An efficient 3-D sound experience for mobile applications

Group 3 Aalborg University
Copenhagen, Denmark

***Abstract*—The computational power of mobile devices has highly increased in the last few years and almost every device now comes with a GPS and compass sensor. This opens up possibilities to enhance the user experience in daily life. In this paper an application which has a 3-D audio engine as its core is presented which makes use of head related transfer function (HRTF) model to improve and augment the auditory experience of the user in addition to provide her or him with natural navigational cues. In a small experiment the developed 3-D audio engine is compared with a cosine panner modell [2] in terms of quality and efficiency of the navigational cues. Although the experiment did not reveal significant differences between the two models a critical observation of this study supports that a more sophisticated 3-D audio engine can increase the user experience in audio navigation.**

***Index Terms*—HTRF, mobile devices, soundscape, Pure Data, OpenFrameworks**

I. Introduction

A lot of research on modeling 3-D sound has been done with fairly good results [3]. The best results can be obtained by those models implementing 3-D sound using personalized Head related transfer functions (HRTF) which needs tedious measuring of impulse responses COMMENT:LARS[write it out here what else instead of saying etc.!] [4]. Mobile applications as well as multimedia productions are usually aimed at a big number of users, thus implementing HRTFs requiring measurements of the individual user is highly unpractical. Furthermore the convolution needed for using databases with Head related impulse responses is an unnecessary heavy technique in terms of computational power and memory, two things that are not abundant in mobile devices. Therefore, the aim of this project was to develop a computationally efficient and general HRTF model, yet, keeping the quality of a 3-D sound experience as good as possible. We used the model provided on a theoretical basis in [1] by implementing a combination of filters and delays in Pure Data and C. This audio engine was then embedded in a mobile application. Compass and GPS data provided by the mobile device was filtered and then used to compute from which direction and at what distance the sound should appear to be coming from. With the application users should be able to discover a virtual sound space situated in Copenhagen. The 3-D audio engine should satisfy two major requirements.

1. It should provide direction cues that are efficient in guiding the user to the position where a specific sound is situated.
2. It should provide an intuitive way of locating a sound source from a qualitative point of view.

The first aim refers to the speed at which the user can find a sound source receiving only the auditory cues, which optimally would not differ to a great extent of the time needed to navigate to the position of a natural sound source. The 3-D audio engine should yield better or at least as good result as a simple stereophonic panning as some studies already provided evidence of the efficiency of a simple panning to guide the user to a specific location COMMENT:LARS[Here we have to put some citations][]. Our second aim was that the model should give a qualitatively more natural and intuitive feeling of the sounds position in the space compared to the panning. While for the panning some level of abstraction as simple as hearing the sound louder from the left meaning that the sound is at the left has to be applied, with the 3-D audio engine we aimed to model a sound coming from a certain direction in a more natural way using a HRTF model which combines direction, distance and externalization cues.

In this paper, first, state of the art research on audio augmented reality will be elaborated on to set the theoretical framework. In the second part, the implementation of an efficient and general HRTF model for mobile devices will be presented. Thereafter, to evaluate the implemented audio engine, a comparative study between the 3-D audio engine and a simple stereophonic panning and its results based on both qualitative and quantitative measures will be presented. In the last part, a thorough discussion of the results in light of the two demands to the implemented 3-D audio engine will be presented and conclusions will be drawn.

A. State of the art in audio augmented reality

Recently, the progress in audio technology and computing paved the way for the introduction of completely new types of interactive audio applications []. Advances in mobile technologies have made it possible to create audio augmented spaces almost anywhere. For instance, spatial auditory displays that can provide the user with landmarks and are capable to attract the user's attention have been tested and introduced []. COMMENTS:LARS[Do we need this sentence? Because we say nothing about multiple sound sources except for the future works. So maybe we can put it there] -¿ Many experiments assessing qualitative and quantitative measures have been designed so far to better understand the way in which people usually perceive multiple simultaneous sources differently placed and to increase the level of immersion in the experience.

Theoretically, audio augmented reality has to be distinguished from the traditional concept of a virtual reality audio environment []. In virtual reality, generally participants are abstracted from the natural environment and are surrounded only by a completely synthetic one (acoustic and/or visual). On the opposite, in augmented reality a virtual environment is *superimposed* on a real one. To be more specific, in a mobile audio augmented environment participants are able to interact with the virtual audio mixed with a real visual scene and/or auditory soundscape. COMMENT:LARS[maybe some reference?]

COMMENT:LARS [We have to reformulate this one once the citatio is put in ;)]According to this definition, audio augmented reality should be within the boundaries of 1) a perfect augmentation of the listener's auditory environment and is achieved when the listener is unable to predict whether a sound source is part of the real or the virtual audio environment and 2) a set of artificial sounds that are not possible in the real world superimposed and fitted to the visual perceived world like in described in FirstAuthorName et al []. Any combination of these two fall within the boundaries of AAR.

Audio Aura (Mynatt 1995) was one of the first projects to deal exclusively with audio in augmented reality systems. It basically consisted in providing auditory information to users as they traveled through their workspace. These information was triggered as soon as the subjects entered particular locations in the workspace. The auditory cues used in *Audio Aura* were particularly interesting because most of them were associated to sounds from nature rather then recorded vocal, speech or synthetic sounds. Similar in approach to *Audio Aura* was the *Automated Tour Guide* [**?**] COMMENT:LARS[add something more about that automated tour guide like: ... in which subjecta had to do ...]. In both cases, triggers are readily identifiable, still and rarely changing. The sounds were assigned to pre-determined locations in space and there was no need to determine the precise location of an individual except for knowing when the individual entered the area of interest. COMMENT:LARS[How did they know when to play the sound???]

A successive work, *Hear&There* was able to determine the location and head position of the user using the information from GPS and a digital compass []. A user could listen to these 'audio imprints'COMMENT:LARS[is this a direct citation?] by walking into the area that a specific imprint occupied, which then was triggered. The essential premise of *Hear&There* was that a physical environment has been augmented with audio. All the sounds and the data were gathered inside that system. Since a 'Field Use' has been developed, in which the user wear a hardware portable system, *Hear&There* has undoubtedly contributed to an improved definition of mobile augmented reality environment. COMMENT:LARS[I am not sure if it contributed to the definition, rather then just falling into the definition?]

Human sound localization:
The auditory system is used by the humans for several purposes in the daily life. One of this purposes is to provide the necessary information to localize sound sources in various dimensions, (width, height, depth) and it is even possible to guess the size of the source.

When a sound event occurs, the waves travel in all directions and , when they reach us, our brain compares the signals received by the left and right ears to localize it in the horizontal plane. The spectrum of the signal reaching each ear is different, since the amplitude and phase information differs. These binaural cues are called *interaural intensity difference* (IID) and *interaural time difference* (ITD). However, these cues are not enough to localize accurately the source since with this information the listener can not determine if the sound is in front, above or behind. This region of positions where all sounds yield the same ITD and IID is called *cone of confusion*. This ambiguity can be solved with the information provided by the filter effect caused by the pinnae, head, shoulders and torso, which modify the spectrum of the sound that reach the listener's ears. The sum of all these features are characterized by the Head Related Transfer Functions (HTRF) which are not only frequency and direction-dependent but also differ from person to person. That dependency makes therefore hard to generalize the spectral features among individuals. It is well known that using a HTRF from one person in another can significantly impair the perception due to the individual differences in the anatomy, but it has also been shown that some people localize better the sounds than others, and their HTRFs are suitable for a large group of listeners.

Localization/Lateralization:
Headphones are an intergraded part of many mobile and wearable application and have been used successfully in countless virtual reality applications. Headphones make the ear receive the sound separately. That is advantageous because the whole scale of differences between the signals coming to ears, i.e. *interaural time differences* (ITD) and *interaural intensity differences* (IID), can be manipulated separately and individually [].
When there is a misfit between the perceived space and the sound the listener hear it is impossible to place it in the physical space. It is thus perceived in lack of better options to origin from inside the head[]. This effect is usually called *lateralization*, or intracranial, or 'inside-head-localization' (IHL). On the opposite, there is the effect of having the sound outside the heed, according to specific direction and distance, i.e. *localization* or 'outside-head-localization' (OHL). It has been necessary distinguish localization inside and outside the head. This difference in terminology serve to emphasis the difference between a sound source conveyed directly by headphones and that of a real source []. It has also demonstrated that a listener can make a clear distinction in headphones listening between localized (that is, sound inside the head) and lateralized sounds sources and that both these type can coexist in the listener's experience [].

Issues in headphones-conveyed sound:
Headphones auralization often produces an incorrect localization of virtual sound sources [] and many issues could be experienced.

a) Externalization errors: externalization is related to the perception of auditory distance such that there is a continuum in perceived locations of sources from inside the listener's head to any external position. One of the most severe problem in AAR is a perceived effect of *lateralization* even in sounds that should be located in a real environment. To avoid such effect several techniques can be used. For instance, as expressed in [], the effect of a lateralized sound in headphone listening can be produced using amplitude and delay differences in two headphone channels corresponding to each source. The main goal of virtual acoustic synthesis should be to produce sounds that seem *externalized*, that is, outside the listener's body []. In order to make a sound source externalized and let the user be capable of a correct judgment of the distance of the sound more sophisticated binaural techniques are needed. In particular, spectrum differences in the two ear signal due to *head-related transfer functions*, HRTF's, play an important role. Moreover, acoustic cues such as the amount of reverberation and control of signal level are necessary for a successful auralization of a virtual sound source. These are explained into details in further sections.

b) Localization errors: localization error refers to the deviation of the reported position of a sound source from a measured 'target' location, i.e. the listener fails in matching the correct location of the sound. Localization errors can be divided in *azimuth* (deviations along the horizontal plane) and *elevation* errors (deviation from eye-level elevation) []. Dynamic cues related to head turning and other movements of either a listener or a sound source should be taken into consideration []. These errors might come from some location accuracy. Determining the accurate position of the user of one of the most important task in an AR system owing to that system always produces output to the used based on his or her location in space []. That said, any location inaccuracy should be avoided using GPS receiver with high sensitivity and reliability. Here, the implementation design takes a crucial role as noted in [].

c) Reversal errors: sometimes called front-back or back-front 'confusion', that error refers to the judgment of a sound source as located on the opposite side of the interaural axis than the target position [] due to the cone of confusion. An informal proposal has been made, which tries to help the user in front-back discrimination on the basis of the familiarity of the effects on timbre cues, e.g. unique patterns in *early reflections* depending on the virtual sound location. Indeed, this has not yet been verified experimentally. Informal Proposal has suggested to use difference in *early reflections* to help judge where the sound origins. However in application where the listener can change its angle to the source, the front and back will soon be obvious through the movement of the sound.

Conclusion:
Although it is a recent field of research, several works and experiments in the context of audio augmented reality has been made so far. Most of research has been done mainly on the use of visual information with very little attention paid to the audio aspects.
The use of audio and its refinement in an augmented reality environment is a big deal. There are many issues and challenges that an audio environment presents []. For instance, providing a way for the user to orient himself or herself in an only-audio virtual space is even more difficult than providing visual cues. Another interesting aspects of audio, is that it, contrary to visual images or signs, is unstable and thus need to be continuously repeated or be infinite in order to work as a directional cue []. Finally, with the advent of portable systems such as sensors-equipped smartphones and laptops a further development of *geospecific AR* has been made possible. That is systems that use locational sensing and user tracking technologies (e.g. Global Positioning System, or GPS) to let the user navigate through a audio-augmented space and to get a more defined listener's position, which is essential for a correct externalizaton. More information will be given in further sections.

II. Design, Implementation and Mobile Application

A. Introduction

The mobile application developed has been called '*Audio Treasure Hunt*" and the user has to find one sound located around him in the shortest time possible. As stated above, a Pure Data external called *headShadow*' has been implemented and it performs a head shadowing effects. The reason that led to this development process is due to efficiency [5], since the application runs on mobile device where the computational power is limited to its hardware. Indeed, there is a Pure Data object called earplug∼ which is a real-time binaural filter based on KEMAR impulse measurement. It allows you to spatialize a sound in real-time. It basically takes the KEMAR data set, and interpolates 366 locations where HRTF measurement exists in a spherical surface. you get azimuth control 0-360 and elevation -40 - 90[1]. Testing this Pd object on a personal computer has shown its inaccuracy. Hence, it has not been tested on a mobile device. Given that the aim of this application is testing the implemented HRTF model, two versions of this application have been developed. The first one uses the HRTF model [1] and the second one uses a cosine Panner model [2]. At a later stage these two applications have been tested on users and the time required to find the sound has been compared. That is, statistical analysis has been applied on the data in order to prove its quality. To develop and implement this application several programming languages have been used[2], with OpenFrameworks[14] as the environment toolkit. This first version of the application runs only on Android platform.

[1]https://puredata.info/downloads/earplug
[2]Java, C/C++, PureData

B. Sensor data retrieving

The implemented mobile application required two kind of sensors data: *GPS* and *orientation*.

- GPS is a way of determining location that has become common with the increasing of portable technologies. Any device that receives a GPS signal is called a GPS *receiver*. As explained in [] *ask Mattia where to find this reference* the receiver calculates its position by precisely timing the signals sent by the GPS satellites. The GPS user position is needed to calculate the distance to the sounds that were placed on fixed locations in the real environment.
- Orientation is computed using the orientation sensor. It is a software-based sensors which derives its data from the accelerometer and the geomagnetic field sensor. *cannot add http://developer.android.com/guide/topics/sensors/sensors_position.html*. Therefore, it is possible to determine a device's position relative to the magnetic North Pole.

To access the GPS data *ofxMaps*[3] and *ofxGeo*[4] addons has been used. The GPS position on an Android platform requires a listening mechanism, which has been implemented in openFrameworks by calling the `ofRegisterGPSEvent()` in the `setup()` method of the source code. In the `ofApp.h` a method has been added in order to handle the updates from the Android OS system calls. It is named `locationChanged()` and adds an event handler for the dispatched event [] *ask Mattia where to find this reference*. Moreover, the `startGPS()` method is called at the beginning and `stopPGS()` should call when quitting the application to avoid an overconsumption of phone battery.

To retrieve compass orientation a specific addon called *ofxOrientation* has been implemented. It accesses the 'TYPE_ORIENTATION' provided by the Android API and it measures degrees of rotation that a device makes around all three physical axes (x, y, z). In this case, only the z-axis is needed. Having the device's position relative to the magnetic North Pole and the sound location, *theta* can be computed.

C. Computing theta and distance

To compute the distance between two point, in this case the sound location and the user location, the *Haversine* formula[5] is taken into consideration.

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 + \cos\varphi_2 + \sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 * atan2(\sqrt{a}, \sqrt{(1-a)}) \quad (2)$$

$$d = R * c \quad (3)$$

where φ is the *latitude*, λ is the *longitude*, R is earth's radius (mean radius = 6,371 km).
Therefore, the distance is computed by calling the *GeoUtils::distanceHaversine* method, which in then used to simulate the sound pressure in a free field. First of all, to calculate the angle *theta* required by the HRTF model the angle *beta* must be computed. The angle *beta* is defined as the angle between the positive y-axis and the sound location. Hence, the angle *theta* is then obtained and fed to the HRTF model.

D. Audio engine

The audio engine is the core part of the mobile application. As shortly mentioned in the introduction a filter based HRTF model has been implemented, which is claimed to be both efficient and of reasonable quality [1] in order to achieve a satisfying user experience and an efficient real time application. The workflow given in [1] has been followed, which is depicted in Figure 1.

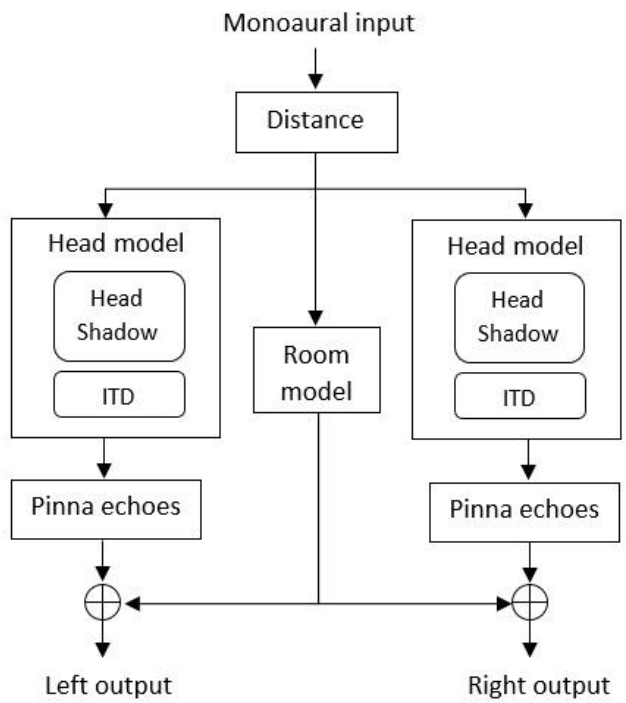

Fig. 1. Components of the model provided on a theoretical basis in [1]

To implement the HRTF model [1] the audio programming language PureData has been used. Additionaly, in order to embed this model in the Android platform *libpd*[6] has been taken into consideration. Therefore, a PureData external called *headShadow* has been implemented and written in C. Each single part of the HRTF model has been implemented using PureData. First, the monoaural sound which has been processed to add a range effect, goes to a head model and a room model. The binaural output of the former is then processed by the pinna model, and finally each one is added to the output of the room model to provide an externalization effect.

In the following formulas, the angles are measured in radians in the interaural-polar system, where θ is the azimuth angle and ϕ is the elevation angle. However, in the mobile

[3]https://github.com/bakercp/ofxMaps
[4]https://github.com/bakercp/ofxGeo
[5]http://www.movable-type.co.uk/scripts/latlong.html
[6]http://libpd.cc/

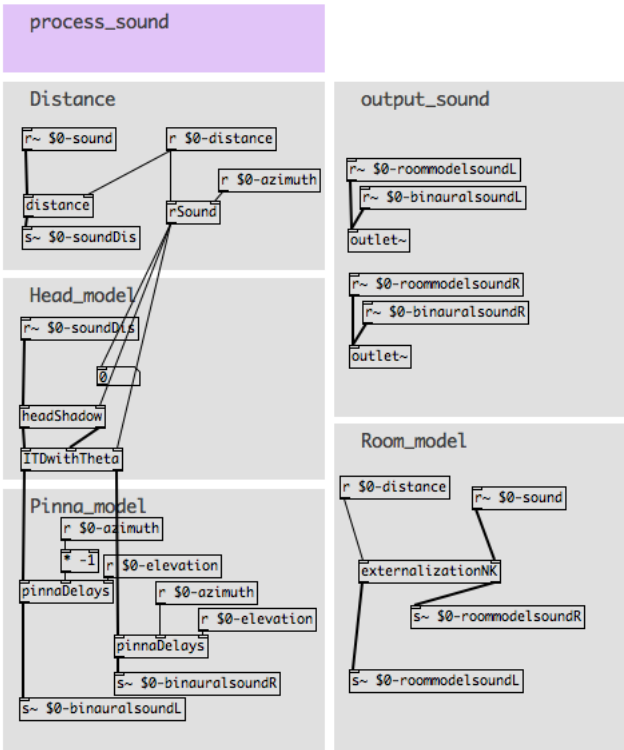

Fig. 2. The audio engine implemented in Pure Data

application the latter is no considered since all the sound sources are placed in the horizontal plane without elevation.

1) The distance model:
In order to simulate the sound pressure in the free field the inverse distance law has been applied. Since sound intensity is proportional to the square of sound pressure, the inverse square law (for sound intensity) becomes the inverse distance law (for sound pressure) [6]. Therefore, sound pressure is inversely proportional to distance *r*:

$$P = \frac{k}{r} \tag{4}$$

where *P* is the sound pressure, *k* is a constant and *r* is the distance from source. Hence, for every doubling of distance *r* from the sound source, sound pressure will be halved. When the distance from the source is doubled, the sound-pressure level decreases by 6 dB [6]. This model has been implemented in Pure Data.

2) The head model:
The path length difference from the sound source to the two ears and the shadowing effect produced by the head at the far ear, lead to a delay and a intensity difference in the sound arriving at the left and right ears. In order to estimate the time delay, the Woodworth's formulas are used [7] :

$$ITD = (a/c)[\theta + sin(\theta)]\ [0 \leq \theta \leq \pi/2] \tag{5}$$

$$ITD = (a/c)[\pi - \theta + sin(\theta)]\ [\pi/2 \leq \theta \leq \pi] \tag{6}$$

where *a* is the approximated head radius, θ is the azimuth angle in radians and *c* is the head radius. The time differences between the audio signal reaches the head and the ears are therefore

$$T_L(\theta) = \frac{a + a\theta}{c} \tag{7}$$

$$T_R(\theta) = \frac{a - asin(\theta)}{c} \tag{8}$$

These formulas refers to a source in front of the head and on the right, with azimuths $0 \leq \theta \leq \pi/2$. If the source is placed on the left ($-\pi/2 \leq \theta \leq 0$), the expressions are reversed.

The head shadow effect is characterized in the following analog transfer function:

$$H\,(s,\theta) = \frac{\alpha(\theta)s + \beta}{s + \beta},\ where\ \beta = \frac{2c}{a} \tag{9}$$

Since this is an analog transfer function, it has to be derived to the digital version. Therefore, the following transfer function is obtained [7]:

$$H\,(z,\theta) = \frac{2\alpha(\theta) + T\beta + z^{-1}(-2\alpha(\theta) + T\beta)}{2 + T\beta + z^{-1}(-2 + T\beta)} = \frac{Y(z)}{X(z)} \tag{10}$$

Hence, the filter coefficients $a_0 = 2\alpha(\theta) + T\beta$ and $a_1 = -2\alpha(\theta) + T\beta$ as well as $b_0 = 2 + T\beta$ and $b_1 = -2 + T\beta$ are given. It was necessary to isolate the output $Y(z)$ in Equation 10 as well as going from the frequency to the time domain. The result is given in Equation 11[8]:

$$Y[n] = \frac{a_0 X[n] + a_1 X[n-1] - b_1 Y[n-1]}{b_0} \tag{11}$$

3) The pinna model:
High frequency components that arrive to the listener's ears are influenced by the pinna, which provide azimuth information. It has been studied mainly because of its contribution to the estimation of the elevation [1].

Our model has the following form:

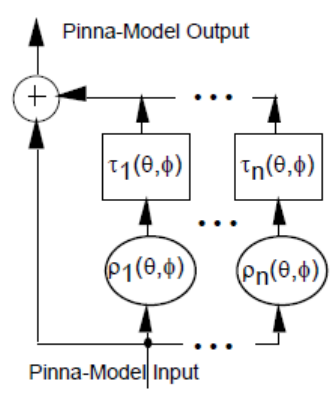

Fig. 3. Pinna model. Image taken from [1]

where the ρ_k are the reflection coefficients and the τ_k are the time delays of the *k*th event of a total of *n*. Informal listening tests showed that 5 events were enough to represent the pinna response and that it was convenient to use constant values for

[7]For the derivation of Equation 10 see appendix A
[8]For the derivation of Equation 11 see appendix B

the amplitudes ρ_k, independent of azimuth, elevation and the subject [1]. The time delays seem to be properly approximated by the following formula:

$$\tau_k(\theta, \phi) = A_k cos(\theta/2) sin(D_k(90^\circ - \phi)) + B_k \qquad (12)$$

In this equation, dependent on the azimuth and elevation, the A_k is an amplitude, B_k an offset and D_k is a scaling factor that should be adapted to the individual listener. In the following table one can see the values for the parameters used in the pinna model. Only one set of values for D_k in our application have been used.

TABLE I
PINNA MODEL COEFFICIENTS

k	ρ	A_k	B_k	D_k
1	0.5	1	2	1
2	-1	5	4	0.5
3	0.5	5	7	0.5
4	-0.25	5	11	0.5
5	0.25	5	13	0.5

4) The room model: In [1], they imply a very simple room model consisting of only one delay with variable delay time and level according to the distance. An attempt to make a similar model was made by setting the delay time to the difference between the direct path from the object to the listener and a path that bounces on the ground half way between the object and the listener. We opted for a more complex model due to the poor experienced effect and the large alteration (comb filtering) of the sound quality

The chosen model is based on the *reverb* algorithm *rev2* implemented in pd. Compared to the original model the amount of early reflections are reduced and spread more out through this we simulate a very large but little reflective room, with a subtle tail, resembling the characteristics of an outdoor environment. the amount of *reverb* is diminished according to the distance between the user and the virtual sound source. This happens significantly slower than the 2/d of the direct sound namely 8/(15+d), in this way the ratio between the direct and the reflected signal change even though the total level go down, this model approximate the test data in [3]. The *rev2* has a direct through signal incorporated, this has been removed in order to easier control this dry/wet relation. As showed in the model (fig. 1) the room model is added in parallel with the Head model and added the output of the pinna model.

A further improvement could implement variable early reflections delay-times. And separate control over reverb tail and early reflections level.

III. THE EXPERIMENT

One of the aims of our experiment was to investigate how fast users were able to locate and navigate to a single sound source using our 3-D audio engine compared to a cosine panner based on [2] and 2/d distance calculation which served as a baseline performance. Therefore the time needed from the beginning of a trial until the sound source was found was measured.

Additionally we assessed a qualitative analysis of how intuitive it was for subjects to locate the sound as well as investigating whether the sound seemed to be embedded in the real scenery. These results were again compared to the cosine panner model [2]. The data was obtained with a 10 point lickertscale questionnaire going from “don’t agree at all” to ‘completely agree”.

(Here we can write a one paragraph preview of our results.)

A. Materials and methods

1) Participants: One participant had to be discarded because of Qualitative data were collected from 11 participants (3 women and 8 men; mean age: 25, ranging from 22 to 30 years). For the quantitative data 6 more participants were measured summing up to in total 17 participants (4 women and 13 men; mean age: 25, ranging from 22 to 30 years). All participants were students coming from different backgrounds.[9] Six students were already familiar with 3-D audio sound and four of them had experience in audio navigation before conduction of the experiment. The other participants were naive. All participants reported normal or corrected to normal vision and no hearing deficits. Participants received no form of compensation other than gratitude.

2) Apparatus and stimuli: The experiment took place in a free field (grass) in ”Valby Parken” in Copenhagen, Denmark (Coordinates: 55°38’22.3”N, 12°31’27.4”E). The mobile device used was a Motorolla Moto G running with Android version 5.0.2. The device was connected to Sony MDR ZX600 stereo headphones. Only one person at a time was tested.

To create the auditory stimulus, digital vocal recordings of a male voice (age: ...) were collected using a ... . Recordings were done with ... bits of resolution for amplitude at a sampling rate of ... kHz. Auditory stimuli were normalized in peak amplitude so that at the minimal distance to the sound source (7 meter) using the maximal level of volume on the mobile device, the sound level of each sound source was at ... dB.

3) Design: The experiment comprised one within subject factor at 2 levels (sound engine). The two levels of the sound engine were our 3-D audio engine (3-D) and a cosine panner [2] and distance amplitude modulation (panning).

The starting point was fixed and the distance of the sound source from that point was always 48 meters.[10] The sound could appear at any angle from the starting point.

For all trials the same soundfile was played. Each participant performed a total of 6 trials meaning 3 trials in each of the two conditions. Participants were encouraged to take a short break whenever they felt fatigued.

4) Procedure: The position (angle from the starting point) of the sound sources in the different trials were completely randomized within and between subjects independently from the sound engine. This ensured that participants could not predict

[9]However, 14 subjects were master students in Sound and Music Computing at Aalborg University Copenhagen.

[10]However, because of poor GPS accuracy, the location of the sound source was calculated with the momentary GPS coordinates provided by the phone sensor instead of the actual fixed position in the free field. This ensured that the initial distance to the sound source was always the same.

the location of the sound in the next trial. The order of trials with the one or the other sound engine was not randomized within subjects. Nevertheless, half of the participants started the experiment with the 3-D audio engine while the other half started with the audio panning sound engine to counterbalance for effects related to training. After each block of trials (one block were 3 trials with one of the sound engines) participants had to fill out the qualitative questionnaire.

Each trial was initiated by the experimenter with a button press after which the sound source was played. No visual feedback about the location of the sound source or the already passed time to locate it was given. For localization, subjects had to rely solely on the auditory cues given via the headphones. /footnoteAdmittedly participants could have used the position of the starting point as some kind of fix point. However, participants did know nothing about the possible distances of the sound source. Additionally, questions after the experiment about applied strategies did not reveal that participants used visual cues for localizing the sound source. As soon as the participants reached a radius of 7 meter around the sound source, an earcon was played indicating the success of the localization. The participants went back to the initial starting point and the next trial was started.

Before the experiment, participants were familiarized with the target sound as well as the earcons that were played when having reached the location of the virtual sound source. Additionally, all participants performed at least two practice trials with the sound engine they were tested with first before the actual measurements with that sound engine began. The location in the practice trials were fixed at the same positions for all participants to allow us to guide the subject when they got lost.

Participants were instructed to find the sound source as fast as possible. No information about the possible locations (distances and angles) was given to the subjects. Trials which lasted longer than 5 minutes were aborted and labeled as ”not found”.

5) Analyses: For the quantitative analysis 26 trials were excluded because either analyses of GPS data for those trials revealed very poor precision (sudden jumps or no changes in the GPS signal for a period of time) or the orientation sensor got stuck which either increased the time needed to find the sound source to a great extent or made it even impossible for the subjects to locate the sound source within 5 minutes. One subject had to be discarded completely from the analyses because there was no single trial with reliable GPS for the panning engine. The other 16 subjects performed at least one trial in each condition. The remaining trials to be analyzed summed up to 41 trials in the 3-D condition and 42 trials in the panning condition.

The reasons for choosing a 10 point likertscale instead of the often referenced 5 or 7 point scales [] were that a greater number of responses should equalize the distances between the possible answer possibilities supporting treating likertscale data also as interval data []. This gives us the grounds to use paired samples t-test for the qualitative analysis which is more powerful than the Wilcoxon signed rank test enabling us to detect smaller differences in ratings []. Since we only had few questions, having so many answer possibilities should not have fatigued the subject too much so that accuracy of the responses should not have suffered due to too many answer possibilities []. Though, it has to be noted that subject were outside when filling out the questionnaire which due to cold temperatures might have driven the subjects to respond in a fast pace leading to less accurate responses.

The threshold chosen to correctly reject the null-hypothesis was always 5% (α = .05).

B. Results

1) Quantitative: On average subjects needed 104.6 seconds (SD = 44.8) to find the sounds for the 3-D conditions and 109.8 seconds (SD = 44.1) for the panning condition.

We computed a paired samples t-test to compare the results for both sound engines which revealed no significant difference between the means for both sound engines (t(15) =-.48, p =.64). However, checking whether our data is normally distributed using the Shapiro-Wilk test showed that for the 3-D data the assumption of normality was violated (Z=.81, p = .004) and close to significance in the panning condition (Z=.89, p =.054). Therefore, we additionally computed the non-parametric Wilcoxon signed rank test which does not assume normal distributed data. Nevertheless, we could not find a significant difference in scores for time needed to find the sound source between both conditions (median: 3-D=85.3, panning=95.9; Z=-.83, p = 0.41).

2) Qualitative:
The Shapiro-Wilk test showed that the data was normally distributed both for the 3D model (Z = 0.933, p = 0.44) and the Panning model (Z = 0.95, p = 0.67).

Evaluation of the questionnaires, Table II, showed that the mean and median values for questions 1, 2, 3 and 4 for the 3D model were only numerically different from the panning. Neither the paired samples t-test nor the Wilcoxon signed rank test showed significant results comparing the two conditions (3-D vs. panning). However, performing a one sample t-test on means of the four questions for each model comparing it with a neutral response (5.5) showed that users rated the connection between their position and what was presented via the headphones better than neutral for the 3-D sound engine (t(10) = 4.32, p =.002; Z =2.74 , p = .006) but not for the panning (t(10) = 1.07, p = .31; Z = .90, p=.367). For all other questions there were no significant differences.[11]

Summing the results for the first four questions to form an overall category of quality of the two sound engines lead to the mean and median values depicted in Table III. Again, the difference between both sound engines were only numerically visible since performing a paired samples t-test and a Wilcoxon signed rank test showed no significant results (t(10) = 1.05, p = 0.32; Z = -1.07, p = .283). Furthermore,

[11]It is worth noting that choosing mean and median as 5 instead of 5.5, showed that also for question two, a one sample t-test and the wilcoxon signed rank test were significant for the 3-D (t(10) = 2.52 , p = .03 ; Z = 2.19, p =.028) but not for the panning sound engine (t(10) = 1.19, p = .26 ; Z = 1.2, p =.23).

[12]It should be noted that for this question lower values represent higher quality of a 3-D sound because of the direction of the question.

Question	Mean (s)		Std (s)		Median (s)	
	3D model	Panning	3D model	Panning	3D model	Panning
1	6.36	5.91	1.75	1.70	6	6
2	6.91	5.82	2.51	2.27	8	6
3	7.09	6.00	1.22	1.55	7	6
4[12]	6.64	6.09	2.58	1.97	7	5
5	6.09	6.55	2.21	2.54	7	7

TABLE II
QUESTIONNAIRE RESULTS

running a one sample t-test and wilcoxon signed rank test to compare the results of both sound engines to a mean (5.5) that represents being neutral towards a question showed no significant results for both conditions (HRTF: t(10) = 1.54, p = .154; Z = 1.54, p = .13; Panning: t(10) = -.22, p =.829; Z = -0.5, p = .96). However, summing only the first three questions, showed that the 3-D model ratings were significantly higher than the neutral mean (5.5) (t(10) = 2.83, p = .018; Z = 2.09, p = .036) which was not the case for the panning condition (t(10) = .86, p = .41; Z = 0.76, p = .447).

Question	Mean (s)		Std (s)		Median (s)	
	3D model	Panning	3D model	Panning	3D model	Panning
1 - 4	5.93	5.41	0.93	1.36	6	5.25

TABLE III
OVERALL RESULTS

IV. DISCUSSION

The results show that both sound engines did not differ in terms of performance. Although numerically subjects were slightly faster to locate the sound source when using the HRTF audio engine, this difference could not be verified statistically.[13]. On top of the small sample size, an unreliable GPS source (different number of GPS satellites fixes for each experiment, with mean of fixes: 11 ranging from 5 to 13) and also the different weather conditions could explain the large variance in the data and thus the probability of finding significant results comparing the two audio engines used.

As for the qualitative analyses the explanation why our 3-D audio engine was not rated significantly better then the other could be due to the choice of a unnatural high reverberation level for a free field. This explanation is inspired by the fact that a lot of participants had background knowledge in acoustics and therefore made more predictions about a proper reverberation level for a free field. Some subjects even reported the reverberation to be the reason for low ratings of the question regarding whether the sound source was perceived to be embedded in the real scenery. This negative effect on the ratings might have nullified the difference between panning and HRTF in terms of perceiving the sound moving around and not inside the head. Some support for this explanation can be gained from noticing that the ratings of the first three questions taken together (as well as the third question alone) were significantly higher than a neutral response for the HRTF but not for the Panning. This might illustrate a slight superiority in quality of the HRTF.

Another issue that should be addressed is the adequacy of the questionnaire used in this experiment. Probably, in spite of the discussion about why we chose to use a 10-point likert scale instead of the often used 5- or 7-point likert scale, subjects might not have been able to accurately respond to the questions posed leading indistinguishable results. Also the fact that most of the questions were not different to the mean of the scale provide some evidence that subjects were either actually indifferent to the posed questions or had difficulties of knowing which number corresponds to which degree of agreement to the stated question.

Finally, as many of our participants listen to audio via headphones more than 3 hours a day (9 out of 11), these participants might have had a great top-down expactation of hearing the sounds coming from the headphones rather than being externalized meaning coming from a specific location in the real world. This might have affected the experience of an externalized sound especially considering that the sounds had no visual representation in the free field. It could be interesting to investigate if for example exposure time to the soundsources has an effect on externalization ratings.

A. Future improvements

V. CONCLUSION

ACKNOWLEDGMENTS

[13]However, it might have been the case that our sample size was so small that the difference in performance would have had to be bigger to show significant results given the sample size in this experiment. As noted by several methodologists Type II errors have a high probability to occur for low sample sizes [8]

LIST OF FIGURES

1 Components of the model provided on a theoretical basis in [1] 4

2 The audio engine implemented in Pure Data . . 4

3 Pinna model. Image taken from [1] 5

LIST OF TABLES

I Pinna model coefficients 6

II Questionnaire results 7

III Overall results 8

APPENDIX A
DERIVATIONS

The analog transfer function has to be derived to the digital version by applying a bilinear transform:

$$s = \frac{2}{T}\frac{z-1}{z+1},\ where\, T\, is\, the\, sampling\, interval\, in\, seconds \tag{13}$$

Applying the substitution in equation 13 to equation 9, following filter function in the digital domain is obtained:

$$H\left(z,\theta\right) = \frac{\alpha(\theta)(\frac{2}{T}\frac{z-1}{z+1}) + \beta}{(\frac{2}{T}\frac{z-1}{z+1}) + \beta} \tag{14}$$

The derivations followed to get it can be seen as follows

$$\begin{aligned}
H\left(z,\theta\right) &= \frac{\alpha(\theta)(\frac{2}{T}\frac{z-1}{z+1}) + \beta}{(\frac{2}{T}\frac{z-1}{z+1}) + \beta} \\
&= \frac{\frac{2\alpha(\theta)(z-1)}{T(z+1)} + \frac{T\beta(z+1)}{T(z+1)}}{\frac{2(z-1)}{T(z+1)} + \frac{T\beta(z+1)}{T(z+1)}} \\
&= \frac{\frac{2\alpha(\theta)(z-1)+T\beta(z+1)}{T(z+1)}}{\frac{2(z-1)+T\beta(z+1)}{T(z+1)}} \\
&= \frac{2\alpha(\theta)(z-1) + T\beta(z+1)}{2(z-1) + T\beta(z+1)} \\
&= \frac{z2\alpha(\theta)(1-z^{-1}) + zT\beta(1+z^{-1})}{z2(1-z^{-1}) + zT\beta(1+z^{-1})} \\
&= \frac{2\alpha(\theta)(1-z^{-1}) + T\beta(1+z^{-1})}{2(1-z^{-1}) + T\beta(1+z^{-1})} \\
&= \frac{2\alpha(\theta) - 2\alpha(\theta)z^{-1} + T\beta + T\beta z^{-1}}{2 - 2z^{-1} + T\beta + T\beta z^{-1}} \\
&= \frac{(2\alpha(\theta) + T\beta) - (2\alpha(\theta) + T\beta)z^{-1}}{(2 + T\beta) - (2 + T\beta)z^{-1}} \\
&= \frac{(2\alpha(\theta) + T\beta) + (-2\alpha(\theta) + T\beta)z^{-1}}{(2 + T\beta) + (-2 + T\beta)z^{-1}}
\end{aligned}$$

APPENDIX B
DERIVATIONS

$$\begin{aligned}
H(z,\theta) = \frac{Y(z)}{X(z)} &= \frac{a_0 + a_1 z^{-1}}{b_0 + b_1 z^{-1}} \\
&\iff Y(z)(b_0 + b_1 z^{-1}) = X(z)(a_0 + a_1 z^{-1}) \\
&\iff Y(z)b_0 + b_1 Y(z)z^{-1} = a_0 X(z) + a_1 X(z)z^{-1} \\
&\iff Y(z) = \frac{a_0 X(z) + a_1 X(z)z^{-1} - b_1 Y(z)z^{-1}}{b_0} \\
&\rightarrow Y[n] = \frac{a_0 X[n] + a_1 X[n-1] - b_1 Y[n-1]}{b_0}
\end{aligned}$$

APPENDIX C
OPENFRAMEWORKS

OpenFrameworks[14] has been choosen as development framework rather than Android Studio[15]. This choice is due

[14] http://openframeworks.cc/
[15] http://developer.android.com/sdk/index.html

to cross-platform reasons. Even though this first application runs only on Android, future improvements will also inlcude iOS development. Discussing all the steps involved in the building process is beyond the scope of this paper, but a short explanation of what is openFrameworks and the main operations used to achieve this application will be given.

1) Description:
OpenFrameworks, by definition given on its website, is:

an open source C++ toolkit for creative coding[14].

Since it is entirely written in C++, distributed under MIT license and actually runs on five operative systems and four IDEs, it is massively cross-compatible. It gives the opportunity to deal with code designed to be minimal and easy to grasp[14]. That *simple and intuitive framework for experimentation*[14] is designed to work as a general purpose glue and wraps together several commonly used libraries, such as *OpenGl, OpenCv, PortAudio* and many more. Nowadays, this is a popular platform for experiments in generative and sound art and creating interactive installations and audiovisual performances[14]. The current operative version is *0.9.0*.

2) Addons:
its design philosophy claims for a collaborative environment. It thrives on the contributions of many people, and collaborate mainly on addons and projects. An *addons* is made of several snippets of code put together in order to extend openFrameworks functionality, bring some external framework and allow it to be integrated into openFrameworks project or make specific and complicated tasks easier and reusable in other project []*ask Mattia where to find this reference* . It also generally contains the library itself in a form that is ready to be linked to project binaries. Several *third-party* addons has been used such as *ofxGui, ofxXmlSettings, ofxAndroid, ofxGeo, ofxMaps, ofxTween, ofxPd*. Additionally, it has been implemented a specific addons called *ofxOrientation*, which accesses sensors data regarding orientation. This step was necessary in order to retrieve the required angle between the sound location and the user orientation, which is then use by the HRTF model.

3) The openFrameworks project:
All openFrameworks project have a similar structure of folders and files. The most important folder among them is for sure the *src* folder. It contains all the source codes and consists at least of `main.cpp` (containing the `main( )` function to let the operating system start the application), `ofApp.h` (containing declaration of the specific class) and `ofApp.cpp`, which contains definition of all functions declared in the previous file. All the methods in that class are *event-handling* methods, hence they are triggered in response to events that happens inside the application such as mouse scrolling and program quitting.To create a new project, the Project Generator wizard has been used which is located in the same directory and directly provided by the environment. Such a way is simple and it is especially useful when dealing with several addons, which are automatically linked. At its simplest, working with an openFrameworks project is adding new code to the appropriate method, or just create a new one and declare it in the `ofApp.cpp`. In[14] a further explanation of the main methods and their workflow is provided.

REFERENCES

[1] C. Brown and R. Duda, "An efficient hrtf model for 3-d sound," in *Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on*, pp. 4 pp.–, Oct 1997.

[2] A. Farnell, *Designing Sound*. The MIT Press, 2010.

[3] D. Begault, *3-D Sound for Virtual Reality and Multimedia*. AP Professional, 1994.

[4] A. Meshram, R. Mehra, and D. Manocha, "Efficient hrtf computation using adaptive rectangular decomposition," in *Audio Engineering Society Conference: 55th International Conference: Spatial Audio*, Aug 2014.

[5] J. Blum, M. Bouchard, and J. Cooperstock, "Whats around me? spatialized audio augmented reality for blind users with a smartphone," in *Mobile and Ubiquitous Systems: Computing, Networking, and Services* (A. Puiatti and T. Gu, eds.), vol. 104 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 49–62, Springer Berlin Heidelberg, 2012.

[6] F. Everest and K. Pohlmann, *Master Handbook of Acoustics*. McGraw-Hill Education, 2009.

[7] N. L. Aaronson and W. M. Hartmann, "Testing, correcting, and extending the woodworth model for interaural time difference," *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 817–823, 2014.

[8] ??, "??." ??