Institut für Hörtechnik und Audiologie
Jade Hochschule Oldenburg
Doktor Jörg Bitzer

AUDIO-RENDERING

Multi-microphone recordings
(EigenMike32) inside a WFS system
(Iosono) through Ambisonics
techniques using Matlab

David Peña Gil
21/08/2012
0. ABSTRACT

In the last years the multichannel audio has increased in leaps and bounds and not only in the playback techniques, but also in the recording ones. That is the reason of both things being in this project: a microphone array, EigenMike32 from MH Acoustics; and a playback system with Wave Field Synthesis technology, installed by Iosono in Jade Höchscule Oldenburg.

To link these two points of the audio chain, 2 different kinds of codification are proposed: the reproduction of the EigenMike32´s horizontal take, and the Ambisonics´ third order (High Order Ambisonics, HOA), a codification technique based in Spherical Harmonics through which the acoustic field is simulated instead of the different sound sources. Both have been developed inside Matlab´s environment and supported by the Isophonics´ scripts collection called Spatial Audio Matlab Toolbox.

To test these, a serial of tests were made in which they were compared with recordings made at the time by a Dummy Head, which is supposed to be the closest method to our hearing way. These tests included other recording and codifications made by a Double MS (DMS) from Schoeps which are explained in the project named “3D audio rendering through Ambisonics techniques: from multi-microphone recordings (DMS Schoeps) to a WFS system, through Matlab”. The way to perform the tests was, a collection made of 4 audios repeated 4 times for each recorded situation (a chat, a class, a street and college canteen or Mensa).

The results were unexpected, because the HOA´s third order stood under the Well valuation, possibly caused by introducing material made for a tridimensional array inside one made only by 2 dimensions. On the other hand, the codification that consisted of extracting the horizontal plane microphones kept the Well valuation in all the situations.

It is concluded that HOA should keep being tested with larger knowledge about Spherical Harmonics; while the other coder, quite simpler, can be used for situations without a lot of complexity with regards to spatiality.
0. RESUMEN

El audio multicanal ha avanzado a pasos agigantados en los últimos años, y no sólo en las técnicas de reproducción, sino que en las de capitación también. Por eso en este proyecto se encuentran ambas cosas: un array microfónico, *EigenMike32* de *MH Acoustics*, y un sistema de reproducción con tecnología *Wave Field Synthesis*, instalado *Iosono* en la *Jade Höchschule Oldenburg*.

Para enlazar estos dos puntos de la cadena de audio se proponen dos tipos distintos de codificación: la reproducción de la toma horizontal del *EigenMike32*; y el 3er orden de *Ambisonics* (*High Order Ambisonics, HOA*), una técnica de codificación basada en Armónicos Esféricos mediante la cual se simula el campo acústico en vez de simular las distintas fuentes. Ambas se desarrollaron en el entorno Matlab y apoyadas por la colección de scripts de *Isophonics* llamada *Spatial Audio Matlab Toolbox*.

Para probar éstas se llevaron a cabo una serie de test en los que se las comparó con las grabaciones realizadas a la vez con un *Dummy Head*, a la que se supone el método más aproximado a nuestro modo de escucha. Estas pruebas incluían otras grabaciones hechas con un Doble MS de *Schoeps* que se explican en el proyecto “Sally”. La forma de realizar éstas fue, una batería de 4 audios repetida 4 veces para cada una de las situaciones grabadas (una conversación, una clase, una calle y un comedor universitario).

Los resultados fueron inesperados, ya que la codificación del tercer orden de *HOA* quedó por debajo de la valoración Buena, posiblemente debido a la introducción de material hecho para un array tridimensional dentro de uno de 2 dimensiones. Por el otro lado, la codificación que consistía en extraer los micrófonos del plano horizontal se mantuvo en el nivel de Buena en todas las situaciones.

Se concluye que *HOA* debe seguir siendo probado con mayores conocimientos sobre Armónicos Esféricos; mientras que el otro codificador, mucho más sencillo, puede ser usado para situaciones sin mucha complejidad en cuanto a espacialidad.
# INDEX

0. ABSTRACT .............................................................................................................. 1
1. INTRO .................................................................................................................. 3
2. THEORETICAL FUNDATIONS .............................................................................. 4
   a. MICROPHONE ARRAY ..................................................................................... 4
   b. WAVE FIELD SYNTHESIS TECHNIQUE ....................................................... 5
   c. AMBISONICS .................................................................................................. 6
   d. MEAN OPINION SCORE .................................................................................. 9
   e. LATIN SQUARE ............................................................................................... 9
3. RECORDINGS ......................................................................................................... 10
   a. EQUIPMENT ...................................................................................................... 10
   b. SOFTWARE SUPPORT ................................................................................... 13
   c. LIST OF SITUATIONS .................................................................................... 14
4. PROGRAMMING METHOD ..................................................................................... 16
   a. CODERS .......................................................................................................... 16
   b. GUI AND TESTING PROGRAM ....................................................................... 18
   c. OTHERS ......................................................................................................... 19
5. LISTENING TESTS ................................................................................................. 20
6. ANALYSIS OF RESULTS ....................................................................................... 21
   a. CHAT ............................................................................................................. 21
   b. CLASS ............................................................................................................ 23
   c. STREET ......................................................................................................... 25
   d. MENSA ......................................................................................................... 27
   e. GLOBAL RESULTS ........................................................................................ 29
7. CONCLUSIONS ..................................................................................................... 31
8. CITATIONS ............................................................................................................. 32
9. BIBLIOGRAPHY .................................................................................................... 33
10. APENDIX .............................................................................................................. 34
    a. LIST OF RESULTS ......................................................................................... 34
    b. SCRITPS ..................................................................................................... 40
1. INTRO

Nowadays the world of acoustic fields’ simulation advances quickly, and gives us lot of chances with these breakthroughs. An obvious example is the devices and techniques which this project deals with: microphone arrays and the Wave Field Synthesis (WFS) technology. These, with the High Order Ambisonics (HOA) techniques, are the bases of this project.

Taking the recordings made with a microphone array as starting point, done inside 4 different acoustics ambiences (with a range of reverberation times and background noise). This microphone, EigenMike32 owned by M&H Acoustics, consists in 32 capsules spread in a spherical surface as the vertex of a Pentakis´ dodecahedron. The given quality is higher than the regular digital audio, since even sampling at 44.1 kHz, the bit depth is 24 instead the typical 16.

From this recordings two different codifications will be made: the horizontal take of the EigenMike, because the playback system has only two audio dimensions; and the other using the High Order Ambisonics techniques, that even they are developed for a tridimensional reproduction system with at least 2 loudspeakers arrays, the angle between them will be fixed to 0 degrees making them coincident. Those coders have been programmed in Matlab, linked to the Isophonics´ scripts known as Spatial Audio Matlab Toolbox.

The obtained audio clips will be played inside the Wave Field Synthesis System set by Iosono in the university Jade Höchscule Oldenburg. The mentioned system is made of 420 loudspeakers distributed in 2 rows, one of tweeters and the other of woofers in a ratio 3 to 1, inside a rectangular array; and 4 subwoofers placed in the corners of the rectangle. This system works at 48 kHz of sampling frequency, so an interpolation must be done inside the previous scripts, and with 24 bits of depth.

Subsequently, it will proceed to test the material from the coders comparing between both codifications and 2 other, in which ones the recordings were made with a Double MS form Schoeps, taking as reference the recordings of an artificial head (it is known that this is the closest system to our hearing way). As it can be expected, the recordings were made at the time for all systems, and also trying to make them as coincident as possible. The test were done following the statistic method known as Latin Square, in which one the closest take to the reference changes in ever round avoiding then the habit of the listener to the order of the testing samples. This program was done also in Matlab, and it stores the results of each participant of these tests inside a file that can be loaded later.

At the end, it will proceed to extract the tests’ results and to analyze them for finally obtain the conclusions of this entire project.
2. THEORETICAL FUNDATIONS

a. MICROPHONE ARRAY

An array is an ordered group of elements. In this case, a group of electro-acoustic transducers, or microphones, placed in a determinate geometry, a Pentakis’ dodecahedron (a 32 vertex regular polyhedron).

"A microphone array is any number of microphones operating in tandem. Typically, an array is made up of omni-directional microphones distributed about the perimeter of a space, linked to a computer that records and interprets the results into a coherent form. Arrays may also be formed using numbers of very closely spaced microphones. Given a fixed physical relationship in space between the different individual microphone transducer array elements, simultaneous DSP (digital signal processor) processing of the signals from each of the individual microphone array elements can create one or more "virtual" microphones." [1]

![Image of microphone array](image)

Fig 1. An array made of microphones. [2]

Those arrays are used with the goal of improving the communication between emitter and receptor; or in this case, the recreation of the surrounding ambience. So, it is supposed to make an improvement comparing to a recording made without array, only with a single microphone.
b. WAVE FIELD SYNTHESIS TECHNIQUE

“Wave field synthesis (WFS) is a spatial audio-rendering technique, characterized by creation of virtual acoustic environments. It produces "artificial" wave fronts synthesized by a large number of individually driven speakers. (...) Contrary to traditional spatialization techniques such as stereo, the localization of virtual sources in WFS does not depend on or change with the listener's position.” [3]

As it can be deduced, it is necessary an array of loudspeakers, in at least one direction, to implement this technique. Later it will see that the system owned is made of a rectangular array that consists in 420 loudspeakers set up in Jade Höchscule Oldenburg.

This technique is based on the Huygens’ Principle, which postulates that every point of the wave front turns in to a new emitting source during the next instant. This way, as it can be watched in the figure, by knowing the origin point of the source and through an array of speakers, it is possible to simulate its wave front (spherical for tri-dimensional arrays, or cylindrical in the case of bi-dimensional ones).

This technique is very promising, since as, in theory, it erases the theaters or cinemas sweet spots making any point in the room an acoustic equivalent to any other one. This is possible thanks to the generation of a uniform acoustic field inside the physic limits that close the system.
c. AMBISONICS

“Ambisonics is a series of recording and replay techniques using multichannel mixing technology that can be used live or in the studio. By encoding and decoding sound information on a number of channels, a 2-dimensional (“planar”, or horizontal-only) or 3-dimensional (“periphonic”, or full-sphere) sound field can be presented.” [5]

This Ambisonics’ reproduction technique consists of representing the acoustic field as a superposition of planar waves. Theoretically, it is possible to recreate in a perfect form the original acoustic field through an infinitive number of loudspeakers placed in a close disposition. So that, using a finite number of speakers placed in a spherical shape it is achieved a real approximation of the original acoustic field and it can be synthesized in a finite place known as the sweet spot, but with higher surface than the known conventional surrounding systems. So, the recreated sonorous field is quite different depending on the listener’s placement, achieving even to be noticed in eternal places outside the loudspeakers set. The lower limit for the needed speakers depends on the number of transmitted acoustic channels, which are defined by the Ambisonics’ approximation order. Therefore, the higher the used system order, the bigger would be the sweet spot and the higher would be the accuracy. The lowest configuration is made of 4 speakers to a bi-dimensional system and 8 for a tri-dimensional one, using the B-format codification (or First Order, the less complex).

The equation for this First Order codification is:

\[ P_n = W + X \cos \theta_n + Y \sin \theta_n \]  

(Equation 1)

Where \( P_n \) is the pressure for this speaker and, \( W \) is an omni-directional microphone input and \( X \) and \( Y \) are eight-figure microphone inputs placed perpendicularly defining a plane.

Fig 3. First Order polar patterns. [6]
In this project it is dealt with the third order; so, an explanation about the Ambisonics’ higher orders can be found next.

**i. HIGH ORDER AMBISONICS:**

“These use more channels than the original first-order B-Format to capture significantly more spatial information. At present, "real" recording techniques using them are in their infancy, it is, however, straightforward to compose synthetic recordings. Benefits include greater localization accuracy and better performance in large-scale replay environments such as performance spaces.

The higher orders correspond to further terms of the multi-pole expansion of a function on the sphere in terms of spherical harmonics. (...) In the absence of obstacles, sound in a space over time can be described as the pressure at a plane or over a sphere – and thus if one reproduces this function, one can reproduce the sound of a microphone at any point in the space pointing in any direction.” [5]

**ii. SPHERICAL HARMONICS:**

Spherical harmonics are harmonic functions that represent the spatial variation of an orthogonal solutions collection form the Laplace’s equation when these solutions are expressed by spherical coordinates.

\[
p(kr, \theta, \varphi) = \sum_{m=0}^{\infty} i^m j^n(kr) \sum_{n=-m}^{m} B_m^n Y_m^n(\theta, \varphi) \quad \text{(Equation 2)}
\]

Where \( B_m^n \) are the expansion coefficients for \( Y_m^n \) Spherical Harmonics. And \( Y_m^n \) is:

\[
Y_m^n(\theta, \varphi) = \begin{cases} 
N_m^n \cos \varphi & \text{if } n = 0, \\
\sqrt{2} N_m^n P_m^n \cos \varphi \cos n\theta & \text{if } n > 0, \\
\sqrt{2} N_m^n P_m^n \sin \varphi \sin n\theta & \text{if } n < 0,
\end{cases} \quad \text{(Equation 3)}
\]

Also, \( N_m^n \) is:

\[
N_m^n = \frac{(2m+1)(m+|n|)!}{4\pi (m-|n|)!} \quad \text{(Equation 4)}
\]

These functions give a proper way to describe deformations of a spherical surface and, therefore, can be used to simulate acoustic waves in every
outline conditions, no matter if they are free space or closed space conditions.

Fig 4. Third Order polar patterns. [7]
d. MEAN OPINION SCORE

“Mean opinion score (MOS) is a test that has been used for decades in telephony networks to obtain the human user's view of the quality of the network. Historically, and implied by the word Opinion in its name, MOS was a subjective measurement where listeners would sit in a "quiet room" and score call quality as they perceived it (...). The MOS provides a numerical indication of the perceived quality from the users’ perspective of received media after compression and/or transmission. The MOS is expressed as a single number in the range 1 to 5, where 1 is lowest perceived audio quality, and 5 is the highest perceived audio quality measurement.” [8]

e. LATIN SQUARE

“In combinatorics and in experimental design, a Latin square is an $n \times n$ array filled with $n$ different symbols, each occurring exactly once in each row and exactly once in each column.” [9]

Here, an example of how it is used in this project is given:

```
A  B  C  D
D  A  B  C
C  D  A  B
B  C  D  A
```

Fig 5. Coders’ variation in each situation.

Being each letter (A, B, C, and D) one of the different codifications that will be tested.
3. RECORDINGS

In this point, all the things that I have used during this project will be explained, together with the list of the different ambient chosen for the recordings.

All of them were made at the time with other recording devices that will not be explained because they belong to another project called 2D Audiorendering: DMS to WFS. And this is not the aim of this project.

a. EQUIPMENT

i. EM 32 EIGENMIKE

“The M&H Acoustics em32 Eigenmike® microphone array is a 32 channel spherical microphone array that uses 32 individually calibrated professional quality 14 mm electret microphones embedded in a rigid sphere baffle.” [10]

“Eigenmike® microphone array technology is a two-step process: First the outputs of the individual microphones are combined using digital signal processing to create a set of Eigenbeams. A complete set of Eigenbeams capture the sound field up to the spatial order of the beamformer. Second, the Eigenbeams are combined to steer multiple simultaneous beampatterns that can be focused to specific directions in the acoustic field.” [11]

Fig 6. Three microphone array EigeMike32. [12]

This microphone array is going to be the one that records the takes for a later treatment. The recordings were down at 44.1 kHz of sampling rate, because the manufacturer recommends this frequency to avoid some troubles in the processor, and with a number of 24 bits per sample.

With the 32 input signals it will be tried to reach the taken acoustic field inside of the WFS system.
Here is a short list of the components that sets all the system.

- EigenMike32 is the 32-microphone array. It has an output in RJ-45.
- Eigenmike Microphone Interface Box (EMIB), which takes the signal from the microphone array (RJ-45 wire) and sends them to the computer trough fire-wire.
- Mac Book Pro Windows; in this computer was installed the EigenSutidio software owned by M&H Acoustics. This laptop has a fire-wire port that receives the signal form the EMIB.

ii. DUMMY HEAD

“In acoustics, Dummy Head recording (also known as artificial head or Kunstkopf) is a method used to make binaural recordings, that allow a listener wearing headphones to perceive the directionality and the room acoustics of single or multiple sources.

Human perception of the direction of a sound source is complex, and consists of:

Simple "left-right" information can be gained from relative level differences and time of arrival differences of the sound in each ear.

For percussive sounds, the impact of a shock wave can register perceptibly on the skin (typically the upper torso), with the earliest and strongest sensory stimulus coming from the regions of skin aligned perpendicular to the direction of the sound source.

The human head imprints frequency-dependent distortions of phase and amplitude on sound reaching the eardrums, that are frequency-dependent level differences and these distortion effects vary with the direction of the sound source (being caused by the geometry and sound-transmitting characteristics of the sinus and throat cavities, Eustachian tubes, inner ear, external ears, and other hard and soft tissues in the head and upper body (see: head-related transfer function, "HRTF").

Conventional stereo recording only makes use of left-right information. Dummy head recording uses both left-right information and frequency-dependent distortions.” [13]

iii. IOSONO WAVE FIELD SYNTHESIS SYSTEM

"Iosono is the product name of an audio system presented by the Fraunhofer Institute and the Iosono GmbH in 2004. It is based on wave field synthesis, a method to use secondary audio sources to recreate primary wave fields; that was developed at Delft University of Technology in the Netherlands in the 1980s.” [14]
The system installed in the *Institut für Hörtechnik und Audiologie* is made of an amount of 420 loudspeakers forming a rectangular shape (2.8 x 2.1 m). As it can be seen on the picture below, there are two different lines set up; one for woofers, the lowest, and another one made of tweeters. The ratio between them is 3 (tweeters) to 1 (woofer), because the high frequency speakers has less power than the low and mid frequency ones. It also counts with 4 subwoofers installed in the corners of the rectangle.

![Picture of the WFS system in Jade Höchschule.](image)

*Fig 7. Picture of the WFS system in Jade Höchschule. [15]*

![Map of the installations.](image)

*Fig 8. Map of the installations. [16]*
b. SOFTWARE SUPPORT

i. MATLAB

"Matlab (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, Matlab allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran." [17]

ii. MSOUND TOOLBOX

A Matlab toolbox developed by S. Fischer, M. Hansen and J. Bitzer (Institut für Hörtechnik und Audiologie, Jade Hochschule Oldenburg), which is the way to communicate Matlab with Iosono’s hardware of the WFS system.

iii. ISOPHONICS´ SPATIAL AUDIO MATLAB TOOLBOX

“This is a Matlab toolbox containing core functions to implement spatial audio techniques. It includes Matlab functions for Ambisonics.” [18]

To implement our coders there were taken the following functions:

- AMbisonicsCF
- AMencodechannel
- AMorder
- AMplot
- AMsp_position
- AMSpeaker_channels
- AMSpeakers
c. LIST OF SITUATIONS

The list was made based on the feats of the surrounding sound atmosphere, paying attention to reverberation and ambience amount of noise. In each situation all the features will be described, but all of them have in common that the EigenMike (and all the microphone set) would be placed like a listener.

Fig 9. Recording set of microphones.

i. CHAT

A group made of 4 people just chatting in a room. It has been searched to record both acoustic fields, but the near one is the more important in this session. Here we have a few sources, and the ambience is silent enough; also, the reverb is not so long. But it had to take care about the first reflection in the table, where the speakers were sat, with a pullover.

ii. CLASS

This time, the micros were like in a student´s place (not in the geometric middle of the class). Now the source is far enough to say that it is in the far field. The reverb conditions are good, it was a room made for speech. But the first reflection in the table, where the speakers sat, had to be taken care of with a pullover.
iii. MENSA (CANTEEN)

Here the conditions got worse; the noise was uncontrollable and actually loud, linked to a quite enough long reverb time, it gave a really noisy far field. But now the speakers were close enough to the micros.

iv. STREET

At this situation the ambience was noisy, with cars and bikes passing; but because it was outside the reverb was not important at all. Now the speakers were in the far field, because the loud noise gets close to the surface limiting the near field.
4. PROGRAMMING METHOD

As it has said before, all the informatics development was done through Matlab platform.

The sets of programs already done were explained in the point Software Support. So now an explanation about the 2 different used coders will be given; as well as about the graphic user´s interface designed for the tests trying to make easy the way of the listeners´ trials. Also, something about the other scripts, which runs under those, will be done.

All the scripts can be found at the chapter called Apendix.

a. CODERS

Both coders have been made to compare the achieved results between them.

The first one picks all the input microphones up, by a script developed by Dr. Bitzer, and saves only the horizontal plane take; in this way it can be achieved an spatial recreation of the recorded situation. Also it fixes each sound-sources´ coordinates where later they will be played; the radius here is 1 (unit radius) looking to a later fixing of it through other scripts. Finally, after a re-sampling at 48 kHz (the recording were done at 44.1 kHz), it creates the .wav files that will be played and also stores the position variable for a later use in the tests.

The second one, based on High Order Ambisonics, extracts the recordings in the same way and, taking care the microphone´s position, makes the output files. Because this one is a bit more complex, it is going to be a more extended explanation.

This script starts giving the user the option to choose the lately coded file, which one should be placed in the same folder as the coder is. After that it allows that 3 features can be fixed to different values: HOA coding order and, the starting and the ending timer for the audio clip.

Fig 12. The interface to choose the codification feats.
The chosen order was the third one, but it has to be said that with the EigenMike’s takes could be reached until the fifth order. This decision was taken based on the amount of spatial aliasing heard in several tests with the sources inside the system; and, after various tests, it was decided the third order was the higher one with less grade of aliasing.

After that, and basing in the microphones’ angles, it calculates the coefficients trough the function AMbisonicsCF, to codify it later using the AMencodechannel script. This process is made for each of the 32 microphone input signals; and the resulting audio clips are saved in a folder called Trash, that should lie inside the workspace folder, or else it will be created.

The next step is to locate each one of the output sources; this is made by the function AMsp_position, which returns the positions in spherical coordinates. They will be switched to Cartesian ones, because the playing scripts are made to work with this type.

![Fig 13. Plot of the Isophonics’ Spatial Audio Matlab Tool with a large amount of speakers.](image)

Now, having the placements of each speaker and the encoded files, it will be proceeded to decode each one of those trough AMspeaker_channel.

As each input signal has given an amount of output channels (loudspeakers), it is needed to add now all these signals. This addition is done in a point to point way for each loudspeaker/output channel.

Finally, the signal is re-sampled at 48 kHz and saved in .wav files, and the variable position too, which stores the loudspeakers placement.
b. GUI AND TESTING PROGRAM

A graphic user interface (GUI) was also developed to make the listener comfortable while he/she is making the test, being this one quite simple and easy to handle with. The test results will be extracted and saved from this GUI.

To get started, the GUI asks the user to type the test number; storing in this way all the results in an unmistakable way.

Immediately after, the GUI is shown in the screen. It is made of 4 buttons (toggle-buttons) to play each one of the different situations (including 5.0 and first order Ambisonics with 12 loudspeakers, which come from the DMS recordings, together with the EigenMike32’s horizontal take and the third Ambisonics’ order codification with 16 loudspeakers).

Fig 14. The GUI appearance.

Down there, there are placed 4 sliders and 4 text boxes, each one below the situation that it scores; so it is possible, write the wished score or just to use the slider.

In the lowest place the scores range is shown, that goes from 1 (Bad) to 5 (Excellent), which is based in Mean Option Score.

In the right side there is text that points in which situation it is; and other button that allows us to go to the next round of audios.

Inside each one of these trials the audios position is changed according to the Latin Square Root statistics method. As there are 4 rounds for each situation, when the fifth is reached the name of the new situation is plotted in the top right corner.

When the fourth situation is done, the GUI thanks to the subject his participation and, with a last click, the results variable is stored for a later use.
c. OTHERS

i. RUNNING INSIDE THE CODERS

- AMspeakers, returns minimum speakers required for a given Ambisonics order.
- ReadWaveEigenMike, reads the 32 input channels from EigeMike’s recordings.
- theta_phi, loads the microphones angles.
- AMbisonicsCF, returns coefficients for each encoded channel.
- AMencodechannel, creates encoded wave files from a given input mono file.
- AMsp_position, generates speakers’ position on a sphere.
- AMplot, visualizes the position of each speaker.
- AMSpeaker_channels, generates decoded files for each speaker.

ii. RUNNING INSIDE THE GUI

- AddWFSSystemPath, adds the direction where the mBox tool is stored.
- Prueba_1, allows the user to write inside the box.
- Prueba_2, allows the user to use the slider.
- Play_Order, fixes the order and plays the audios.
- NAMES, sets the names that will be readen.
- Latin_Square, creates the reproduction order matrix.
- Play_CERO, loads the speakers position and the audios.
- Block_send, divides the audio matrix in blocks and plays them.
- counter_situation, fixes the situation and the round.

iii. RUNNING BY ITSELF

- charge_audio, loads the audio files, cut them in 1 min long, searches the maximum level and gives gain to level them.
5. LISTENING TESTS

At this point development of the tests is going to be explained, as the way in the listener makes them.

This test consists in a set of 4 audio clips, which will be played after a previous first monitoring of the reference, in which ones lies each of the 4 different codifications already named (5.0 and first order Ambisonics, from the DMS; and, 8.0 and third order Ambisonics, from the EigenMike32). The listener shall hear the reference (Dummy head) at first place and then hear the samples, one by one and following the order, evaluating them immediately after the reproduction of each one. This exercise will be repeated 4 times per situation.

As the tests try to evaluate the quality about the achieved played spatiality it is thought that the time should be long enough to allow the listeners to hear several speakers (more than 2, if there are) and variations in the background noise (for example, in the street recording it can be perceived different vehicles passing by close to the speakers). Based in this premises, the time length was fixed to 1 minute; a time long enough to induce listening fatigue in the listener, but in another way some details could be missed with shorter times.

Starting from the things said before, the listener shall make 4 repetitions of 4 situations and 5 audio clips of 1 minute, so a total of 80 minutes duration is obtained. To them should be add the times that the listener takes evaluating each sample and the loading audios time, so at the end the approximately length duration will be 90 minutes.

It thought that proper way to evaluate the tests was the MOS statistic method, giving the subject a range of values form 1, meaning the worst, until 5, being the best. Although based in it, a little variation with the original can be perceived: the variation step instead of being 1 was reduced to 0.25 to open the chosen possibilities range and to make the results more accurate in the differentiation task between them.

It must be said also that, the method for each iteration variation was based in Latin Square; through it all the samples are once in the closest position to the reference.

Now that all have been explained, there were 10 subjects that made the test, spread between men and women; and, people with knowledge about the topic and people without any one. This opened range gives us the chance to obtain results quite close to a sample of the general population.
6. **ANALYSIS OF RESULTS**

In this chapter it will proceed to analyze the results of all the tests made.

The picked sample spreads to 40 different valuations (10 persons, with 4 repetitions per situation) to evaluate each one of the coders inside each different situation. So, a detailed analysis of each situation will be done, to doing later a global evolution.

**a. CHAT**

In this situation and in all things as it will be seen later, the horizontal EigenMike32’s take codification obtains better results than the Ambisonics codification.

<table>
<thead>
<tr>
<th>8.0 EIGENMIKE HORIZONTAL TAKE</th>
<th>THIRD ORDER AMBISONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean 3.37</td>
<td>Mean 1.92</td>
</tr>
<tr>
<td>Standard Deviation 0.69</td>
<td>Standard Deviation 0.52</td>
</tr>
</tbody>
</table>

Fig 15. Table of the chat situation.

It is clearly observed that the trial of using high order of Ambisonics (HOA) did not work as the expected way; making even that the horizontal take, which was only put inside the system without any process, has higher quality.

Fig 16. Graph of the results in the chat situation for the horizontal take coder.
It is needed to remark the fact that the horizontal codification achieves results lightly over Good, while the Ambisonics’ third order only obtains Poor.
b. CLASS

Here, as it can be expected, the results are quite similar to the ones in the previous situation because the ambient conditions (background noise and reverberation time) are virtually the same; the distance to the speaker/speakers is the only thing that slightly changes.

Also should be said that this one is the only take recorded in German, and not all the people that make the test have knowledge enough of the language, causing in this way more tiredness because they are not able to understand the broadcast message.

<table>
<thead>
<tr>
<th>8.0 EIGENMIKE HORIZONTAL TAKE</th>
<th>THIRD ORDER AMBISONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>3.31</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Fig 18. Table of the class situation.

It can be caused by all the stuff form before, language switching and the distance to the source, that the results go a bit down and the dispersion of them rises; but both marks keep on the values Good for 8.0 Horizontal take and Poor for the third Ambisonics’ order.

Fig 19. Graph of the results in the class situation for the horizontal take coder.
It can be perceived in the graphics, that the answering patron is quite similar between these situations, which give some safety about the obtained data and the way in which the test were made.
c. STREET

In this situation, a little decrease in the tests’ scores is observed; being a plausible cause the rise of the background noise to elevated values (a street in which one, cars, buses and motorbikes pass by; also regular bikes pass by closer to the speakers). Although the data dispersion is lower, suggesting that the listener’s opinion is more clear, this can be caused by the induced hearing fatigue in the subject.

<table>
<thead>
<tr>
<th>8.0 EIGENMIKE HORIZONTAL TAKE</th>
<th>THIRD ORDER AMBISONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>3.20</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Fig 21. Table of the street situation.

It should be said also that this one was the take with more load meaning spatiality, since the before named sources were on movement in the same way but in both directions. Because of that, it can be thought that this one is the fairest system valuation, since it includes sources’ movement, fact that does not exist in the rest of the situations.

Fig 22. Graph of the results in the street situation for the horizontal take coder.

In this picture, belonging to the horizontal EigenMike´s take, a change in the answering patron can be observed, which suggests that the sources movement causes some phase distortion in the reception levels that later, in the playback, are not adjusted to the typical human hearing way. The microphone-array dimensions are lower than the human head ones, and also it counts with 8 reception points (microphones) instead of 2 ears.
Fig 23. Graph of the results in the street situation for the Ambisonics coder.

The Ambisonics’ third order codification still remains in Poor, but this it will be commented later in the Global Results and Conclusions points.
d. MENSA

The last situation shows a rise in the scores, even being this one the noisiest and the one with more influence meaning the reverberation time. The causes can be, as much as the listener’s hearing fatigue after an hour of test, as a better response to noisy and reverberant ambient.

<table>
<thead>
<tr>
<th>8.0 EIGENMIKE HORIZONTAL TAKE</th>
<th>THIRD ORDER AMBISONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>3.49</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Fig 24. Table of the Menza situation results.

Here can be seen that the Ambisonics´ score is the highest of all achieved until now (beating a bit the Poor score, 2), this can be because in this situation the recreated acoustic field was more uniform and the information belonging to this comes from everywhere instead of being focused in a part of it.

Fig 25. Graph of the results in the Menza situation for the horizontal take coder.
As it can be expected, the answering patrons change also for this situation, all the before earlier facts are possible causes to this. But the induced hearing fatigue after an hour and a half of test, linked to a high number of repetitions and to a big length of the audio clips, is the more plausible. Or also, the listener´s wish to end the test.
e. GLOBAL RESULTS

After the evaluation of the different situations, a global evaluation for each codification lefts will be done below.

First, the EigenMike’s horizontal take (8.0).

<table>
<thead>
<tr>
<th>8.0 EIGENMIKE HORIZONTAL TAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (Chat)</td>
</tr>
<tr>
<td>Mean (Class)</td>
</tr>
<tr>
<td>Mean (Street)</td>
</tr>
<tr>
<td>Mean (Mensa)</td>
</tr>
</tbody>
</table>

Fig 27. Table with the horizontal take coder results.

From on the top board, it can be said that it responds quite well in all situations, the surrounding conditions not having any effect, like the background noise and the reverberation time. The achieved score, 3.34, places it a bit over the reference (Dummy Head, 3); but it is still not a significant quality rising. It is worth mentioning, that this codification can be played in any system made of 8 speakers or more, while the Dummy Head´s takes are made only for being listened with headphones; this allows that more listeners could take part in a simultaneous hearing.

The HOA codification will be evaluated next.

<table>
<thead>
<tr>
<th>THIRD ORDER AMBISONICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (Chat)</td>
</tr>
<tr>
<td>Mean (Class)</td>
</tr>
<tr>
<td>Mean (Street)</td>
</tr>
<tr>
<td>Mean (Mensa)</td>
</tr>
</tbody>
</table>

Fig 1. Table with the Ambisonics coder results.

The 1.90 score leaves this codification lightly under Poor (2), which means that it is not better that the Dummy Head´s recordings taken as reference. Here, several ideas will be commented about why this coder does not work as it was expected.

First, as it has already said, the Isophonics´ Spatial Audio Matlab Toolbox scripts were designed for a spherical reproduction system, while here it had...
a rectangular array. So, here there are 2 problems: the change from a tri-dimensional to bi-dimensional one, and the switch from a sphere shape (circular) to a rectangular one. It should be emphasized that, for the dimensional problem, the angle between the rings was fixed to 0 degrees making them to coincide in the space reducing in this way the required array to a 2 dimensional one.

Also, it has to take account of the spatial aliasing caused by the amount of loudspeakers involved in the reproduction, 16. The aliasing was the reason to reduce the order to 3 instead of 5, which was the highest achievable, because little fluctuations happened in the pressure level inside the hearing space. It looks like this problem is happening still.

For the end, the spatial placement of the emitting sources inside the system (the loudspeakers array) instead of outside linked to all the already said ideas can be possible causes to the low score obtained by the Ambisonics coder.
7. CONCLUSIONS

The goal of this project was to test the EigenMike32 multi-microphone array recordings inside the Iosono’s Wave Field Synthesis system.

All the development can be evaluated as positive, the coders work properly, as it does the GUI, and the tests ran without any significant trouble.

Now, evaluating each coder, the same thing cannot be said.

HOA is a technique where the future of multichannel audio may lie, but this time the results point to a different direction. Several causes of this behavior have been already said, but here is a list of them:

- Spatial aliasing.
- Emitting sources inside the array instead of out of it.
- Bi-dimensional array instead of tri-dimensional one.
- Possible errors in the codification script.

The other codification, the simplest without doubt, has shown better quality to the listeners. But it should be said that, although its performance is well, 24 input signals are wasted. If the available array was tri-dimensional all the microphones could be used.

In conclusion, Ambisonics should not be abandoned only because this time it has not worked as it expected; but large skills are needed to transform all the material to only 2 dimensions. The horizontal take can be used for cases of voice and space location as it was supposed; it advised 8 speakers, to 8 microphones, surrounding the microphone array.

All the development in this project has meant a forward step to a better use of HOA and WFS techniques inside the system of Institut für Hörtechnik und Audiologie, Jade Höchscule Oldenburg.
8. CITATIONS

[18] http://www.isophonics.net/content/spatial-audio-matlab-toolbox
[19] Spatial Audio Toolbox v.1.0, Documentation, Ambisonics
9. **BIBLIOGRAPHY**

- Dave Malham, "High order Ambisonics systems", University of New York, April 2003.
- L. Scopece, A. Farina and A. Capra, "360 degrees video and audio recording and broadcasting employing a parabolic mirror camera and a spherical 32-capsules microphonearray", University of Parma, 2011.
- MH Acoustics, "em32 Eigenmike - microphone array release notes".
- Patricia Casado, "Simulation of sound wave propagation in ducts by the finite-difference time-domain method", University of Southern Denmark June 2011.
- T. Okamoto, Y. Iwaya, S. Sakamoto and Y. Suzuki, "Implementation of higher order Ambisonics recording array with 121 microphones", Tohoku University, Japan, October 2010.
## 10. APPENDIX

### a. LIST OF RESULTS

#### i. CHAT

<table>
<thead>
<tr>
<th>8.0 Eigen</th>
<th>3th Ambi</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,35</td>
<td>1,92</td>
</tr>
<tr>
<td>3,67</td>
<td>1,74</td>
</tr>
<tr>
<td>2,45</td>
<td>1,50</td>
</tr>
<tr>
<td>2,87</td>
<td>1,56</td>
</tr>
<tr>
<td>2,50</td>
<td>1,75</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>2,50</td>
<td>2,00</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>3,00</td>
</tr>
<tr>
<td>2,75</td>
<td>2,25</td>
</tr>
<tr>
<td>4,00</td>
<td>2,00</td>
</tr>
<tr>
<td>1,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,25</td>
</tr>
<tr>
<td>3,00</td>
<td>1,25</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>3,58</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
</tbody>
</table>

2D Audiorendering. David Peña Gil.  
Doktor Jörg Bitzer. Institut für Hörtechnik+Audiologie.
### ii. CLASS

<table>
<thead>
<tr>
<th>8.0 Eigen</th>
<th>3th Ambi</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,35</td>
<td>1,92</td>
</tr>
<tr>
<td>3,67</td>
<td>1,74</td>
</tr>
<tr>
<td>2,45</td>
<td>1,50</td>
</tr>
<tr>
<td>2,87</td>
<td>1,56</td>
</tr>
<tr>
<td>2,50</td>
<td>1,75</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>2,50</td>
<td>2,00</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>3,00</td>
</tr>
<tr>
<td>2,75</td>
<td>2,25</td>
</tr>
<tr>
<td>4,00</td>
<td>2,00</td>
</tr>
<tr>
<td>1,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,25</td>
</tr>
<tr>
<td>3,00</td>
<td>1,25</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>3,58</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>4,26</td>
<td>1,83</td>
</tr>
<tr>
<td>3,55</td>
<td>1,83</td>
</tr>
<tr>
<td>3,93</td>
<td>1,80</td>
</tr>
<tr>
<td>4,29</td>
<td>2,19</td>
</tr>
<tr>
<td>4,00</td>
<td>2,00</td>
</tr>
<tr>
<td>3,00</td>
<td>3,00</td>
</tr>
<tr>
<td>3,00</td>
<td>3,00</td>
</tr>
<tr>
<td>2,00</td>
<td>1,75</td>
</tr>
<tr>
<td>3,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>2,25</td>
<td>1,00</td>
</tr>
<tr>
<td>2,25</td>
<td>1,00</td>
</tr>
<tr>
<td>3,75</td>
<td>2,00</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,25</td>
</tr>
<tr>
<td>3,75</td>
<td>2,00</td>
</tr>
</tbody>
</table>
### iii. STREET

<table>
<thead>
<tr>
<th>8.0 Eigen</th>
<th>3rd Ambi</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,50</td>
<td>1,50</td>
</tr>
<tr>
<td>4,50</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>2,81</td>
<td>1,53</td>
</tr>
<tr>
<td>3,25</td>
<td>1,41</td>
</tr>
<tr>
<td>2,93</td>
<td>1,36</td>
</tr>
<tr>
<td>2,87</td>
<td>1,59</td>
</tr>
<tr>
<td>3,90</td>
<td>1,74</td>
</tr>
<tr>
<td>3,04</td>
<td>1,92</td>
</tr>
<tr>
<td>3,84</td>
<td>1,89</td>
</tr>
<tr>
<td>3,73</td>
<td>1,68</td>
</tr>
<tr>
<td>4,00</td>
<td>1,25</td>
</tr>
<tr>
<td>3,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,00</td>
<td>1,75</td>
</tr>
<tr>
<td>3,75</td>
<td>2,25</td>
</tr>
<tr>
<td>2,00</td>
<td>1,00</td>
</tr>
<tr>
<td>2,75</td>
<td>1,00</td>
</tr>
<tr>
<td>3,00</td>
<td>1,00</td>
</tr>
<tr>
<td>2,00</td>
<td>2,00</td>
</tr>
<tr>
<td>2,75</td>
<td>1,75</td>
</tr>
<tr>
<td>4,25</td>
<td>2,25</td>
</tr>
<tr>
<td>2,25</td>
<td>1,50</td>
</tr>
</tbody>
</table>
iv. MENSA

<table>
<thead>
<tr>
<th>8.0 Eigen</th>
<th>3rd Ambi</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,50</td>
<td>1,50</td>
</tr>
<tr>
<td>4,00</td>
<td>1,50</td>
</tr>
<tr>
<td>3,43</td>
<td>2,69</td>
</tr>
<tr>
<td>3,37</td>
<td>1,65</td>
</tr>
<tr>
<td>3,19</td>
<td>1,47</td>
</tr>
<tr>
<td>3,64</td>
<td>1,47</td>
</tr>
<tr>
<td>3,25</td>
<td>2,75</td>
</tr>
<tr>
<td>3,00</td>
<td>2,75</td>
</tr>
<tr>
<td>2,00</td>
<td>2,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,00</td>
</tr>
<tr>
<td>4,25</td>
<td>2,25</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>4,00</td>
<td>3,50</td>
</tr>
<tr>
<td>4,00</td>
<td>3,00</td>
</tr>
<tr>
<td>3,50</td>
<td>2,00</td>
</tr>
<tr>
<td>3,00</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>3,00</td>
</tr>
<tr>
<td>4,00</td>
<td>3,00</td>
</tr>
<tr>
<td>3,75</td>
<td>3,00</td>
</tr>
<tr>
<td>4,50</td>
<td>2,50</td>
</tr>
<tr>
<td>4,00</td>
<td>2,50</td>
</tr>
<tr>
<td>3,00</td>
<td>3,00</td>
</tr>
<tr>
<td>4,50</td>
<td>2,50</td>
</tr>
<tr>
<td>3,79</td>
<td>1,41</td>
</tr>
<tr>
<td>3,93</td>
<td>1,74</td>
</tr>
<tr>
<td>3,61</td>
<td>1,62</td>
</tr>
<tr>
<td>3,76</td>
<td>1,71</td>
</tr>
<tr>
<td>2,00</td>
<td>1,00</td>
</tr>
<tr>
<td>2,00</td>
<td>1,00</td>
</tr>
<tr>
<td>3,00</td>
<td>1,00</td>
</tr>
<tr>
<td>3,00</td>
<td>1,00</td>
</tr>
<tr>
<td>2,25</td>
<td>1,00</td>
</tr>
<tr>
<td>4,25</td>
<td>1,00</td>
</tr>
<tr>
<td>3,50</td>
<td>1,00</td>
</tr>
<tr>
<td>3,25</td>
<td>1,00</td>
</tr>
<tr>
<td>3,75</td>
<td>2,75</td>
</tr>
<tr>
<td>3,25</td>
<td>2,50</td>
</tr>
<tr>
<td>3,50</td>
<td>2,50</td>
</tr>
<tr>
<td>3,25</td>
<td>2,50</td>
</tr>
</tbody>
</table>
b. SCRIPTS

i. HORIZONTAL TAKE CODER

% Horizontal take script
% David Peña Gil
%
% Takes the EigenMike’s recordings and extracts the mics of the % horizontal plane
%
clear all
close all
clc

% Creates the constants

fs=44100;  % sample rate
fs2=48000;  % resampling rate
SoundSpeed=340.25;  % speed of sound
num_mics=8;

situation=input('Choose a situation: 1.Street   2.Class   3.Chat  
4.Mensa   ');

switch situation
    case 1
        filename = '8.0_EMK_str';
    case 2
        filename = '8.0_EMK_cls';
    case 3
        filename = '8.0_EMK_cht';
    case 4
        filename = '8.0_EMK_men';
end

% MENUS

[szFileName , path]  = uigetfile( {
    '*.wav', ...
    'Audio data ( wav )'}, ...
    'Open audiofiles');
dlg_title='TIME FEATS';
prompt={
    'Starting time (in seconds): ',
    'Ending time (in seconds): '};
num_lines=1;
def={
    '15',
    '76'};
answer=inputdlg(prompt,dlg_title,num_lines,def);
tmp_a=str2double(answer{1});

if tmp_a<=0
    TMP_A=1;  % Conversion to samples
else

2D Audiorendering. David Peña Gil.
Doktor Jörg Bitzer. Institut für Hörtechnik+Audiologie.
\[ TMP_A = \text{tmp}_a * \text{fs}; \quad \% \text{Conversion to samples} \]
end
\[ \text{tmp}_b = \text{str2double(\text{answer[2]});} \]
\[ TMP_B = \text{tmp}_b * \text{fs}; \quad \% \text{Conversion to samples} \]
\[ \text{Length} = \text{TMP}_B - \text{TMP}_A; \quad \% \text{Clip's length in samples} \]

\% Reads the EigenMike file
\[
[\text{Data}, \text{stWaveInfo}] = \text{ReadWaveEigenMike(szFileName, [\text{TMP}_A \text{ TMP}_B]);}
\]
\% Picks the mics 11 4 2 7 27 20 18 23
\[
\text{clips} = \text{[Data(:,2) Data(:,4) Data(:,7) Data(:,11) Data(:,18) Data(:,20) Data(:,23) Data(:,27)]};
\]
\% Saves 8 different audiofiles
\[
\text{for} \quad j = 1: \text{num_mics} \\
\quad \quad \text{clip} = \text{resample(clips(j,:),fs2,fs)}; \\
\quad \quad \text{p} = \text{strcat(int2str(j),'.wav')}; \\
\quad \quad \text{filename1} = \text{strcat(filename,p)}; \\
\quad \quad \text{wavwrite(clip,fs2,filename1)}
\]
\% Sets the angles
\[
\text{phi} = [32 328 69 291 212 148 249 111]; \\
\text{position} = \text{zeros(8,2)};
\]
\[
\text{for} \quad i = 1: \text{num_mics} \\
\quad \quad \text{position}(i,2) = \cos(\text{phi}(i)*2*\pi/360); \\
\quad \quad \text{position}(i,1) = -\sin(\text{phi}(i)*2*\pi/360);
\]
\% Saves the workspace and the position variable
\[
\text{savefile} = \text{strcat(filename,'.mat')}; \\
\text{save(savefile, 'position');}
\]
\% End

\textbf{ii. HIGH ORDER AMBISONICS CODER}

\% High Order Ambisonics script
\% David Peña Gil
\%
\% Takes the EigenMike's recordings and codifies them in High order Ambisonics
\%
\text{close all}
\text{clear all}
\text{clc}

\% Creates the constants
\[
\text{filename} = '\text{Amb}_\text{EMK}_\text{str'};
\]
fs=44100; % sampling rate  
fs2=48000; % resampling rate  
SoundSpeed=340.25; % speed of sound  
num_mics=32;  
switch situation  
case 1  
    filename = 'Amb_EMK_str';  
case 2  
    filename = 'Amb_EMK_cls';  
case 3  
    filename = 'Amb_EMK_cht';  
case 4  
    filename = 'Amb_EMK_men';  
end  
% MENSUS  
[szFileName, path] = uigetfile( { '*.wav' }, ...  
    'Audio data ( wav )'), ...  
    'Open audiofiles');  
dlg_title='SETTINGS AMBISONIC´S HIGH ORDER FEATS';  
prompt={'Order (max. 5): ', 'Starting time (in seconds): ', 'Ending time (in seconds): '};  
num_lines=1;  
def={'3','6','67'};  
answer=inputdlg(prompt,dlg_title,num_lines,def);  
order=str2double(answer{1});  
tmp_a=str2double(answer{2});  
if tmp_a<=0  
    TMP_A=1; % Conversion to samples  
else  
    TMP_A=tmp_a*fs; % Conversion to samples  
end  
tmp_b=str2double(answer{3});  
TMP_B=tmp_b*fs; % Conversion to samples  
Length=TMP_B-TMP_A; % Length in samples  
num_speakers=AMspeakers(order); % returns the number of speakers needed for given ambisonics order  
% Reads the EigenMike file  
[Data, stWaveInfo] = ReadWaveEigenMike(szFileName, [TMP_A TMP_B]);  
% Sets the input angles  
[theta, phi]=theta_phi;  
% Codifies the signals  
for i=1:num_mics  
    coef_sp_array=AMbisonicsCF(phi(i),theta(i),order);
% Generates the coefficients for the encoding
p=strcat(int2str(i),"_");
filename_A='Trash\enc_'; % setting filename for encoded channels
filename_B=strcat(path,filename_A, p);
AMencodechannel(Data(:,i),fs,coef_sp_array,num_speakers,
filename_B); % generating encoded channels
end

% Locates the speakers
speaker_array_azimuth=[];
speaker_array_elevation=[];

[speaker_array_azimuth,speaker_array_elevation]=AMsp_position(order,1,
0.01,0); % Generates the speaker elevation and azimuth angles.
position=zeros(num_speakers,2);
for g=1:num_speakers % Changes between Spherical to Cardinal coordinates
    position(g,2)=cos(speaker_array_azimuth(g)*2*pi/360);
    position(g,1)=-sin(speaker_array_azimuth(g)*2*pi/360);
end
figure; AMplot(speaker_array_azimuth,speaker_array_elevation,0); % Plot speakers location

% Decodifies the coded signals
for j=1:num_mics
    p=strcat(int2str(j),"_");
    filename2=strcat(filename_A, p);
    filename_C='Trash\decsp_';
    % setting file name for output channels
    output_f=strcat(path,filename_C, p);
    AMSpeaker_channels(order,speaker_array_azimuth,
speaker_array_elevation,filename2,output_f);
    % generates decoded channels and save the files.
end

% adds the signals of each speaker
speaker=zeros(num_speakers,Length);
for k=1:num_speakers
    d=strcat(int2str(k));
    for l=1: num_mics
        p=strcat(int2str(l),"_");
        filename4=strcat(filename_C, p,d);
        sp=wavread(filename4);
        sp=sp';
        if l==1
            speaker(k,:)=sp(1,:);
        else
            for j=1:Length
                speaker(k,j)=speaker(k,j)+sp(1,j);
            end
        end
    end
end
% Saves the audiofiles

% for j=1:num_speakers
spkr=resample(speaker(j,:),fs2,fs);
p=strcat(int2str(j),'.wav');
filename1=strcat(filename,p);
wavwrite(spkr,fs2,filename1);
end
% Saves the workspace's position variable

savefile = strcat(filename,'.mat');
save(savefile, 'position');

% End

iii. GUI

% Call: Main
% Creates all the gui, 4 "play" buttons for the situations, 4 sliders and text boxes, the Next button and shows the scale valuation values
% Author: Sally Castro and David Peña
% Date: 7.08.2012

clear all
close all
clc

AddWFSSystemPath % Adds the location of the mSound tool

% Variables declaration
mvar1=3;
mvar2=3;
mvar3=3;
mvar4=3;

options=4;

counter=1;
situation=1;

result=zeros(4,4);
mute_array = ones(options,1);

name=input('N° Test: ');
name=strcat(name);
% Plots the range of values

scale1=uicontrol('style','text','string','1 = BAD', 'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[0 0 0], 'position', [20 150 70 20]);
scale2=uicontrol('style','text','string','2 = POOR', 'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[0 0 0], 'position', [130 150 70 20]);
scale3=uicontrol('style','text','string','3 = FAIR', 'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[0 0 0], 'position', [240 150 70 20]);
scale4=uicontrol('style','text','string','4 = GOOD', 'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[0 0 0], 'position', [350 150 70 20]);
scale5=uicontrol('style','text','string','5 = EXCELLENT', 'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[0 0 0], 'position', [460 150 70 20]);

% Creates the slidders and the text boxes

u2=uicontrol('style','edit','string',mvar1,'backgroundColor',[1 1 1], 'foregroundColor',[0 0 0], 'position', [60 270 60 16]);
u3=uicontrol('style','slider','value',mvar1,'backgroundColor',[0.7 0.7 0.7], 'foregroundColor',[1 1 1], 'position', [50 180 2020], 'Max',5, 'Min',1, 'SliderStep',[0.0625 0.0625]);
u4=uicontrol('style','edit','string',mvar2,'backgroundColor',[1 1 1], 'foregroundColor',[0 0 0], 'position', [160 270 60 16]);
u5=uicontrol('style','slider','value',mvar2,'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[1 1 1], 'position', [150 180 2020], 'Max',5, 'Min',1, 'SliderStep',[0.0625 0.0625]);
u6=uicontrol('style','edit','string',mvar3,'backgroundColor',[1 1 1], 'foregroundColor',[0 0 0], 'position', [260 270 60 16]);
u7=uicontrol('style','slider','value',mvar3,'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[1 1 1], 'position', [250 180 2020], 'Max',5, 'Min',1, 'SliderStep',[0.0625 0.0625]);
u8=uicontrol('style','edit','string',mvar4,'backgroundColor',[1 1 1], 'foregroundColor',[0 0 0], 'position', [360 270 60 16]);
u9=uicontrol('style','slider','value',mvar4,'backgroundColor',[0.8 0.8 0.8], 'foregroundColor',[1 1 1], 'position', [350 180 2020], 'Max',5, 'Min',1, 'SliderStep',[0.0625 0.0625]);

% Sets the action of the buttons

set(u2, 'callback', 'mvar1=Prueba_1(u2,u3)');
set(u3, 'callback', 'mvar1=Prueba_2(u2,u3)');
set(u4, 'callback', 'mvar2=Prueba_1(u4,u5)');
set(u5, 'callback', 'mvar2=Prueba_2(u4,u5)');
set(u6, 'callback', 'mvar3=Prueba_1(u6,u7)');
set(u7, 'callback', 'mvar3=Prueba_2(u6,u7)');
set(u8, 'callback', 'mvar4=Prueba_1(u8,u9)');
set(u9, 'callback', 'mvar4=Prueba_2(u8,u9)');
% Creates the playback buttons

hbo1 = uicontrol('style',
'togglebutton', 'String', 'A', 'ForegroundColor',[1 1 1], 'BackgroundColor',[0 0 0], 'Position',[10 390 100 20], 'FontSize',12);
hbo2 = uicontrol('style',
'togglebutton', 'String', 'B', 'ForegroundColor',[1 1 1], 'BackgroundColor',[0 0 0], 'Position',[110 390 100 20], 'FontSize',12);
hbo3 = uicontrol('style',
'togglebutton', 'String', 'C', 'ForegroundColor',[1 1 1], 'BackgroundColor',[0 0 0], 'Position',[210 390 100 20], 'FontSize',12);
hbo4 = uicontrol('style',
'togglebutton', 'String', 'D', 'ForegroundColor',[1 1 1], 'BackgroundColor',[0 0 0], 'Position',[310 390 100 20], 'FontSize',12);

% Sets the action of the buttons

set(hbo1, 'callback', 'if Play_Order(counter,situation,1); else (get(gcbo, ''Value'')==1), mute_array(1)=0; end')
set(hbo2, 'callback', 'if Play_Order(counter,situation,2); else (get(gcbo, ''Value'')==1), mute_array(1)=0; end')
set(hbo3, 'callback', 'if Play_Order(counter,situation,3); else (get(gcbo, ''Value'')==1), mute_array(1)=0; end')
set(hbo4, 'callback', 'if Play_Order(counter,situation,4); else (get(gcbo, ''Value'')==1), mute_array(1)=0; end')

% Shows the results variable
res=[mvar1 mvar2 mvar3 mvar4]

% Creates the Next button

next_button = uicontrol('style', 'togglebutton', 'Position', [ 440 250 100 60], 'String','NEXT TRIAL', 'ForegroundColor',[0 0 0]);

% Sets the action of the button

set(next_button, 'callback','res=[mvar1 mvar2 mvar3 mvar4];result=saving(name,counter,situation,res,result);if(situation~=
5),[counter, situation]=counter_situation(counter,situation);else, close all,end');

% END

iv. OTHERS

a. AddWFSSystemPath

addpath('F:\Hoertechnik_Audiologie\IHA_Wellenfeldsynthese\matlab_wfs')

%msound('DeviceInfo');
b. charge_audio

```matlab
function charge_audio(filename)
%Call: charge_audio(filename)
%Load filename, cut the matrix audio in 1 minute long, search de max
%SPL level, gave one gain g and save the filename.
%
% Author: Sally Castro
% Date:3.08.2012

M_A=0.5
file=strcat(filename, '.mat');
load(file, 'position');
num_speakers=length(position);
for i=1:num_speakers
    p=strcat(int2str(i), '.wav');
    filename1=strcat(filename, p)
    audio(:,i)= wavread(filename1);
    sourcetype(:,i)=0;
    audio2(:,i)= audio(1:2928000,i);
end
M_B=MAX_SPL(audio);
g=M_A/M_B;
audio2=audio2*g;
for i=1:num_speakers
    p=strcat(int2str(i), '.wav');
    filename1=strcat(filename, p)
    wavwrite(audio2(:,i), filename1);
end
```

c. counter_situation

```matlab
function [counter, situation]=counter_situation (counter, situation)
%Call: [counter, situation]=counter_situation (counter, situation)
% Controls what is the test situation.
% % Author: Sally Castro
% % Date:3.08.2012

if counter==4
    situation=situation+1
```
counter=1
scal=uicontrol('style','text','string',situation,'backgroundcolor',[0.8 0.8 0.8], 'foregroundcolor',[0 0 0], 'position', [20 50 70 20]);
else
    counter=counter+1
    scal=uicontrol('style','text','string',counter,'backgroundcolor',[0.8 0.8 0.8], 'foregroundcolor',[0 0 0], 'position', [90 50 70 20]);
end

switch situation
    case 1
        text_sit='CHAT'
    case 2
        text_sit='CLASS'
    case 3
        text_sit='SREET'
    case 4
        text_sit='MENSA'
    otherwise
        text_sit='END Thanks!'
end
txt_sit=uicontrol('style','text','string',text_sit,'backgroundcolor',[0.8 0.8 0.8], 'position', [440 330 100 60], 'fontsize',16);

function N=Latin_Square(M)
    A=M(1,1:11)
    B=M(1,12:22)
    C=M(1,23:33)
    D=M(1,34:44)
    N=[A B C D ; B C D A; C D A B; D A B C; A B C D];
end

d. Latin_Square

e. MAX_SPL

function Maximo=MAX_SPL(A)
    [a,b]=size(A);
    suma=zeros(a,1);
    for i=1:b
suma(:,1)=suma(:,1)+(A(:,i).^2);
end

Maximo=sqrt(max(max(suma)));

f. NAMES

function N= NAMES(situation)

%Call: NAMES
%Introduce all the data audio names in a matrix order by Latin_Square
% function
% Author: Sally Castro
% Date: 1.08.2012

S=11;

switch situation
    case 1,
        M(1,1:S) = '5.0_DMS_cht';
        M(1,1*S+1:1*S+S) = 'Amb_DMS_cht';
        M(1,2*S+1:2*S+S) = '8.0_EMK_cht';
        M(1,3*S+1:3*S+S) = 'Amb_EMK_cht';
        N=Latin_Square(M)
    case 2,
        M(1,1:S) = '5.0_DMS_cls';
        M(1,1*S+1:1*S+S) = 'Amb_DMS_cls';
        M(1,2*S+1:2*S+S) = '8.0_EMK_cls';
        M(1,3*S+1:3*S+S) = 'Amb_EMK_cls';
        N=Latin_Square(M)
    case 3,
        M(1,1:S) = '5.0_DMS_str';
        M(1,1*S+1:1*S+S) = 'Amb_DMS_str';
        M(1,2*S+1:2*S+S) = '8.0_EMK_str';
        M(1,3*S+1:3*S+S) = 'Amb_EMK_str';
        N=Latin_Square(M)
    case 4,
        M(1,1:S) = '5.0_DMS_men';
        M(1,1*S+1:1*S+S) = 'Amb_DMS_men';
        M(1,2*S+1:2*S+S) = '8.0_EMK_men';
        M(1,3*S+1:3*S+S) = 'Amb_EMK_men';
        N=Latin_Square(M)
    otherwise display ('fin');
end

M(1,1:S) = '5.0_DMS_str';

g. Play_CERO

function Play_CERO(filename)
%Call: Play_CERO(filename)
%Open filename (.mat and all the speakers signals .wav) and send the
%audio and positions to Block_send function
% Author: Sally Castro
% Date: 21.07.2012

AddWFSSystemPath

msound('close')

file=strcat(filename,'.mat');
load(file,'position');

num_speakers=length(position);

%Open channels
for i=1:num_speakers
  p=strcat(int2str(i),'.wav');
  filename1=strcat(filename,p);
  audio(:,i)=wavread(filename1);
  sourcetype(:,i)=0;
end

posfs = 20;

% play radius
R = 2;
position=(position*R)

%wfssound(audio,audiofs,position(:,1),position(:,2),posfs,sourcetype)
Block_send (audio,position, num_speakers);

h. Play_Order

function counter = Play_Order (counter,situation,choose)

%M=11;
switch counter
  case 1
    Mc=M(1,:)
    Mc(1,2*S+1:2*S+S)
    switch choose
      case 1, play_CERO(Mc(1,1:S))
      case 2, play_CERO(Mc(1,1*S+1:1*S+S))
case 3, \texttt{play\_CERO}(\texttt{Mc}(1,2*S+1:2*S+S))
\textbf{end}
case 4, \texttt{play\_CERO}(\texttt{Mc}(1,3*S+1:3*S+S))
\textbf{end}
case 2,
\texttt{Mc} = \texttt{M}(2, :)
\textbf{switch} \texttt{choose}
\textbf{case} 1, \texttt{play\_CERO}(\texttt{Mc}(1,1:S))
\textbf{case} 2, \texttt{play\_CERO}(\texttt{Mc}(1,1*S+1:1*S+S))
\textbf{case} 3, \texttt{play\_CERO}(\texttt{Mc}(1,2*S+1:2*S+S))
\textbf{case} 4, \texttt{play\_CERO}(\texttt{Mc}(1,3*S+1:3*S+S))
\textbf{end}
case 3
\texttt{Mc} = \texttt{M}(3, :)
\textbf{switch} \texttt{choose}
\textbf{case} 1, \texttt{play\_CERO}(\texttt{Mc}(1,1:S))
\textbf{case} 2, \texttt{play\_CERO}(\texttt{Mc}(1,1*S+1:1*S+S))
\textbf{case} 3, \texttt{play\_CERO}(\texttt{Mc}(1,2*S+1:2*S+S))
\textbf{case} 4, \texttt{play\_CERO}(\texttt{Mc}(1,3*S+1:3*S+S))
\textbf{end}
case 4
\texttt{Mc} = \texttt{M}(4, :)
\textbf{switch} \texttt{choose}
\textbf{case} 1, \texttt{play\_CERO}(\texttt{Mc}(1,1:S))
\textbf{case} 2, \texttt{play\_CERO}(\texttt{Mc}(1,1*S+1:1*S+S))
\textbf{case} 3, \texttt{play\_CERO}(\texttt{Mc}(1,2*S+1:2*S+S))
\textbf{case} 4, \texttt{play\_CERO}(\texttt{Mc}(1,3*S+1:3*S+S))
\textbf{end}
\textbf{otherwise}
\textbf{end}
\textbf{end}

\section{Prueba\_1}

\begin{verbatim}
function mvar=Prueba\_1(u,v)

%Call: Prueba\_1(u,v)
% Loads the text box and the slider, allows the user to write inside
% the box and then, reads the value inside the text box and returns it
% for a later storage inside the main script.
%
% Author: David Peña
% Date:4.08.2012

f=get(u, 'parent');
mvar= str2num(get(u, 'string'));

if mvar>=1 && mvar<=5
    set(v, 'value',mvar);
    set(u, 'value',mvar);
else
    mvar=get(u, 'value');
    set(u, 'string',num2str(mvar));
    set(u, 'value',mvar);
end
\end{verbatim}
j. Prueba_2

function mvar=Prueba_2(u,v)

%Call: Prueba_2(u,v)
% Loads the text box and the slider, allows the user to write inside
% the box and then, reads the value of the slider box and returns it
% for a later storage inside the main script.

% Author: David Peña
% Date:5.08.2012

f=get(v,'parent');
mvar=get(v,'value');

set(u,'string',num2str(mvar));
set(u,'value',mvar);

drawnow
end

k. saving

function result =saving (name, counter,situation,res,result)

%Call: result =saving (name, counter,situation,res,result)
%Save the results of a test when the situation has exceeded 5. res is
%an array
%containing the results of a situation relying on the counter, these
%results
%are being stored in the array result, once this whole stored in a
%file. mat for later reading.

% Author: Sally Castro
% Date:3.8.2012

if situation==1
    result(counter,:)=res
else
    result((situation-1)*4+counter,:)=res
end

if situation==5
    filename=strcat(int2str(name),'.mat');
save(filename, 'result');
end
1. theta_phi

function [theta, phi]=theta_phi

theta=[69 90 111 90 32 55 90 125 148 125 90 55 21 58 121 159 69 90 111
90 32 55 90 125 148 125 90 55 21 58 122 159];

phi=[0 32 0 328 0 45 69 45 0 315 291 315 91 90 90 89 180 212 180 148
180 225 249 225 180 135 111 135 269 270 270 271];

end