important. Consequently, creation of virtual characters’ facial animation based on the speech signal analysis could become notable method in the face animation production for such applications.

Portable computing devices have limited computational and memory resources and strict power consumption constraints. So, various demands are being set on the applications for the mobile devices. Important issues to be considered when
designing such applications are: processing power, bandwidth, handoffs, battery power, network instability caused with the nature of wireless medium or mobility of the device, QoS (Quality of Service). Further challenges arise on the terminal application using 3D faces driven by speech processing algorithms for the automatic lip sync, primarily involving speech processing, graphics and animation.

Talking virtual characters, especially if driven by a voice, enable rich multimedia services on mobile platforms and at the same time bring personality and human touch into everyday use of mobile devices. Since appearance of the virtual character, along with its speech, plays considerable role, visual quality is very important for success off applications for virtual humans' technologies.

Depending whether a speech processing and viseme classification is done on the mobile terminal or on the server, we can distinguish two different types of applications using lip sync, as described in the following examples.

### 6.3.1 Personalized multimedia messaging

Messages with talking virtual character are pleasant and more persuasive then just text or graphics, so their usage in the entertainment and communication is quite desirable. Instead of sending textual messages or plain picture, rich multimedia service can be provided using lip sync system. A possible scenario is shown on the (Figure 28).

User records a voiced message on the mobile phone and sends it to the desired mobile phone user. The voiced message is received on the server side and analysed. First, speech processing and viseme classification is done and then appropriate MPEG-4 FBA bitstream is created. Contents generated in the lip sync process, together with original speech, need to be delivered to the final destination. The easiest way is to put up entire contents in the MMS (Multimedia Messaging Service), whether as video created from animation and sound stream or as animated gif in combination with the speech. In that case an only requirement for the destination user is the ability to read MMS. Another way is to send MPEG-4 FBA bitstream together with associated speech to the user. However, in this case, Face Animation
Player must be running on the destination terminal. Additionally, terminal needs to be equipped with suitable animated face model.

Figure 28: Scenario for personalized multimedia messaging on the mobile phones

The whole lip synchronization process is made on the server side in the offline mode. Therefore, there are only few requirements on the terminal side, depending of the chosen content representation. By using own voice for creating messages, widespread messaging on the mobile phones is becoming personalized, and the perception more intense.

### 6.3.2 Synthetic video conference call

An example of the rich voice communication service is videoconference. Using lip sync methods plain user-to-user and conference calls might be transformed into synthetic video conference calls (Figure 29) [41].
The users see live animated 3D faces driven by the speech processing algorithms for the automatic lip sync. Moreover, users can choose their appearance as realistic, enhanced or completely virtual and thus fully control the way they want to appear to others. This eliminates one of the big problems of classical video conferencing – the fact that users often do not want to be seen as they are and dislike having to dress up for a conference-based meeting.

Using the speech-driven animated faces enables rich multimedia service while keeping the bandwidth low (extra bandwidth approx. 2 kbit/s). Additionally, no camera is needed. However, requirements on the terminal side are significant, since, the whole lip synchronization process is made on the terminal side in the real time mode. In order not to exceed processing power of the mobile device, the lip sync system must be optimised for low-power speech signal processing and viseme classification.
Conclusion

The research area of this Master Thesis is automatic lip synchronization of synthetic 3D avatars based only on a speech input. After the investigation done on existing models, we have constructed and implemented a lip synchronization system suitable for the real time and offline applications.

Main contributions of this thesis are: the improved lip synchronization algorithm with viseme based training of neural networks, automatically configured by the genetic algorithm, the lip sync system adjusted to the characteristics of Croatian language, system validation based on three different evaluation methods carried out during several phases of development and a detailed discussion of potential applications of these technologies.

In our approach for the lip synchronization system by speech signal analysis, the speech is classified into viseme classes by neural networks. Genetic algorithm is used for obtaining a near optimal neural network topology. By introducing segmentation of a speech directly into viseme classes instead of phoneme classes, computation overhead is reduced, since only visemes are used for the animation of the face, i.e. the lips. The automatic design of neural networks with genetic algorithms saves much time in the training process. Moreover, better results are achieved than with manual search of a network configuration. However, some improvements in the algorithm itself are still possible, e.g. limiting the scope of possible solutions or introducing the parallelism - although GA saves a lot of human time, it is still time consuming for the computer.

According to the feedback that we have received from testing subjects, we can conclude that our lip sync system can be used in various applications since animations generated in different conditions are fairly convincing. Therefore, prototypes of proposed services could be implemented.
Because natural speech always involves some facial gestures, a face that only moves the lips looks extremely unnatural. Our next step will be to extend the automatic lip synchronization system with facial gestures and emotions. Therefore, following efforts will be focused on the extraction of face expressions in addition to lip movements from the speech signal. For the animation of the speech driven facial gesturing more information is needed. Driven with that fact, the speech prosody will also be taken into consideration.
References


[23] [21], Chapter 7, P.Hong, Z.Wen, T.S.Huang, "Real-Time Speech-Driven Face Animation"


[38] 3ds max, [http://www4.discreet.com/3dsmax/](http://www4.discreet.com/3dsmax/)


[41] I. S. Panžić, M. Matijašević, "FaceTalk: Rich voice communication in IMS".


Acronyms

MFCC  Mel-Frequency Cepstrum Coefficients
FLDT  Fisher Linear Discriminate Transformation
NN    Neural Network
ANN   Artificial Neural Network
TDNN  Time-delay Neural Network
GMM   Gaussian Mixture Model
VQ    Vector Quantization
HMM   Hidden Markov Model
FFT   Fast Fourier Transformation
DCT   Discrete Cosine Transformation
GA    Genetic Algorithm
EM    Expectation-Maximization
MPEG  Moving Picture Experts Group
MPEG-4 FA  MPEG-4 Facial Animation
MPEG-4 FBA  MPEG-4 Face and Body Animation
FPs  Feature Points
FAPs  Face Animation Parameters
MUs  Motion Units
MUPs  MU Parameters
LP    Linear Predictive
PCA   Principal Components Analysis
PCM   Pulse Code Modulation
VRML Virtual Reality Modeling Language
GUI   Graphical User Interface
QoS   Quality of Service
PDA   Personal Digital Assistant
MMS   Multimedia Messaging Service
## Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lip synchronization</strong></td>
<td>Lip synchronization is a method for generating an animation of 3D human face model where the animation is driven only by a speech signal.</td>
</tr>
<tr>
<td><strong>Viseme</strong></td>
<td>A viseme is the visual representation of a phoneme or unit of sound in spoken language.</td>
</tr>
<tr>
<td><strong>MPEG-4 FBA</strong></td>
<td>MPEG-4 FBA is specification concerned with the shape and animation definition of the human body and face. It is only standard for facial animation.</td>
</tr>
<tr>
<td><strong>Virtual characters</strong></td>
<td>Virtual characters or virtual humans are simulation of humans on the computer in such way that virtual model looks and acts almost like a real human.</td>
</tr>
<tr>
<td><strong>Digital speech processing</strong></td>
<td>Digital speech processing is method which involves first obtaining representation of the speech signal and then the application of some higher level transformation in order to put signal in suitable form. The last step is extraction and utilization of information needed for the specific application.</td>
</tr>
<tr>
<td><strong>Neural networks</strong></td>
<td>Neural networks are simulation of the information processing capabilities of nervous systems. Just as humans apply knowledge gathered from the past experience to new problems, a neural network uses previously solved examples to build system capable for solving new problems.</td>
</tr>
<tr>
<td><strong>Genetic algorithms</strong></td>
<td>Genetic algorithms are a method for solving optimisation or search problems inspired by biological processes of inheritance, mutation, natural selection and genetic crossover.</td>
</tr>
</tbody>
</table>
Summary

This master thesis investigates automatic lip synchronization. It is a method for generating an animation of 3D human face model where the animation is driven only by a speech signal. The whole process is completely automatic and starts from the speech signal. The automatic lip synchronization consists of two main parts: audio to visual mapping and a face synthesis.

The thesis proposes and implements a system for the automatic lip synchronization of synthetic 3D avatars based only on the speech input. The speech signal is classified into viseme classes using neural networks. The topology of neural networks is automatically configured using genetic algorithms. Visual representation of phonemes, viseme, defined in MPEG-4 FA, is used for face synthesis.

The system is adopted for specificity of the Croatian language. Detailed system validation based on three different evaluation methods is done and potential applications of these technologies are discussed in details. This method is suitable for real-time and offline applications. It is speaker independent and multilingual.

Keywords:
lip synchronization, facial animation, MPEG-4 FBA, virtual characters, speech processing, neural networks, genetic algorithms
Sadržaj

Ovaj magistarski rad istražuje automatsku sinkronizaciju usana. To je metoda generiranja odgovarajuće animacije 3D modela ljudskog lica s obzirom na postojeći zvučni signal govora, tako da se dobiva animirani lik koji govori. Postupak je potpuno automatiziran i počinje od zvučnog signala govora. Automatska sinkronizacija govora sastoji se od dva dijela: preslikavanja iz ulaznog audio u izlazni vizualni skup parametara i sinteze lica.

U radu je predložen i napravljen sistem za automatsku sinkronizaciju usana temeljen samo na govornom signalu. Klasifikacija govornog signala u grupe vizeme izvedena je pomoću neuronskih mreža. Optimalna topologija neuronskih mreža se automatski određuje pomoću genetičkih algoritama. Vizemi, vizualni predstavnici fonema, definirani su u MPEG-4 FA standardu, a koriste se za sintezu lica.


Ključne riječi:
sinkronizacija usana, animacija lica, MPEG-4 FBA, virtualni ljudi, obrada govora, neuralne mreže, genetički algoritmi
Biography

I was born in Zagreb, on October 16th, 1978. In 1997 I finished mathematical and natural sciences high school (V. gymnasium in Zagreb). The same year I continued my education at the Faculty of Electrical Engineering and Computing, University of Zagreb. I received my B.S. (Dipl.-Ing.) degree in electrical engineering with a major in telecommunications and information science from the University of Zagreb in May 2002. My Diploma Thesis topic was "Mobility of agents in IPv6 Network". In November 2002 I started postgraduate program at the same faculty. I have been working at the Department of Telecommunications at Faculty of Electrical Engineering and Computing as an associate assistant since December 2002. I have published five papers with international review. My current research interests are in the field of face animation of virtual characters, with particular focus on the speech driven animation. I am member of IEEE and fluent speaker of English and German.
Životopis