Master ATIAM internship report

Parametric crowd synthesis for virtual acoustic environments

Author:

Vincent Grimaldi

Supervisors:

Prof. Dr. Stefan Weinzierl - Henrik Von Coler - Christoph Böhm

Audio Communication Group - Technische Universität Berlin

February 15 to July 25, 2016

Contents

Acknowledgement

I would like to thank Prof. Dr. Stefan Weinzierl for welcoming me in the Audio Communication Group of Technische Universität Berlin and giving me the opportunity to work on this truly engaging topic.

I also want to thank Henrik Von Coler and Christoph Böhm for following my work and for the advice and help in their field of expertise, the synthesis and the auralization respectively.

I also want to thank Hannes Helmholz for his precious support and the time he took to help me set up the listening test. I also want to generally thank the whole of the Audio Communication Group for the nice spirit and any little help or advice along the way.

I want to thank all the participants of the recording session in the anechoic chamber and also those who took part in the listening tests.

Finally I would like to thank all the people involved in ATIAM and more generally in IRCAM for a really exciting year in a very stimulating environment.

Summary

The goal of this assessment is the implementation of a parametric sound texture synthesis, more specifically a crowd, in virtual acoustic environments. Granular and corpus-based concatenative synthesis were used to generate single streams of gibberish speech. Anechoic speech material was recorded and processed to compose the corpus. The resulting database was used in a real-time implementation including the rendering in a binaural virtual acoustic scene. The implementation allows to modify the density of the crowd, the level of excitement, different speech models and the position of the sources. Finally, listening tests were conducted to evaluate the synthesis and the influence of the parameters.

Keywords : crowd synthesis, sound texture, parametric, granular synthesis, corpus-based concatenative synthesis, virtual acoustics, binaural

Résumé :L'objectif de ce stage est l'implémentation d'une synthèse de son de texture paramétrable, plus précisément de foule, dans un environnement acoustique virtuel. La synthèse granulaire et la synthèse concatenative par corpus furent utilisées afin de générer des signaux de voix parlée incompréhensibles. Des enregistrements anéchoïques de voix parlée ont été réalisés et traités afin de composer le corpus. La base de donnée résultante a été utilisée pour l'implémentation en temps réel dans un espace acoustique virtuel en binaural. L'implémentation réalisée permet de modifier des paramètres tels que la densité de la foule, son degré d'exctiation, différents modèles de parole et la position des sources. Enfin, des tests d'écoute ont été menés afin d'évaluer la synthèse et l'influence de certains paramètres.

Mots-clés : synthèse de foule, sons de texture, paramétrable, syntèse granulaire, synthèse concatenative par corpus, espace virtuel acoustique, binaural

Introduction: context and purpose

Context of the internship

This work was conducted during an internship as part of the ATIAM Master 2 of IRCAM. It took place at Technische Universität Berlin, in the Audio Communication Group under the supervision of Prof. Dr. Stefan Weinzierl, Henrik Von Coler and Christoph Böhm. Research of the Audio Communication Group is dedicated to the communication of music and speech in acoustical or electro-acoustical systems. Main topics of interest of the Audio communication Group are: electro-acoustic recording and reproduction technologies, 3D audio by binaural technology or sound field synthesis, technologies for composition and realisation of electroacoustic music and sound art, and empirical approaches to study the reception of media content.

Motivation and purpose of this work

In the past, the creation of so-called soundscapes for application in virtual acoustic environments has been conducted by recording of real acoustic environments with binaural recording methods. The application of the MTB-method (Motion- Tracked Binaural Sound) [1] provides the opportunity to achieve head related recordings for later use in dynamic binaural synthesis. The main drawback of this method is that the recordings always have to be done in a setting similar to the desired auralized environment. Moreover the content of the recording is permanent and cannot be subsequently changed. An ideal application that calls for the need of sound texture synthesis are virtual acoustic environments. Indeed, soundscapes in those environments are playing an important role in providing an immersive and realistic experience to the user. Moreover, in interactive environments such as computer games, the user might stay in the same place for an undetermined amount of time. The solution commonly used by sound designers would be to prepare recordings for seamless looping. Nevertheless, this is not a flexible solution. Furthermore it requires hours of recordings to provide a decent variety of soundscapes and extensive data storage in order to avoid the loop to be distinguished and lead to boring situations for the user. Sound texture synthesis offers a more dynamic and flexible solution. The aim is here to be able to synthesize in real-time those sound textures from a reduced amount of data. The challenge is also to give the sound designer the ability to modify parameters, if possible, in real-time. For instance, the mood or excitement of a crowd, the density of rain or the strength of the wind could be modified along with the action.

The goal of this work is to explore a solution for parametric real-time sound texture synthesis for virtual acoustic environments. Due to other works on virtual acoustic environments being carried out in parallel at Technische Universität Berlin including a virtual scene with people, it was considered interesting to focus on crowd synthesis.

1

State of the art

The aim of this part is the definition of sound texture and summary of the existing methods for its synthesis. The existing methods for generating crowd synthesis will also be described. Finally the principles of binaural and virtual acoustics will be briefly explained.

1.1 Sound texture

Defining sound texture can be a tricky task. In one of the first attempts by Saint-Arnaud and Popat, 1995 [11],], it is specified that ”it should exhibit similar characteristics over time, that is, a two-second snippet of a texture should not differ significantly from another two-second snippet.” As well as being correlated to a wall-paper in that: ”it can have local structure and randomness, but the characteristics of the fine structure must remain constant on the large scale.”

In Schwarz, 2011 [2], the following interesting properties are described, also based on [11] and [12]: Sound textures are formed of basic sound elements, often called atoms, that follow a high level pattern which can be either periodic, random or both. The high-level characteristics must remain the same over long time periods. The high-level pattern must be completely exposed within a few seconds. High-level randomness can be used, as long as there are enough occurrences within those first few seconds. This is visualized on Figure 1.1.

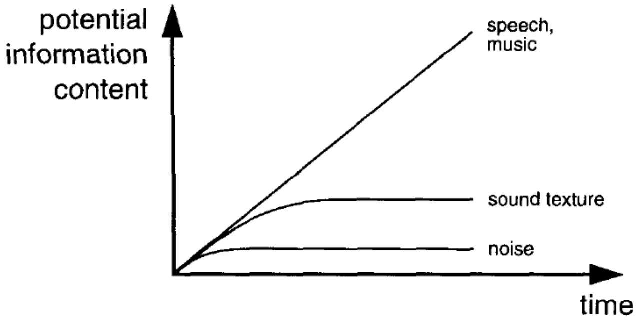


Figure 1.1: Potential information content for speech, music, noise and sound texture, from [11]

Typical sound texture examples from this definition are rain, crowds, water streams, wind or fire.

1.2 Synthesis of sound texture

An overview of the existing methods for generating texture sound synthesis will be listed here. Those were summarized in Schwarz, 2011 [2]. Both granular and concatenative synthesis which were used for this project will be described in further detail .

1.2.1 Granular synthesis

Granular synthesis theory was first developed by Gabor in 1947 in the context of microsound study. Afterwards, it was used in a musical context for producing complex sounds by Iannis Xenakis in 1958 with the piece Concrete Ph and by Curtis Roads who implemented the first real-time granular synthesis on computer in 1978. It is a time domain technique based on the construction of signals by combining very short sounds called grains. Those grains can be considered as a kind of atomic sound particles. Their combination is used to generate new sounds, timbres, or sonic textures. Human perception becomes ineffective in recognizing pitch and amplitude when sonic events are under a threshold of 50 milliseconds (Whitfield, 1978 as cited in Fischman [22]). Therefore, typical durations of grains are usually chosen to be between 10 to 60 milliseconds.

The grain is composed of two elements: the content and the envelope. Only one grain itself has a very short duration and as a single entity has no particular sonic interest, but can lead to interesting results when combined with other grains. Any type of sound-wave can be used as the content of the grain. The chart on Figure 1.2 sums up the principle of a simple grain generator.

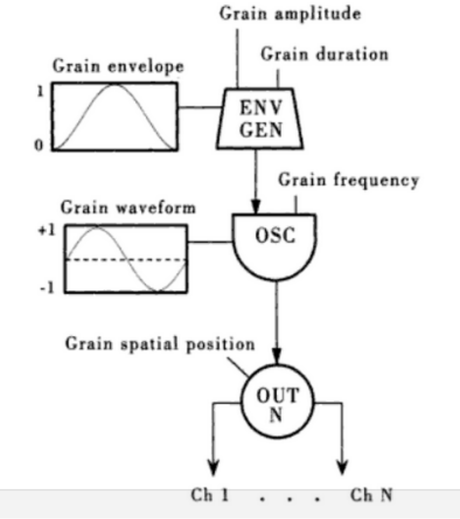


Figure 1.2: Chart for a simple grain generator with a Gaussian envelope and output on N channels, from Roads [23]

The envelope determines the duration and the amplitude of the grain. Many types of envelopes can be chosen, but the most commonly used are: Gaussian, Hanning or trapezoid.

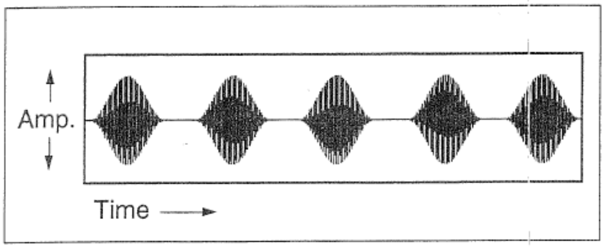


Figure 1.3: A stream of five grains and Hanning envelopes, Roads [24]

As the grain content is windowed it imposes an amplitude change over the content as was seen in Figure 1.3. The amplitude follows the shape of the envelope. The choice of the envelope is crucial for obtaining the desired sound. As an example, short linear attack and decay is an interesting choice for preventing clicks being added to the sound. Changing the form of the grain envelope changes the resulting grain spectrum, and sharper attacks produce broader bandwidths.

Different instances of time organization of the grains can be used and can lead to very different results. The principal examples are: synchronous/quasi-synchronous: variable delay between the grains, asynchronous: stochastic distribution and pitch synchronous: designed to be synchronous with the frequency of the grain waveform. More details can be found in Roads [24].

1.2.2 Corpus-based concatenative synthesis

Corpus-based concatenative synthesis could be seen as an extension of granular synthesis. The principle of concatenative corpus-based sound synthesis is to use a large database of source sounds. This material is segmented into units. Then a unit selection algorithm is used in order to generate a sequence of units to synthesize the targeted sound. Several methods can be used for the selection. Finally the units are concatenated.

Advantage of corpus-based concatenative synthesis

In the context of sound texture synthesis, corpus-based concatenative synthesis has undoubtedly attractive characteristics. Indeed, with usual sound synthesis methods based on a model of the sound signal, generating all the subtle details of the sound is typically quite intricate and a lot of information is lost. Concatenative synthesis allows to preserve those details as the units are actually extracted from real recordings containing these features. The sound of a crowd is a perfect example of this complexity.

Database

The database is built from the audio material itself, the unit segmentation and the unit descriptors. A relationship between the units can also be stored. The part of the database that is chosen for one particular synthesis is called the corpus.

Target

The target is defined according to desired descriptor characteristics. For example an angry crowd. Then, only a subset of the available database descriptors is used in order to reach this target. It is also possible to transform the selected units in order to match the target specification or extend the possibilities of the synthesis. However it is preferable to use a large database in order to do as little sound transformations as possible to avoid degrading sound quality.

Selection of unit

Accurately chosen high level descriptors and audio features allow to automatically extract characteristics from the source sounds and are a powerful tool to classify the data. It is also possible to use metadata and hand labeling as a method for more subtle or subjective classification of the corpus. The unit selection algorithm is a key element for concatenative corpus-based synthesis.

1.2.3 Other methods

Alternative methods for sound texture synthesis that were not used for this work will be briefly described. More details and substantial references can be found in Schwarz [2].

Subtractive synthesis

This method consists of filtering noise. It could be seen as the most commonly used method for sound texture synthesis. This includes techniques that rely on statistical models. Numerous methods for filtering exist and are summed up in Schwarz [2].

This method can also be completed by the use of additive synthesis, and classic sum of sinusoidal partials.

Physical modeling

Physical modeling relies on a mathematical model which can be seen as a set of equations and algorithms to simulate a physical source of sound. The synthesized waveform of the sound is generated from this model. In [2] a large set of applications of this method to sound texture are listed.

Wavelets

This method, applied to sound texture synthesis, uses multiscale decomposition of a signal into a wavelet coefficient tree. A reorganization of the order of paths from the tree structure can be done afterwards. Inverse wavelet transform is applied for those paths in order to re-synthesize a short part of signal. Kersten [5] recently applied this technique and delivered interesting results in the context of this work that will be discussed later. Several wavelet related works are also mentioned in Schwarz [2].

1.3 Crowd synthesis

For immersive virtual environments, audio is a crucial element. Its principal role is the creation of a sound ambiance or soundscape and crowds are one of the most frequent sound in this framework. [8] is an approach of a full crowd simulation, including audio. Many interesting elements to be taken into account are specified for the creation of such environment. When recreating a realistic scene, the perception of listening to a static or repetitive loop should be primarily avoided. Additionally, the quality of generated sounds should ensure that listeners recognize the nature of the sound sources. In [8], the approach chosen for audio is a sample-based concatenative synthesis, using a set of real speech recordings retrieved from *freesound.org*. This work aims to generate the necessary variability to build a realistic sound of a crowd with a very small set of recordings. The samples used here are directly the sentences themselves. In this ”talking soundscape”, two types of sound zones are defined: near-field and diffuse-field. Near-field content consists of voices from individual speakers located at a short distance from the listener. Those are spatialized accurately and voice sources are distinguishable with timbre, prosody and timing.

A drastically different approach of crowd synthesis was achieved by Kersten [5] using wavelet decomposition and hidden Markov model trees. Even though this analysis-synthesis approach works well for sounds such as water streams, rain or fire, the intelligible aspect of speech composing crowds is not successfully re-synthesized, and results in a completely unnatural crowd sound. Sound samples of those can be listened at [6].

1.4 Binaural hearing and technology

Through evolution and the need to survive in nature, Humans acquired a powerful ability to locate the sound in space. Time and level differences between both ears or spectral information are interpreted by the auditory system as cues for locating sound sources.

The goal of binaural recording and reproduction technology is to recreate this effect so that the listener can experience sound localization using headphones.

Head-related transfer function

A head-related transfer function (HRTF) describes the influence of a listeners head on the sound-field and thereby the influence on the perception of a certain sound event reaching the listeners ear drums. Diffraction and reflections on elements of the body such as the head, pinnae and shoulders are taken into account.

HRTFs are defined for each ear. They include information for magnitude and phase shift. It is a Fourier transform of a head-related impulse response (HRIR). The HRTF depends mainly on the location of the sound source compared to the position of the listener [13][14].

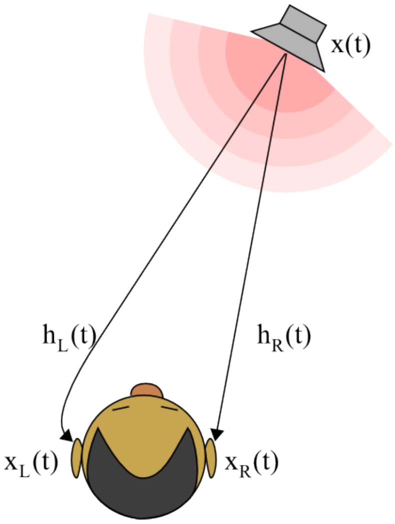


Figure 1.4: Filtering of a signal x(t) by two separate transfer functions $h_L(t)$ and $h_R(t)$ [16]

As shown on Figure 1.4, let $h_L(t)$ and $h_R(t)$ be the impulse responses in the time domain for the left and the right ear respectively. $H_L(\omega)$ and $H_R(\omega)$ in the frequency domain. $x(t)$ is the function corresponding to the pressure of the sound source and let $x_L(t)$ and $x_R(t)$ be the pressure at the left and the right ear respectively.

The pressure at the ears in the time domain as a convolution ($\circledast$ will be used) of the sound signal and the HRIR of the corresponding ear, so for example for the right ear is:

$$x_{L,R}(t) = h_{L,R}(t) \circledast x(t) = \int_{-\infty}^{+\infty} h_{L,R}(t-\tau)\ x(\tau)d\tau$$

This then leads in the frequency domain to:

$$X_{L,R}(\omega) = \mathscr{F}(h_{L,R}(t) \circledast x(t)) = H_{L,R}(\omega)X(\omega)$$

1.5 Virtual acoustic environment

Binaural room impulse response

Reflections that occur in rooms are an important element for spatial auditory perception. Using this parameter when simulating the virtual acoustic environment leads to several advantages such as the ability to recreate the spatial auditory perception in a room or perceiving the distance of the virtual source [15]. It can also prevent undesirable effects such as perceiving the source from an in-ear position. In order to take reflections into account, the method is to measure the physical propagation of sound from the source to the listener. The room and its response to an acoustical event can be seen as a linear system invariant in time which could be specified by an impulse response.

Assuming a linear system invariant in time, a room impulse response can describe the acoustic properties of a room concerning sound propagation and reflections for a specific source-microphone configuration [36]. Let $h_j(k)$ be the set of room impulse responses (where $j, .., M$ and with M the total number of microphones), and $s(k)$ the anechoic speech or audio signal, the resulting microphone signals is obtained by:

$$x_j(k) = s(k) \circledast h_j(k)$$

To obtain the impulse responses, two methods are generally used: simulation with geometrical acoustics or measurement in rooms, more details can be consulted in [28].

An impulse response characterizing a room is called a room impulse response (RIR) and can be measured with binaural technique to achieve a binaural room impulse response (BRIR) which includes also spatial information. To make the recorded data audible the BRIR can be convolved with an arbitrary stimulus.

Simulation

The simulation of sound propagation inside enclosed environments is achieved by applying methods of geometrical acoustics to imitate realistic behavior. Hybrid acoustic simulation models can be used to generate the different components of the room impulse response. Typical room impulse response is composed of the direct sound, early reflections and a reverberation tail as shown on Figure 1.5. For example, the real-time room acoustics simulation framework *RAVEN* [5] combines the Image Sources model (Allen and Berkley (1979) as mentionned in [29]) and a Ray Tracing model (Rokstad, Strom, and Sorsdal (1968) as mentionned in [29]) to compute impulse responses.

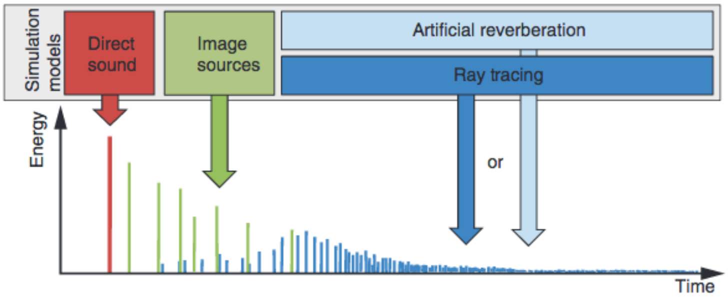


Figure 1.5: Simulation components and their contributions to the room impulse response, taken from [29]

Auralization

The generated data from the simulation is used afterwards for the auralization. Kleiner [21] defined auralization the following way in 1993: ”Auralization is the process of rendering audible, by physical or mathematical modeling, the sound field of a source in a space, in such a way as to simulate the binaural listening experience at a given position in the modeled space.” Thus, the character of sound signals which are generated at the source can be predicted and modified by reinforcement, propagation and transmission in the simulated environment. *RAVEN* or the *SoundScape Renderer* [20] can be used for this purpose.

2

Strategy and principle of implementation

2.1 Choice of concatenative corpus-based synthesis

Concatenative corpus-based synthesis has the advantage of offering a large amount of flexibility in terms of implementation, manipulation and resulting sound which makes it a particularly interesting tool for exploring the desired ”parametric” aspect of sound texture. It is also highly suitable in this context as a time-based method.

Another fundamental and interesting aspect of concatenative corpus-based synthesis is its strong ability to reproduce subtle timbre features. Indeed, the sound is actually built from the elements of the targeted material itself. This is especially interesting for crowds where analysis-synthesis methods can sometime fail to restitute the intelligible aspect of speech. An illustration of this, using hidden Markov tree model and wavelet decomposition can be heard at the bottom of the web page [6], from the PhD thesis of Kersten [5] on sound textures.

2.2 Principle

The idea is to employ concatenative granular synthesis where every unit will correspond to a speech unit or syllable. This way every single source will generate a concatenation of syllables picked in a controlled random way to generate gibberish talk. Indeed the aim is to obtain a crowd texture with no comprehensible words. This leads to the implementation of a granular synthesis where the size of the grain is the size of the syllable and where no overlap of sample should appear. Each stream can be seen as one person of the crowd corresponding to one timbre.

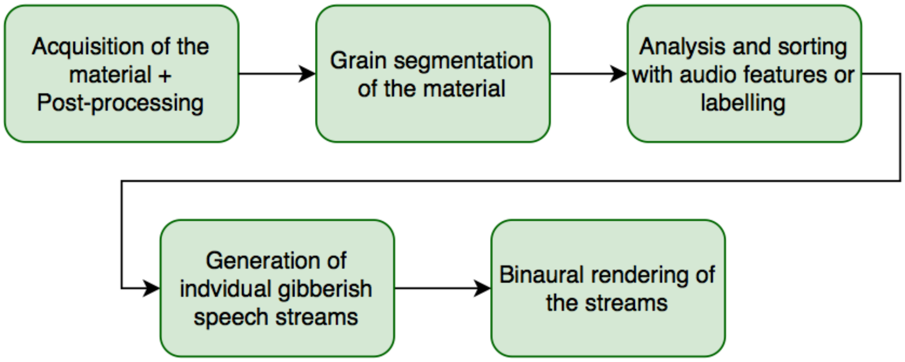

Acquisition of the material + Post-processing
Grain segmentation of the material
Analysis and sorting with audio features or labelling
Generation of indvidual gibberish speech streams
Binaural rendering of the streams

Figure 2.1: Flow chart for the crowd synthesis

Figure 2.1 sums up the principal steps of the method developed and implemented to achieve the crowd synthesis. The next sections of this report will not follow a chronological order of the work achieved, but the logical order of this chart.

3

Creation of the corpus for concatenative corpus-based crowd synthesis

3.1 Database

The material used is a crucial element for the quality of the synthesis utilizing concatenative corpus-based synthesis.

For our purpose, it is important to gather enough material to avoid distinguishable repetition during the synthesis. Different types of valence or emotion are also necessary for each speaker to create an engaging corpus to explore when selecting the elements while changing parameters of the synthesis. The recordings also need to be free of any reverberation as it will be additionally generated in the auralization, and should not contain any background sounds to allow a clean synthesis. This means that taking material from movies or public speech recordings is not suitable for our use.

After an exploration of the available voice and speech databases available on the Web, the EmoDB [9] database was chosen. It consists of recordings of ten different actors from Germany, 5 females and 5 males made in the anechoic chamber at Technische Universität Berlin. Emotional utterances were delivered from a set of 10 different sentences in German. The actors were instructed to convey the following emotions: anger, boredom, disgust, anxiety/fear, happiness, sadness. A neutral version was also recorded.

The recordings are labeled according to those emotions. It was then necessary to sort them by speaker and by emotion for the use in the implementation on *Pure Data*. One .wav file per speaker with the utterances grouped by emotion. A syllable segmentation algorithm (described further in 3.3) is applied on those categorized recordings. This way we obtain a range of syllable onsets sorted by emotion for each speaker.

This database was a perfect tool for the first implementations and testing the efficiency of corpus-based concatenative synthesis. Nevertheless it suffers some imperfections. The content is sometimes too short (only two sentences for example) for some speakers and repetitions could be heard during the listening tests in the wave-field synthesis studio. Also the data is now available at only 16 kHz sample rate which is insufficient for our needs. Finally, the fact the utterances were delivered by actors through instructions resulted in an unnatural feel. Concerning the neutral speech, a monotonous feel was perceived and the result gave the impression of a crowd reading out.

3.2 Acquisition of the material for crowd synthesis

In order to improve the result, we decided to record our own source material. Two recording sessions were planned, each with five speakers with no previous acting experience. They were students from Technische Universität, 5 males and 5 females. The recordings were made in the anechoic chamber.

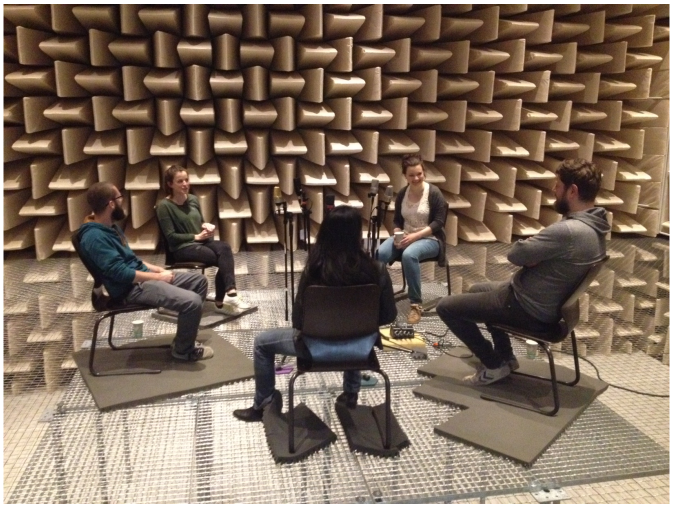


Figure 3.1: Recording of speech material in the anechoic chamber

Five Neumann TLM103 large diaphragm microphones were used. They were carefully placed at a sufficient distance in order to avoid proximity effect that comes with the cardioid polar pattern. This turned out to be a major issue on some of the recordings used from the EmoDB database. The recordings were made at a 48 kHz sample rate and 24-bit resolution. The speakers were instructed from outside the chamber with a playback system.

The choice was made to give as few instructions as possible in order to obtain a natural feel of the speech material which was one of the dominant characteristic missing on the previously used audio recordings. As a consequence of this choice, an extensive amount of post-processing time is required to select the usable material and extract unintentional artifacts. In a second phase, the speakers were asked to act out some scenes one by one to induce emotions and different speech tones. For example, acting like their bike is being stolen or calling the police after an accident.

After selection of the usable data, the recordings were satisfying and sounded more natural than the EmoDB database previously used.

However, there are several ways to improve the recordings and make them more efficient for our purpose and which should be taken in account in a similar recording situation. First, though the platform in the anechoic chamber was fixed with a lot of care, generous moves were leading to vibrations and unwanted low frequencies in the recordings. Making sure that the speakers are aware of this in a subtle way, so that they still continue to act naturally, was decided after few minutes of recording and successfully improved the quality of the data. Also many simultaneous talking caused unusable data, for example noisy echoing, giggling, or cutting off the speaker. Nevertheless, this is hardly preventable when natural behavior is desired.

3.3 Segmentation of data

The first steps of implementation were made using a simple onset detection. This was made upfront using a basic *Matlab* algorithm inspired by the first steps of a tempo detection algorithm. After this, the pseudo-random selection of grains is made within those onsets and triggered at a random pace between a maximum and minimum value determined with listening tests. However, this method is not sufficient enough as the length of the grain is not known which usually leads to gaps or abusive overlap between two consecutive grains. This is especially inconvenient for crowd synthesis, as it results in unnatural speech model.

Thus, we need to be able to detect syllables in the audio recording. For this, we used an algorithm inspired by a paper on bird species detection by A. Harma [10]. The principle is to look for maxima in the .wav file, and then check before and after the corresponding time for when the data goes below a certain threshold. Then we remove the syllable from the data and look for the next maximum. The threshold was chosen empirically after different tests for the best result. This way we keep the time of onset for each syllable and its length. This data for the real-time synthesis is then used in *Pure Data*. The implementation was made so that after the onset is randomly selected within the wanted target, the corresponding length of the syllable is used as the value of a continuously varying delay before the next unit is triggered for each source. The result is then a consecutive triggering of syllables, in order to generate gibberish speech.

In Figure 3.2, the result of the syllable segmentation for a short sentence from the speech corpus is shown, with the corresponding spectrogram. Syllables are the red segments.

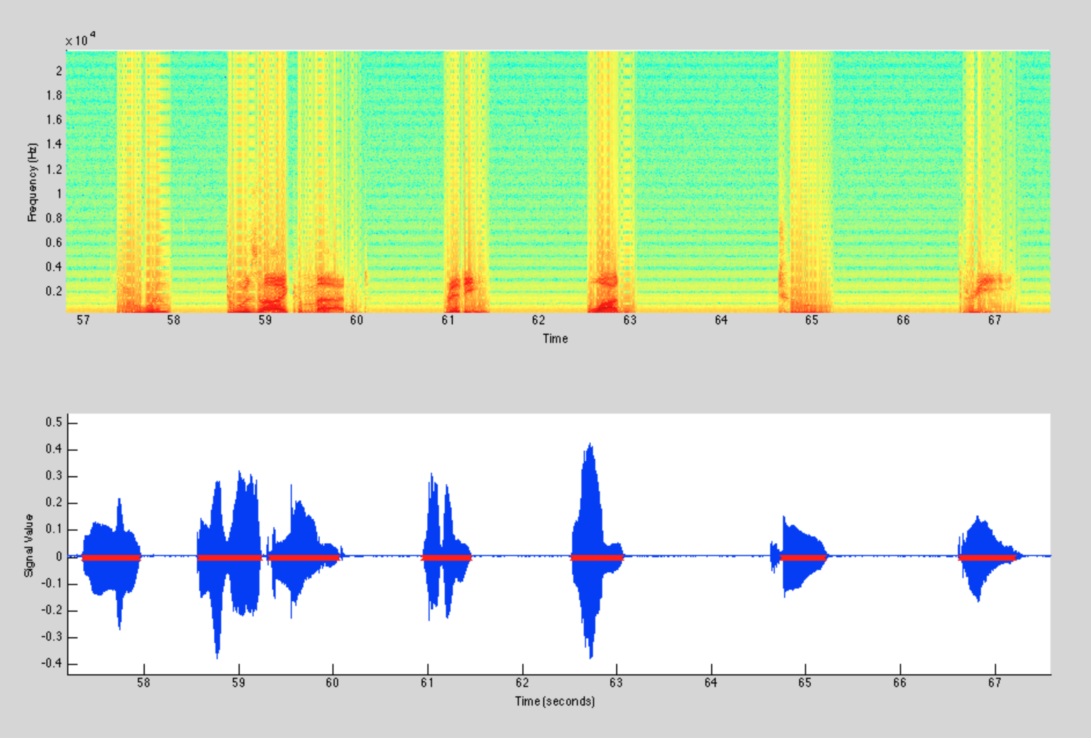


Figure 3.2: Syllable segmentation

This method offered efficient results. Nevertheless, depending on the threshold and fluidity of the speech the segmentation sometimes extracts whole words instead of only syllables. Even though this is still interesting for the synthesis quality as words are often understandable from crowds, this leads to more distinguishable repetitions when the material is not large enough. Syllables shorter than 80ms, most of the time corresponding to mouth noises, clicks or other artifacts were removed.

3.4 Classification of data

3.4.1 Label

A simple way to classify the data is to use manual labeling, consisting of simply annotating the data according to specified parameters. For example the mood of the crowd is a good application for this, as automatic detection of mood with audio features can be a complex topic. For the first version of the implementation, the data used from the EmoDB database were already including meta-data corresponding to the mood of each recording. This method is interesting when trying to capture subtle features or characteristics of the audio material that are too complex for a purely automatic method. The main drawback is the amount of time required for executing such an approach and a need for large database, but methods such as [3] exist for a semi-automatic annotation.

3.4.2 Audio features

Using audio features to detect human moods in speech such as anger, fear, happiness or sadness is a delicate task and a topic of research of its own. Nevertheless it is possible to detect simpler behaviors such as the degree of arousal or valence. According to [26], fundamental frequency and RMS energy are the most revealing audio features for this topic. The length of the syllables is a piece of data to take into account as well.

RMS energy

Measuring the RMS energy of the syllables is a basic method to assess the different level of excitement in the recorded material. Indeed, assuming that the speaker who was sat during the recording session was not changing the distance between his mouth and the microphone. RMS energy is calculated with the following formula:

$$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

A good example of the use of RMS energy in this context, as mentioned in [26]: sad speech has less median value and lower spread in RMS energy than that of other emotions whereas angry and happy speech have higher median values and greater spread in RMS energy.

Fundamental frequency

Fundamental frequency (F0) is also a crucial feature to evaluate in order to analyze speech behavior. To determine F0, an algorithm based on a Fourier transform followed by a spectral product from which maximum is extracted as the first steps of Klapuri [27] is used.

According to [26], the mean F0 is lower in sad speech compared to that of neutral speech and angry and happy speech have higher F0 values and greater variations compared to that of neutral speech.

Also, simultaneous analysis from RMS energy and fundamental frequency is a really interesting factor to study for mood classification.

Sorting the data

In order to sort the data, an ideal method would be to pick the grains around a target RMS energy value with minimum deviation or use a clustering method such as k-means. The issue in our case is that there are not enough values with significant difference from the recordings, so this approach would lead to whether too few grains for significantly different values (and then lead to repetitions), or not enough difference from the neutral setting when selecting a target value or limit that allows to pick more grains. Consequently, only the grains with higher and then lower RMS energy were selected. The neutral grains are selected around the average value.

This method has the advantage of allowing significant differences of settings concerning the excitement or arousal of the crowd. The drawback is that it also leads to some interruptions when selecting two consecutive grains with significant energy difference.

The first solution to this problem would be acquiring more data with significant differences concerning the audio features used, that would make the use of clustering methods possible and interesting. Another solution would be using a more complex algorithm that takes into account the audio features of the previous grains to avoid abrupt variations between two consecutive grains within the selected range of the target.

4

Generation of the texture

In this part, the generation of gibberish speech for one speaker/source will be described, in terms of model, synthesis and implementation.

4.1 Speech model

With the previous steps of implementation a continuous restless concatenation of syllables is obtained for each timbre which is perceptually quite far from a realistic situation, especially when listening with spatialization. Thus a model that includes the natural pauses of the speaker is needed. For this, a two stages second order Markov chain models at two different time scales is used.

4.1.1 Markov chain model

A Markov chain is a process that consists of a finite number of states and some known probabilities P_ab, where P_ab is the probability of moving from state "a" to state "b". This can be visualized in a simple way in Figure 4.1:

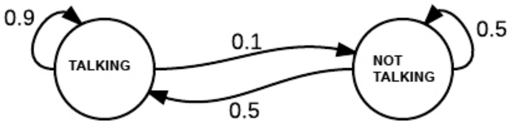


Figure 4.1: Simple example of a Markov chain model where the state "a" is "Talking" and "b" is "Not talking"

4.1.2 Two stages second order Markov chain model

In our case a second order Markov chain process is used, which means that the state probability is based on the two previous states. The states here are ”talking” and ”not talking” respectively. For the training, several speeches found on the website *freesound.org* were used, with interesting differences in terms of pace or chattiness. From the analysis and syllable detection of this speech material with the algorithm detailed in 3.3 A transition matrix is then calculated. This matrix is storing all the state transitions from the two previous events.

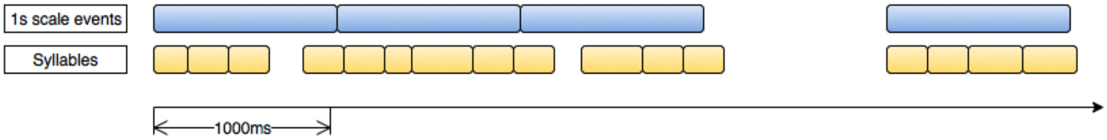


Figure 4.2: Example of the speech model using two stages: syllable time scale and 1000ms time scale

A first training is executed with a time scale of 1000 ms on the whole recording in order to analyze long term occurrences of pauses or speech. Afterwards, only sentences are taken into account and concatenated without long pauses and another analysis is done based on a shorter time scale of 250 ms corresponding approximately to the average length of a syllable. This second training allows to capture short pauses from the speaker.

Considering the implementation, longer term occurrences are calculated upfront independently, using a metronome. A small randomization of the value of this metronome with an arbitrary range of values very close to 1000ms (used for the training) can be used to implement additional irregularity. The syllable sized events are calculated right after on the condition that the previous stage state is ”talking” at every new syllable that is triggered.

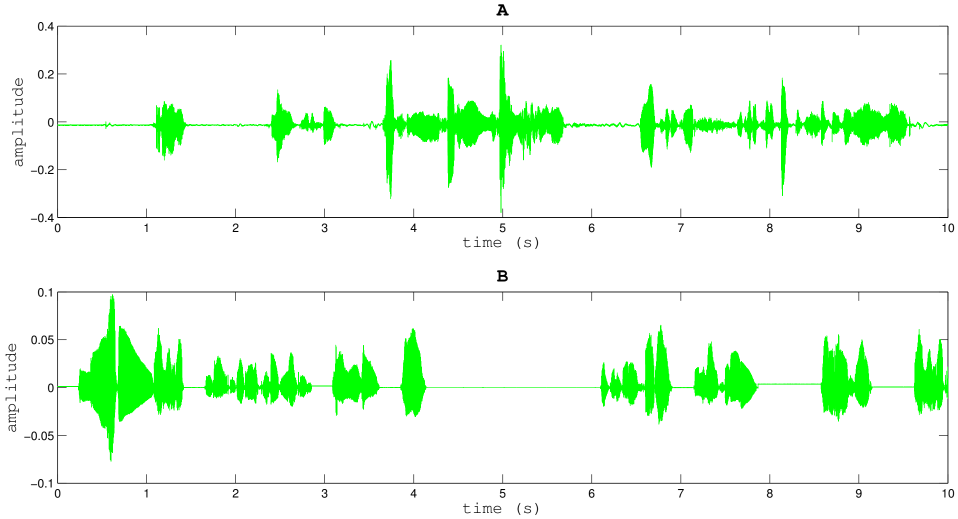


Figure 4.3: 10 second sample examples of (A) Speech material used for training and (B) the resulting synthesis with the model used

An extract of one speech material used for the training and a sample of the generated synthesis from the model are shown on Figure 4.3

This simple model was implemented on *Pure Data* and lead to very interesting and convincing results. A more complex model and training would be a key element to explore for further improvements.

4.2 Synthesis engine

The base of the implementation is the same as a simple granular synthesis player. The audio material is stored in arrays on *Pure Data*. Due to accuracy limitations when reading long arrays with tabread∼.pd from *Pure Data*, namely about 3 minutes at 44100 Hz sampling rate, it was decided to store all the concatenated syllables from the previous stages in different arrays for each single speaker and each selected area of target from audio features.

The *Pure Data* object grain∼.pd from [7] was used to achieve the playing of the grain. It takes as input: the position of the grain in samples, the array from which the grain is taken, the selected envelope to be applied and its desired length. More details on implementation of granular synthesis can also be found in the *Pure Data* tutorial [25]. The choice of envelope was made arbitrarily from the best sounding results based on tests made on trapezoid, Hahn, Gaussian and triangular envelopes. A Gaussian envelope was chosen, providing satisfactory results in terms of fluidity of the produced gibberish speech while conserving the spectral content of the voice. The following equation describes the Gaussian curve:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

with σ the standard deviation that allows to control the spread of the bell and is for us the crucial setting and μ the center peak.

The onsets corresponding to the position of the syllables in each array are stored in list objects. The length values of these syllables are stored in a separate list in the same order. When a syllable is randomly picked from within an array, its corresponding length is chosen simultaneously and used for two purposes: as a delay for triggering the next grain and as the length of the envelope of the current grain. Instead of the object "random", the "urn" object is used. The "urn" object performs a random selection within a specified range without duplicate numbers which allows avoiding repetitions and disturbing stammering artifacts.

All elements listed above and the array corresponding to the corpus of the target (for example "excited" syllables) are sent to the grain player in the instant of each new syllable being triggered. Therefore, a continuous flow of syllables with no gap or overlap is obtained. This continuous flow is modulated upfront in real-time with pauses, based on the two stages Markov chain model described in 4.1 resulting in more realistic and natural generation of gibberish speech.

4.3 Increasing the number of streams of speech

The sources are then duplicated with similar algorithms fed with the .wav files, sorted onset and corresponding length of syllables equivalent to each of our 10 recorded speakers. Afterwards, those 10 sources can be multiplied by performing small pitch shift to modify the timbre and give the illusion of perceiving a wider range of voices. The pitch modification should be subtle enough in a limited range in order to keep the "human" characteristic of the speech. The message *seed* from *Pure Data* was also used in order to make sure every random object is not correlated.

4.4 Implementation on *Pure Data*

The implementation of a selection of interesting parts on *Pure Data* can be consulted in appendix:

Figure 7.1: One of the stream of speech with comments

Figure 7.2: The main patch

Figure 7.3: The second order Markov chain process

Figure 7.4: The Markov chain model selector

Figure 7.5: The grain player

Figure 7.6: Writing of the Gaussian envelope

Figure 7.7: The control interface

5

Auralization

As defined in [4] ”the auralization of virtual environments describes the simulation of sound propagation inside enclosures, where methods of Geometrical Acoustics are mostly applied for a high quality synthesis of aural stimuli that go along with a certain realistic behavior.” This part aims to explain how the crowd synthesis is spatialized for a binaural experience within a virtual acoustic environment.

5.1 First tests with wave-field synthesis

In order to evaluate the quality of the synthesis in a spatialized situation, the first steps of implementation were tested in the wave-field synthesis studio of the Technische Universität Berlin. Wave-field synthesis is a spatial audio rendering technique for virtual acoustic environments. It produces artificial wave fronts synthesized by a large number, usually arrays, of individually driven speakers. The details will not be explained here, but the theory can be consulted from [17].

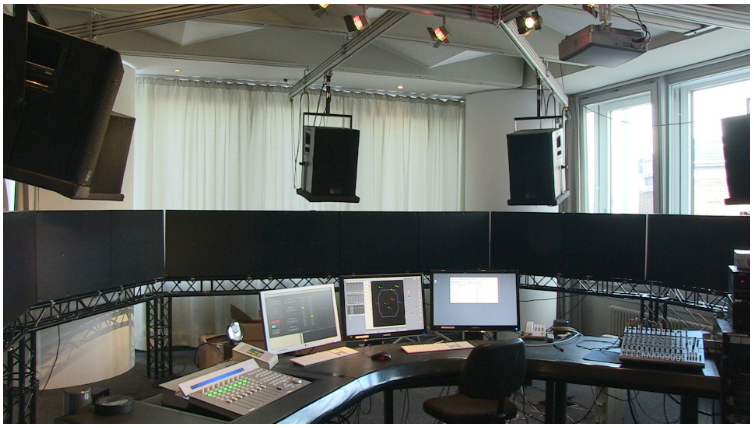


Figure 5.1: Wave-field synthesis studio of Technische Universität Berlin used for listening tests

This step was really useful for identifying the imperfections of the synthesis. Some of these defects were concealed when listening on simple stereo set. The main example of this is the case of using only one timbre per source. Indeed, to be able to detect the position of the source allows us to point out lack of fluidity when one source can use different speakers and timbres. The need to detect syllables instead of simply onset was more obvious from the listening tests in question as well. Furthermore, it revealed the need to not synthesize constantly and include pauses for every source to offer a more realistic feeling and aspire to the speech model described earlier in this report.

5.2 Implementation in virtual acoustic environment

The goal of this project is the implementation in virtual acoustic environments, and more precisely in our case: a binaural environment. For this purpose, the *SoundScape Renderer* software [20] is used. It is a tool for real-time spatial audio reproduction providing a large variety of rendering algorithms including binaural techniques. Binaural resynthesis is more efficient with head tracking. Therefore, the binaural renderers of the *SoundScape Renderer (SSR)* include built-in support for a selection of tracking devices.

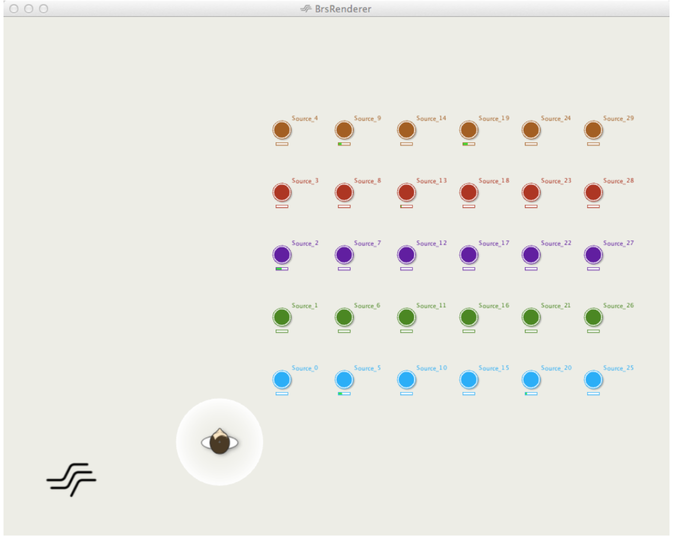


Figure 5.2: Screenshot from the *SoundScape Renderer* interface and the 30 sources, the head in the circle displays the head motion

Binaural rendering uses binaural room impulse responses to reproduce the sound arriving to the listeners ears. A pair of BRIRs is chosen depending on the position of the listener's head. These BRIRs are applied to the input signal by convolution. Users head movement is included using a head-tracking device and is used to select the right BRIR for each head orientation and make the soundfield remain immobile and not follow the head motion, resulting in a more immersive and realistic experience of the virtual environment. The binaural room impulse response (BRIR) used were calculated upfront from Raven [4] (details in 1.4.4) in the acoustic scene of the Roman Forum in Rome shown on Figure 5.3. The sources are located on 3 circles around the listener at 5, 15 and 25 meters. A configuration script file of the scene is written and specifies the BRIRs used in order to load them into the *SSR*. Then the 30 outputs from the *Pure Data* patch are connected to the 30 inputs of the *SSR* to obtain the auralized scene by listening to the *SSR* output.

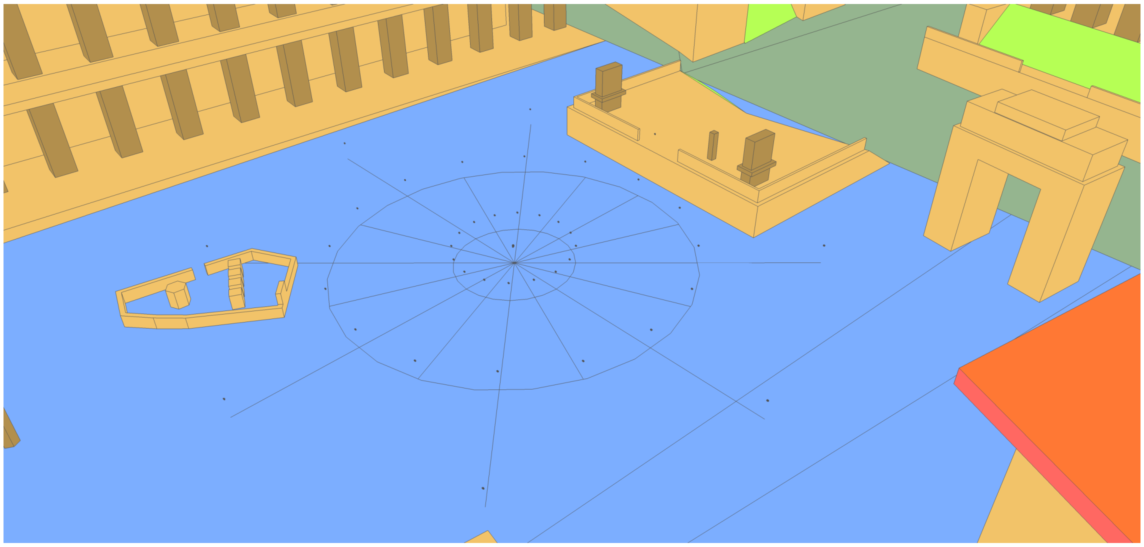


Figure 5.3: Source positions and virtual environment scene simulating the Roman Forum used for the simulations on *RAVEN*

5.3 From 30 to 96 streams

This method is limited by the high CPU consumption caused by the 30 convolutions produced by the *SoundScape Renderer* (about 80% of the CPU). The synthesis itself only takes about 10% of CPU usage for those 30 streams. Thus a method to increase the number of streams without incrementing the number of convolutions is needed. Two methods were experimented for this purpose.

5.3.1 Summing streams

Three circles respectively located at 5, 15 and 25 meters from the listener are now used. Each of those circles contains 8 sources placed as in Figure 5.4. Consequently, the number of convolutions is now reduced to only 24. The first and most basic method consists in simply summing streams (speakers) at one source.

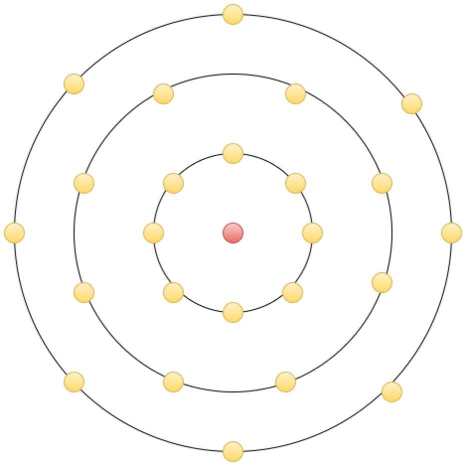


Figure 5.4: Schematic representation of the position of the 24 sources over the 3 circles at respectively 5, 15 and 25 meters from the listener

There are numerous possibilities for distribution. In order to make a choice among them, The assumption is made that the further from the listener the source is, the more people the space can contain. As an example, the distribution of 96 streams over the 3 circles of 8 sources from inner to outer circle can be: 2x8 / 4x8 / 6x8.

5.3.2 Amplitude panning

Amplitude panning [19] is another solution for a more sophisticated distribution of streams over the available sources. The idea is to recreate phantom sources between two sources with weighted amplitude on each side. This is achieved by distributing the amplitude of the phantom sources according to the weights in the graph of Figure 5.5.

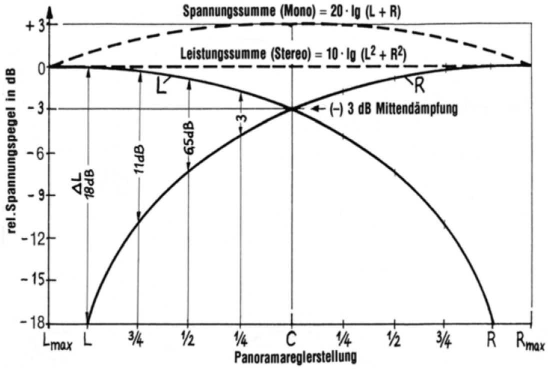


Figure 5.5: Panning distribution weights, from [18]

5.3.3 Distribution of the streams over the sources

In order to be able to compare the two methods, the same distribution for both techniques is used. The effect of placing the streams closer (inner circles 1 and 2) or further (outer circles 2 and 3) is also evaluated. To test the synthesis, the following choices of distributions for both methods were used arbitrarily and assuming that the further the circle is from the listener, the more people can fit in the space.

sources	circles	sources on inner to outer circle
16	2	8 8
16	3	4 4 8
32	2	16 16
32	3	8 8 16
96	2	32 64
96	3	16 32 48

6

Listening tests and results

6.1 Set of tested stimuli

It was decided to test all combinations of the following parameters:

streams	crowd behavior	distribution	panning
16	neutral	all circles	no panning
32	excited	2 inner circles	panning
96	quiet	2 outer circles	

For 16 sources, there are less streams than sources, so no such technique as panning was applied. This leads us to 45 different stimuli. The choices for distribution of the streams over the sources can be consulted in the table of 5.3.3.

The streams were pre-rendered and recorded on *Ardour* from the anechoic outputs of the *Pure Data* implementation. The mix-down to include panning and summing techniques previously described for the 32 and 96 streams was executed with *Matlab* in preparation of the material for 24 channels. In order to create a scene closer to what someone would hear in a real life situation, an additional subtle background noise was recorded in a park of Berlin with binaural in-ear microphones. The choice was made to record in a park to avoid sounds of cars which are not expected for an historical environment.

6.2 Test procedure

The tests were conducted during one week at the very end of this internship. 17 unpaid participants living in Berlin were attending. Nine of them were native German speakers, the other ones usually had good knowledge of German. More than half of them were already interested in sound synthesis or soundscape related topcis thanks to their hobbies or occupation. Notably, 5 of them were familiar with granular synthesis and 2 others were working as sound editors for radio and documentaries.

The tests occurred in the studios of Audio Communication Group. High quality headphones (AKG K1000) equipped with a head-tracker were used. The interface was made on *Pure Data* and consisted of 45 numbered bangs. Each of them was sending OSC messages to *Ardour* in order to play the selected stimulus. Each session lasted between 30 and 45 minutes.

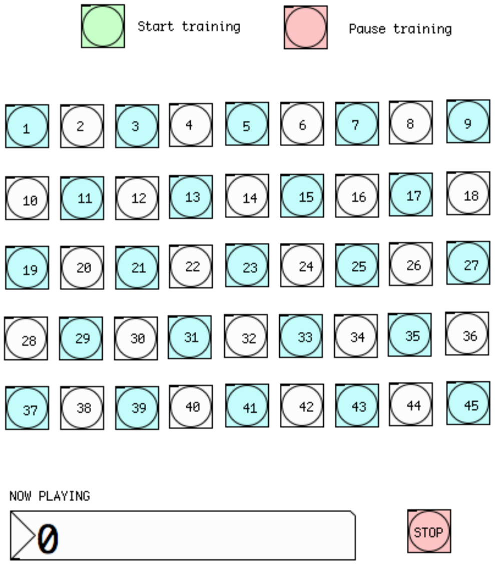


Figure 6.1: Interface for the listening test sending OSC messages to *Ardour*

For each stimulus, they were asked to rate:

- Naturalness: From 0 (unnatural) to 7 (natural)
- Excitement: From 0 (calm) to 7 (excited)
- Distance: From 0 (really close) to 7 (distant)
- Evaluate the number of people

The order of the questions (number of the playing crowd) was generated randomly. Before starting to listen to the audio which was to be rated, the participants had to listen a training set of 6 selected samples that illustrated the range of sounds they were about to rate and make them used to the sounds. This helped to let them get used to the sounds and additionally, they were also asked to use the training round to consider the parameters they will have to rate later on.

The end of the questionnaire consisted of an enquiry about the noticeable artifacts in the material and information about their sound knowledge and experience.

6.3 Results

The full list of results for each stimuli with mean and standard deviation for the ratings of naturalness, excitement and guess of the number of people is available in the appendix Table 7.1. In the first place, our main focus concerns the naturalness evaluation.

Our experiment follows a factorial design, where each observation has data on all factors, and we are able to look at one factor while observing different levels of another factor.

Important note: An ANOVA (analysis of variance) analysis would be well suited in order to evaluate the significance of these factors concerning the listening test results. Nevertheless the time left after the listening tests was too short to fully understand and apply this method in a n-factor situation before the deadline of this report. Also, more participants would be necessary to improve the data. Consequently, the following observations are only based on tendencies on variables considered independently and on Table 7.1.

Naturalness is the most important criterion at this state of the synthesis. The following graphs display average, minimum and maximum evaluation, for the four factors considered independently.

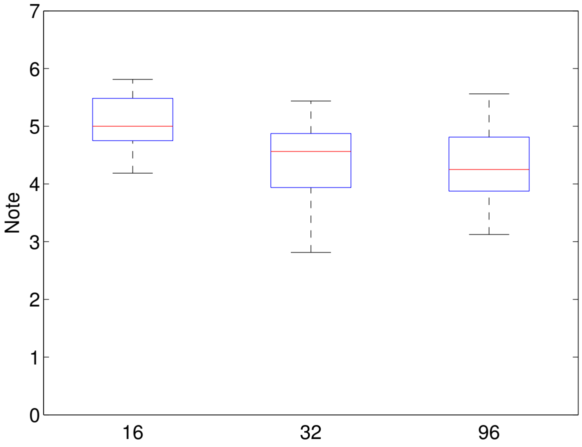


Figure 6.2: Boxplot with average, first and third quartile in naturalness evaluation in consideration of the number of sources independently

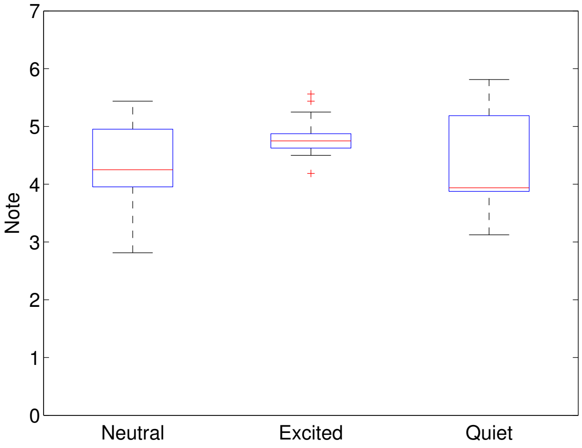


Figure 6.3: Boxplot with average, first and third quartile in naturalness evaluation in consideration of the excitement level independently

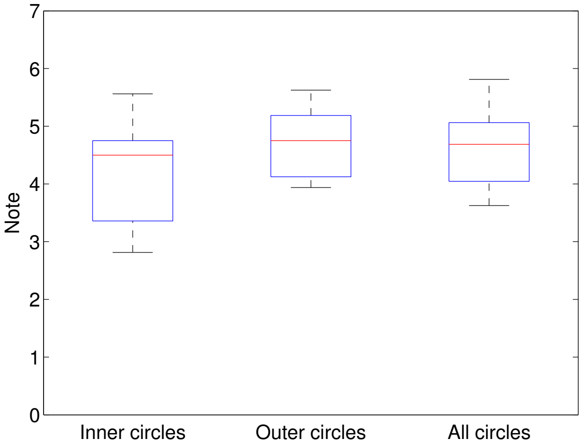


Figure 6.4: Boxplot with average, first and third quartile in naturalness in consideration of the position on the circles independently

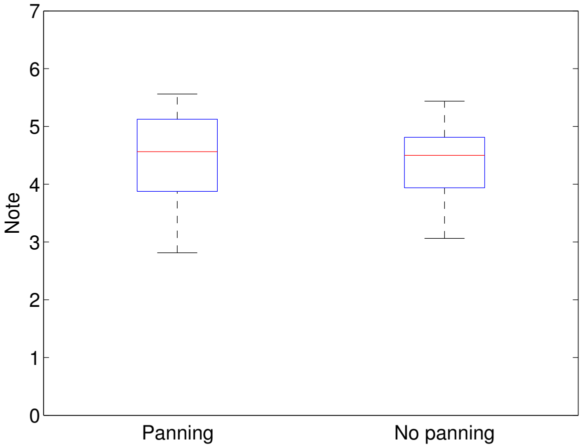


Figure 6.5: Boxplot with average, first and third quartile in naturalness evaluation in consideration of the use of panning independently

6.4 Observations and analysis

Firstly, 91% of the stimuli obtained an average rating above the median value of the scale in term of naturalness.

Naturalness

The average evaluation of naturalness tends to decrease with the number of streams. This can be explained by the augmentation of repetitions when multiplying the streams. The pitch shift could also be a reason.

The position of the sources in the scene seems to have an impact on the perceived naturalness. Positioning sources on the outer circles or on all circles leads to better results compared to the settings on the inner circles only. Logically, artifacts and flaws encountered in the streams synthesis are singled out when the sources are closer.

Quiet settings with a large amount of streams were usually poorly rated. This could be due to the situation of 96 people whispering which is itself very unusual. This could be improved by applying a different training for the speech model depending on the mood or excitement degree. For example, the quiet crowd would probably exhibit longer pauses.

The use of amplitude panning instead of simply summing streams for the settings with more streams than sources does not seem to provide substantial improvement.

Perceived artifacts and solutions to improve naturalness

Among the noticed artifacts enquired at the end of the session, two of them were recurring and can be considered the main reasons of the stimuli being perceived as unnatural.

The first being the distinguishable repetitions of overly long syllables. The available data is indeed too small when multiplying the number of sources from the same material. This is also a consequence of the syllable segmentation algorithm which can sometimes output lenghty elements (even though every detection above 1000ms was removed). On the other side, the same long syllables were also mentioned as participating to the natural feel when no repetition was perceptible. The acquisition of more material from each speaker, and an improvement of the syllable detection algorithm could easily solve this problem.

The second annoying artifact was the apparition of sudden grains with really high RMS energy compared to the preceding grain for the ”excited” settings and close sources. Very few grains with such properties were available, so the solution would have been to define a maximum deviation to assure their removal. An algorithm that takes into account audio features from the previous grains could also offer a solution to reaching those grains with such small volume of material. Acquiring larger amount of material with this kind of high RMS energy during the recordings would also be a solution to accessing the realms of an extensively excited crowd behavior.

Size of the crowd

A large standard deviation is often observed concerning the evaluation of the size of the crowd, especially for the 96 streams settings. The proximity of the sources and the excitement are influential parameters in this perception. Furthermore, this question could be considered delicate as in most real life situation, not every single person of the surrounding crowd is talking. Considering the percentage of people talking is real life situations depending on the excitement level could be included for improvement.

Excitement

The intended excitement behavior from RMS energy based corpus selection was always in sync with what the listeners experienced. It is interesting and reasonable to notice that for the same corpus, when the number of sources is increasing, the perception of excitement is increasing.

7

Conclusion and perspectives

Conclusion

In this work, a crowd synthesis method based on granular and corpus-based concatenative synthesis was implemented. This method is totally suited for the desired parametric characteristic of the produced synthesis. Many settings can be changed with the implemented synthesis: number of people in the crowd, arousal/valence of the crowd, chattiness according to different speech model and position in the space. Audio features applied on the grains or hand labeling allows us to classify the data of the corpus upfront for each speaker. Afterwards, suited algorithms and real-time synthesis are used to generate single timbre streams of gibberish speeches. Different pitches and randomization allow us to multiply the number of these streams. Auralization was produced using the *SoundScape Renderer*, with binaural room impulse responses pre-calculated on *RAVEN*. Techniques such as amplitude panning are used in order to multiply the number of speakers with a constant number of sources to reduce the number of convolutions.

The resulting synthesis is highly satisfying on many aspects and was well received in terms of perceived naturalness during the listening tests for most of the settings. With a very limited amount of audio material, hours of crowd synthesis can be generated, and different parameters can be changed in real-time. This tool can be especially exciting and beneficial when applied within the environment of motion picture or virtual reality where the sound designer would be able to follow the action, and, for example, change the mood or excitement of the crowd with a slider along with the progression of the scene. Also, many different types of crowd textures could be generated with only a small amount of data with no need to own and store hours and gigabytes of recordings. The listening tests showed interesting features and confirmed some ideas for improvement in further implementations. Nevertheless the listening test results and observations should be reconsidered by using tools such as ANOVA (analysis of variance) to evaluate the significance of the factors and by conducting the tests on more participants.

Perspectives

Even though a parametric crowd synthesis that can be considered further than proof-of-concept resulted from his work, several directions for improvement can be mentioned already. First step would be the acquisition of more material for the corpus, especially including more data for different moods and behaviors. This would allow the use of more sophisticated classification tools to lead to significantly wider range of behaviors and would allow greater degrees of freedom for navigating in the corpus. Additionally, it would also allow to multiply the number of streams without repetitions. The syllable detection algorithm could also be improved for more accuracy and to avoid artifacts such as perceived repetitions from exceedingly long grains. Furthermore, it would be interesting to explore a deeper focus on the use of audio features to detect extensively complex moods, further than just calm and excited (differentiating happiness from anger for instance). Finally, it could also be interesting to incorporate inter-grain audio feature relationships from training on real speech data into the speech stream generation.

For further development, it would be necessary to explore the possibilities of extension of this technique to other texture soundscapes such as rain or a conflagration scene. The major differences would be in considering what are now the sound units, what parameters could be modified and what model is used for the generation of streams. Due to complications and limitations from the implementation on *Pure Data*, moving on another system could also be considered.

Bibliography

[1] V. R. Algazi, R. O. Duda, and D. M. Thompson, “Motiontracked binaural sound,” J. Audio Eng. Soc, vol. 52, no. 11, pp. 1142–1156, 2004.

[2] D. Schwarz, “State of the art in sound texture synthesis,” in Digital Audio Effects (DAFx), Paris, France, Sep. 2011.

[3] D. Schwarz et B. Caramiaux, “Interactive sound texture synthesis through semi-automatic user annotations,” in Sound, Music, and Motion: Lecture Notes in Computer Science, Vol. 8905, Marseille, France: Springer International Publishing, 2014, pp. 372–392.

[4] D. Schröder et M. Vorländer, “RAVEN: a real-time framework for the auralization of interactive virtual environments,” in Forum Acusticum 2011, Aalborg, Denmark, 2011, pp. 1541–1546.

[5] S. Kersten, ”Statistical modelling and resynthesis of environmental texture sounds”,Chapter 3 ”A hidden Markov tree model for sound textures”, PhD, Universitat Pompeu Fabra, 2015.

[6] S. Kersten, Sound examples for Chapter 3 ”A hidden Markov tree model for sound textures”

http://declaredvolatile.org/research/phd_thesis/sound_texture_synthesis_smc2010/index.html

[7] L. Sutton, Granita - Minimalist granular synthesis - Pd Community Site, http://lorenzosu.altervista.org/pd/granita/

[8] J. Janer, R. Geraerts, WG. van Toll, J. Bonada, ”Talking soundscapes: Automatizing voice transformations for crowd simulation”, AES 49th International conference, London, UK, 2013 February 6-8.

[9] EmoDB - Berlin Database of Emotional Speech
http://emodb.bilderbar.info/start.html

[10] A. Harma, "Automatic identification of bird species based on sinusoidal modeling of syllables," Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on , vol.5, no., pp. V- 545-8 vol.5, 6-10 April 2003

[11] Analysis and Synthesis of Sound Textures, Nicolas Saint-Arnaud and Kris Popat.

[12] Classification of Sound Textures, PhD, Nicolas Saint-Arnaud.

[13] B. Groethe, M. Pecka, D. McAlpine: Mechanisms of Sound Localization in Mammals, Physiol Rev 90: 983–1012, 2010 doi:10.1152/physrev.00026.2009

[14] T. Potisk: Head-Related Transfer Function, University of Ljubljana Faculty of Mathematics and Physics Seminar Ia, 9th January, 2015.

[15] Xiao-li Zhong and Bo-sun Xie (2014). Head-Related Transfer Functions and Virtual Auditory Display, Soundscape Semiotics - Localization and Categorization, Dr. Hervé Glotin (Ed.), ISBN: 978-953-51-1226-6, InTech, DOI: 10.5772/56907.

[16] http://avblog.nl/woordenboek/h/hrtf-head-related-transfer-function/

[17] A. J. Berkhout, D. de Vries, and P. Vogel, Acoustic control by wave field synthesis, Delft Universitoyf TechnologLya, boratoroyf Seismicasn dA cousticPs,. O.B ox5 046,2 600 GA Delft, The Netherlands.

[18] UdK Berlin: Sengpiel, 09.2003. Panpot: Die Regel- und Pegelunterschiede

[19] M. Frank, Localiation using different amplitude-panning methods in the frontal horizontal plane, Institute of Electronic Music and Acoustics, University of Music and Performing Arts Graz.

[20] M. Geier, J. Ahrens, S. Spors: The SoundScape Renderer: A Unified Spatial Audio Reproduction Framework for Arbitrary Rendering Methods 124th Convention of the Audio Engineering Society, 2008.

[21] Kleiner M., Dalenbäck B.-I., Svensson P.: Auralization - an overview, J. Audio Eng. Soc., Vol. 41, No. 11, p. 861-875, November 1993

[22] http://www.eumus.edu.uy/eme/ensenanza/electivas/csound/materiales/book_chapters/11fischman.html

[23] C. Roads, Microsound (2001), page 91

[24] C. Roads, The Computer Music Tutorial (1996), Chapter 5, page 172

[25] http://www.pd-tutorial.com/english/ch03s07.html

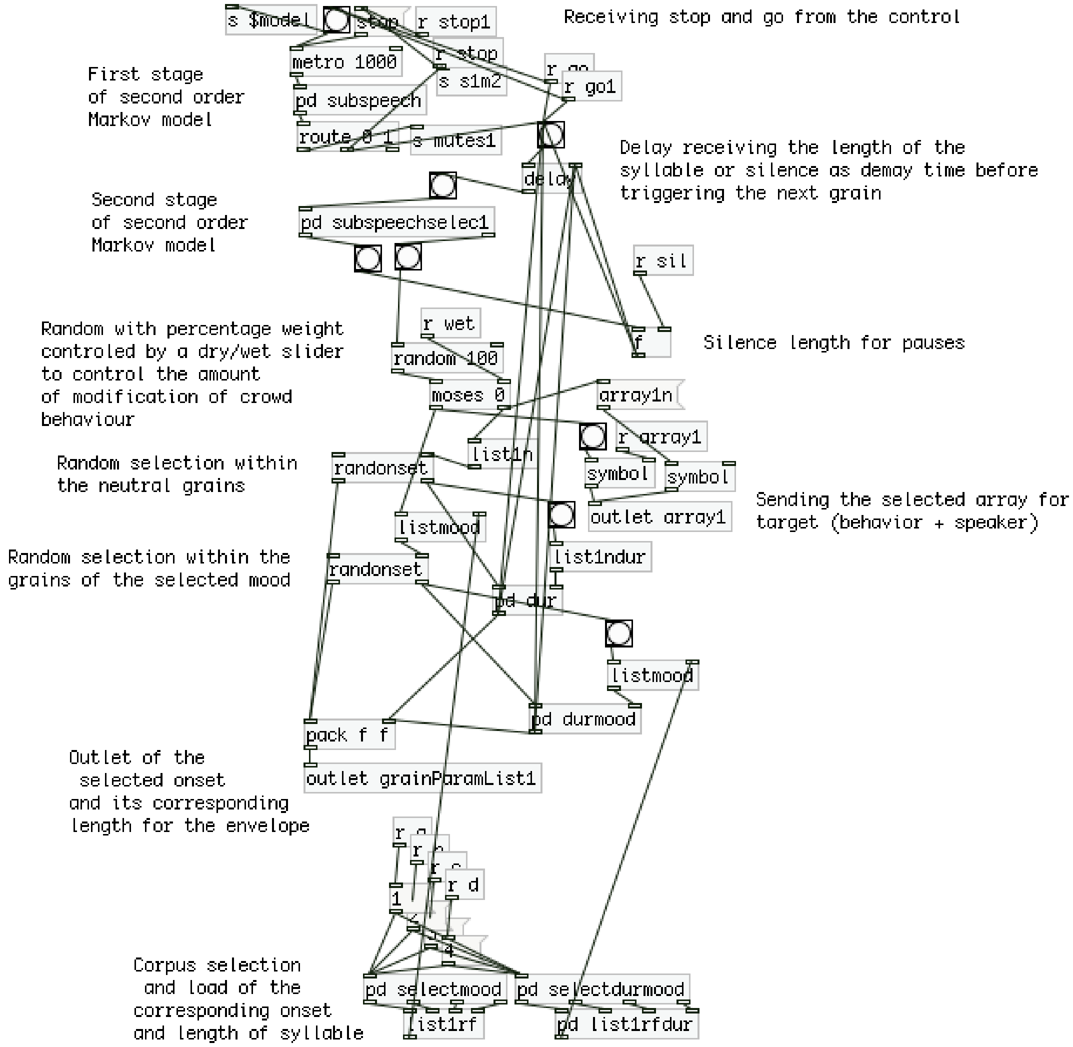
[26] S. Yildirim, M. Bulut, C. Min Lee, A. Kazemzadeh, C. Busso, Z. Deng S. Lee, S. Narayanan, An acoustic study of emotions expressed in speech, Emotion Research Group, Speech Analysis and Interpretation Lab, University of Southern California, Los Angeles.

[27] A. Klapuri, Multipitch Analysis of Polyphonic Music and Speech Signals Using an Auditory Model, IEEE transactions on audio, speech, and language processing, VOL. 16, NO. 2, February 2008.

[28] M. Jeub, M. Schafer, and P. Vary, A binaural room impulse response database for the evaluation of dereverberation algorithms, Institute of Communication Systems and Data Processing, RWTH Aachen University, Germany

[29] S. Pelzer, L. Aspöck, D. Schröder and M. Vorländer, Interactive Real-Time Simulation and Auralization for Modifiable Rooms, JOURNAL OF BUILDING ACOUSTICS, Volume 21 · Number 1 · 2014

Appendix



Receiving stop and go from the control

First stage
of second order
Markov model

Delay receiving the length of the
syllable or silence as demay time before
triggering the next grain

Second stage
of second order
Markov model

Random with percentage weight
controled by a dry/wet slider
to control the amount
of modification of crowd
behaviour

Silence length for pauses

Random selection within
the neutral grains

Sending the selected array for
target (behavior + speaker)

Random selection within the
grains of the selected mood

Outlet of the
selected onset
and its corresponding
length for the envelope

Corpus selection
and load of the
corresponding onset
and length of syllable

Figure 7.1: One of the 30 streams of speech

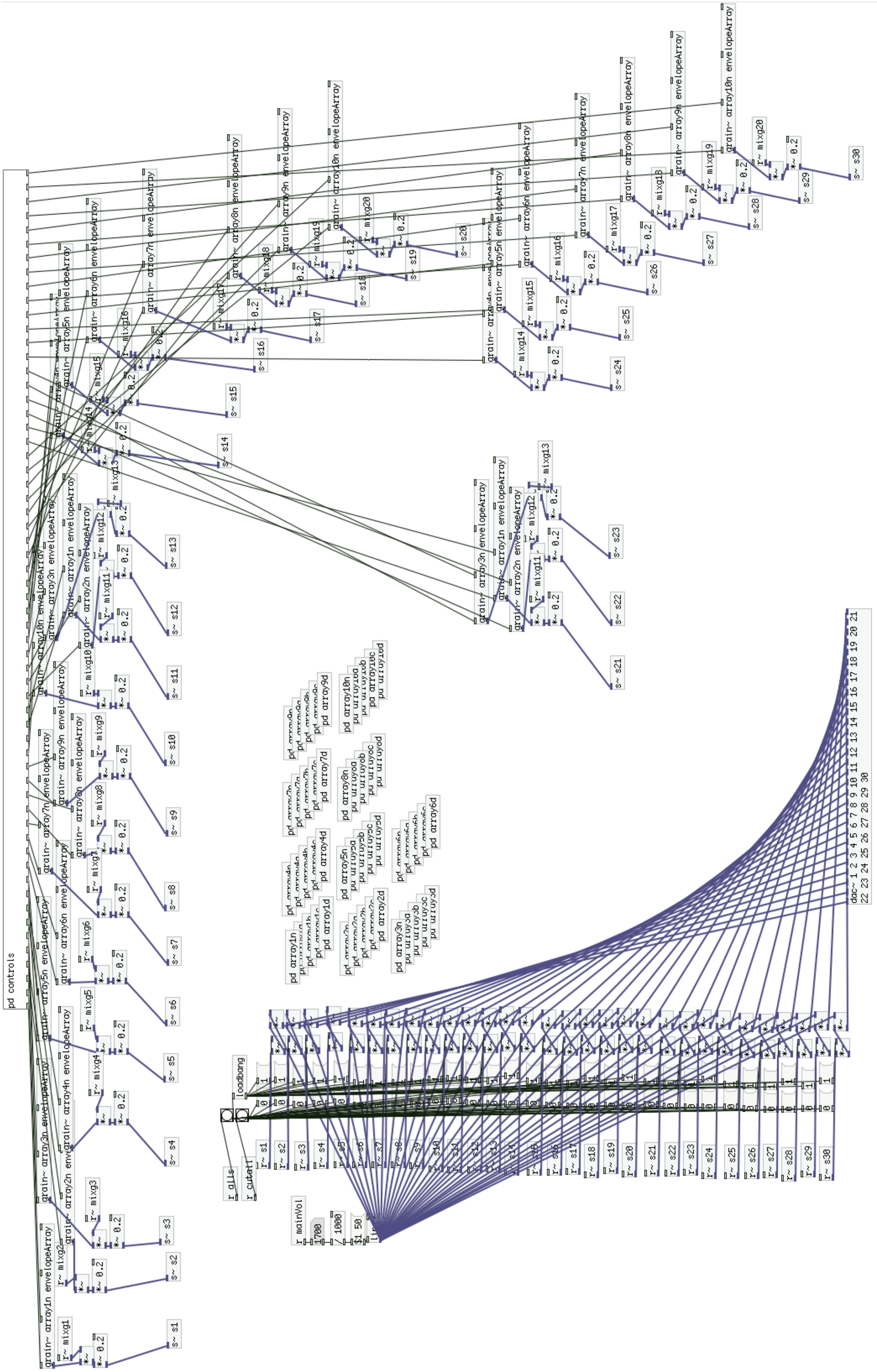


Figure 7.2: Main patch

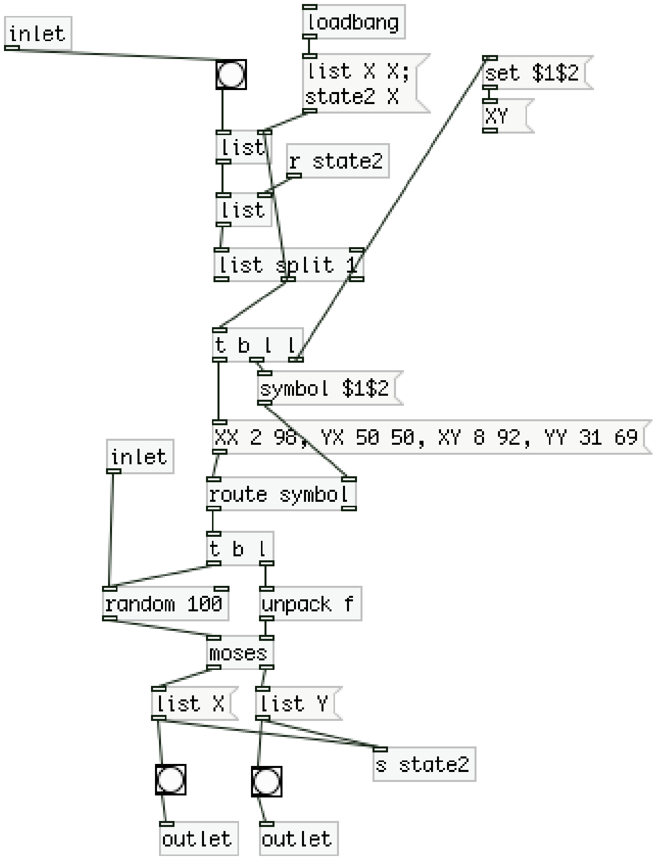


Figure 7.3: Implementation on Pure Data of second order Markov chain

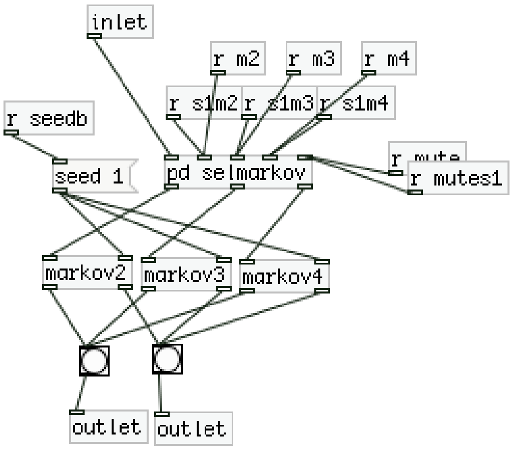


Figure 7.4: Markov model selector, corresponding to *pd subspeech*

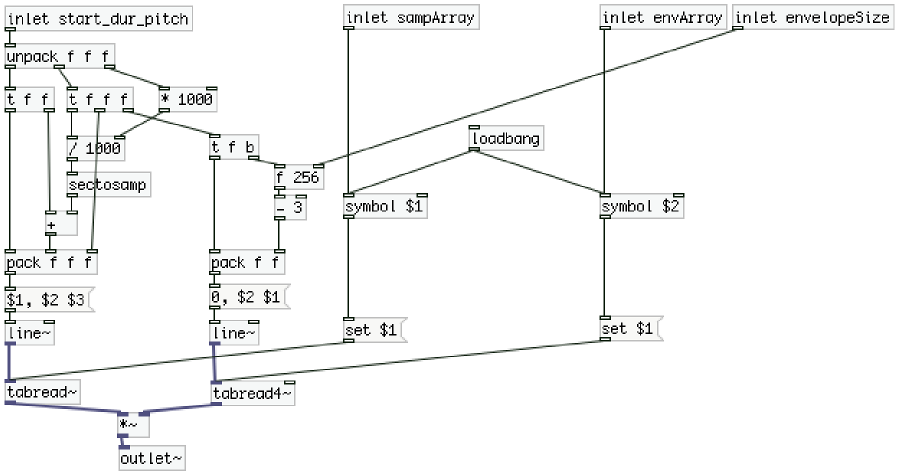


Figure 7.5: Grain player: outputs the waveform at the selected onset and with the selected envelope and length[7]

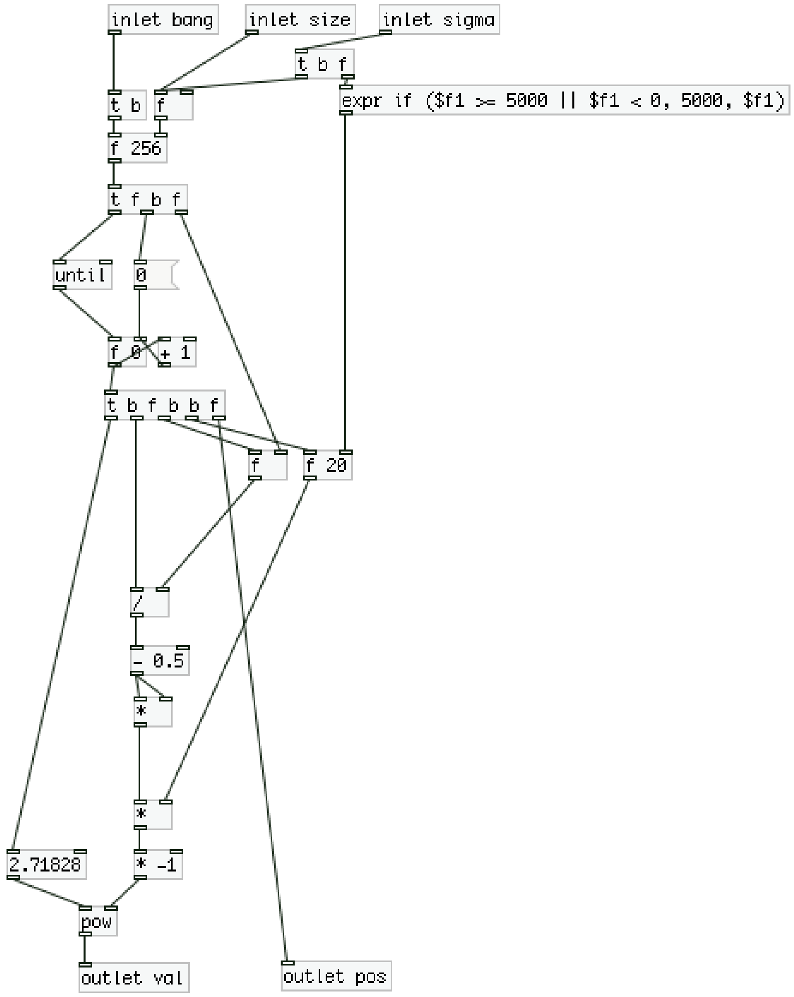


Figure 7.6: Gaussian envelope, outputs value and position to *tabwrite Pure Data* object [7]

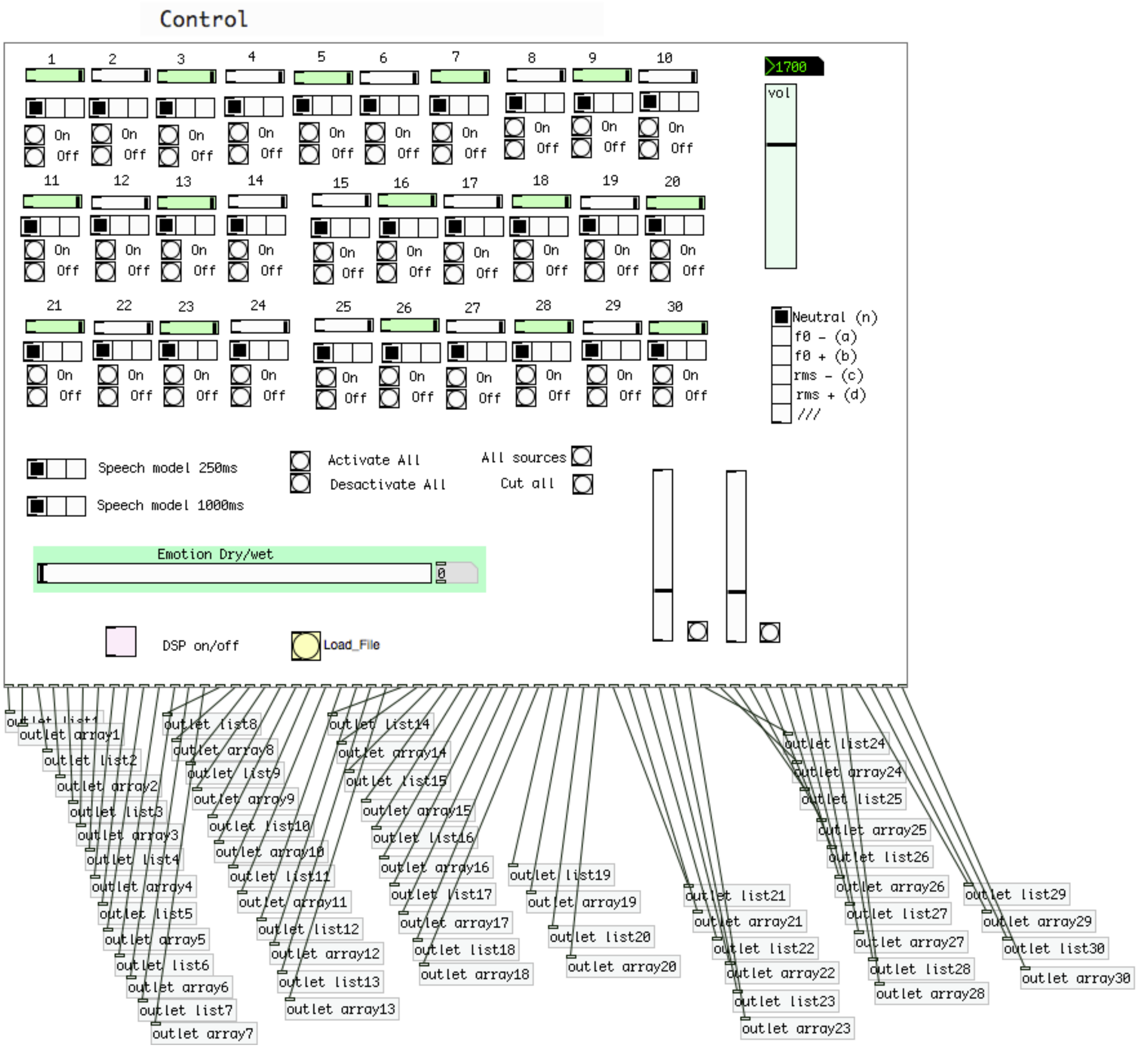


Figure 7.7: Control interface for choosing the corpus, activating the sources, and selecting speech model

Table 7.1: Results of the listening tests

Stim.	Streams	Behav.	Distr.	Pan.	Excitment		Naturalness		Size	
					Mean/7	StdDev	Mean/7	StdDev	Mean	StdDev
1	16	n	in	NA	1,75	0,77	5,44	1,26	20	16
2	16	e	in	NA	4,69	1,08	4,75	1,84	34	19
3	16	q	in	NA	1,31	0,87	4,75	1,57	15	13
4	16	n	out	NA	1,31	1,25	5,00	2,07	14	7
5	16	e	out	NA	4,00	1,83	4,75	1,57	49	40
6	16	q	out	NA	0,69	0,79	5,63	1,67	21	35
7	16	n	all	NA	1,13	0,62	5,13	1,67	15	10
8	16	e	all	NA	3,63	1,20	4,19	1,33	41	30
9	16	q	all	NA	0,50	0,52	5,81	0,83	19	22
10	32	n	all	pan	2,50	1,10	5,13	1,54	43	36
11	32	e	all	pan	4,81	1,05	5,44	1,15	65	35
12	32	q	all	pan	2,13	1,36	4,50	1,32	34	37
13	32	n	in	pan	3,81	1,22	2,81	1,72	47	42
14	32	e	in	pan	5,69	0,79	4,75	1,48	75	63
15	32	q	in	pan	2,69	1,08	3,88	1,50	46	48
16	32	n	out	pan	2,44	1,26	4,00	1,37	30	17
17	32	e	out	pan	4,50	1,41	4,63	1,78	101	79
18	32	q	out	pan	1,88	1,20	5,31	1,49	34	30
19	96	n	all	pan	4,06	1,24	3,63	1,82	108	164
20	96	e	all	pan	6,63	0,50	4,88	1,71	198	166
21	96	q	all	pan	2,88	1,41	3,88	1,67	56	55
22	96	n	in	pan	4,63	0,96	4,00	1,51	94	73
23	96	e	in	pan	6,31	0,70	5,56	1,41	283	259
24	96	q	in	pan	4,31	1,30	3,13	2,03	115	143
25	96	n	out	pan	4,00	1,21	4,69	1,35	135	140
26	96	e	out	pan	5,75	0,86	5,25	1,06	218	225
27	96	q	out	pan	2,19	1,42	3,94	2,02	104	113
28	32	n	all	sum	2,69	1,20	4,25	1,95	26	20
29	32	e	all	sum	4,81	1,22	4,88	1,75	83	57
30	32	q	all	sum	2,00	1,10	4,81	1,47	26	21
31	32	n	in	sum	3,81	0,54	3,06	1,84	44	36
32	32	e	in	sum	5,38	1,02	4,63	2,00	77	35
33	32	q	in	sum	3,31	1,30	3,88	1,93	42	27
34	32	n	out	sum	2,81	0,91	3,94	1,65	34	15
35	32	e	out	sum	4,94	1,24	4,50	1,67	119	124
36	32	q	out	sum	1,38	0,96	5,44	1,75	42	49
37	96	n	all	sum	3,75	0,77	4,00	1,90	73	55
38	96	e	all	sum	5,88	1,09	4,69	1,74	208	174
39	96	q	all	sum	3,25	1,34	3,88	1,63	80	77
40	96	n	in	sum	4,56	1,03	4,63	1,50	92	70
41	96	e	in	sum	6,38	0,72	4,50	2,13	229	160
42	96	q	in	sum	3,88	1,26	3,19	1,87	94	80
43	96	n	out	sum	3,25	1,39	4,81	1,72	103	85
44	96	e	out	sum	5,75	0,86	4,88	1,63	200	119
45	96	q	out	sum	2,75	1,81	3,94	2,17	130	94