Audio Engineering Society

Conference Paper

Presented at the International Conference on Audio for Virtual and Augmented Reality, 2020 August 17–19, Online

Virtual Reality Music in the Real World

Itai Neoran[1], Matan Ben-Asher[1], and Gal Alchanati[1]

[1] *Waves Audio Ltd., Azrieli Center 3, Tel-Aviv 6701101, Israel*

Correspondence should be addressed to Itai Neoran (itai@waves.com)

ABSTRACT

Virtual sound localization techniques are becoming increasingly popular when listening to music over headphones. These methods involve binaural audio rendering, which requires adaptation to movements of the listener's head and body, for example when listening to music in a train or in a car. Proposed solutions use virtual loudspeakers controlled by head-tracking devices, where the directional cues in a musical track relate to a reference direction in the real-world coordinate system. This paper also discusses the question: Which is the ideal reference direction for head tracking, and how should head tracking adapt to a listener in motion? We propose the NX-tracker algorithm, addressing a wide-range of music-listening and movie-watching use cases, with solutions for one and two tracking sensors.

1 Introduction

Virtual sound rendering techniques allow headphones listeners to experience music, movies, and game audio in an unprecedented manner. Unlike the legacy stereo reproduction over headphones, binaural audio makes it possible to place virtual sound sources all around the listener, for surround audio reproduction and for increased realism. Listeners can perceive voices and instruments arriving from the front, rear, sides, overhead, below the listener, and at various distances from the listener.

Binaural audio takes advantage of two important properties of headphones audio reproduction:

- Each of the left and right audio channels is played only to the corresponding (ipsilateral) ear, with a minimal acoustic leak into the opposite (contralateral) ear.
- The sound arrives directly to the ears, without first reflecting from walls or objects in the room.

With these two properties, binaural audio can simulate what we would hear with any sound source positioned anywhere around the listener. While binaural audio can be recorded using a dedicated binaural microphone, it can also be generated from regular audio: given the audio signal representing the sound source and information about its position, this process is called "binaural audio rendering", and is achieved through digital signal processing. The rendering involves filtering the audio with the spatial acoustic filter, corresponding to the desired virtual source position, and synthesizing the sound that would have arrived at each of the ears from a physical sound source in an acoustic space.

The binaural filters are called "head-related-transfer-function" (HRTF), including the pinna filter from the desired direction, the inter-aural time difference (ITD), and the inter-aural level difference (ILD), described in [1] and others. Binaural systems can also include a reverberation filter, simulating a virtual listening room. Adding synthetic reverberation contributes to the realism of the sound sources, such as perception of distance and externalization (hearing the sound from outside the head). In an earlier work, the authors of this paper developed NX, a commercial binaural audio algorithm that applies HRTF filters and simulated room reverberation to audio signals, see [2, 3, 4].

For music enthusiasts, binaural rendering presents new opportunities: Music can be consumed at home or while travelling. Traditionally, the way to listen to music in surround involved a complex setup of loudspeakers, impractical for most homes, and inaccessible when moving around and during commuting. Unlike movie watching or gaming, which assume a screen in front of the listener, music is mostly listened-to outside any "sweet spot" seating position, and with the listener's head not necessarily turned to the direction of the frontal loudspeakers.

With headphones and binaural audio combined, one can now experience surround music without any constraints on the specific use case, location, or head and body pose. By nature, the sound from regular headphones follows the head movements, thus bringing the same wave-front to the eardrums, independently of the head orientation. If a multichannel loudspeaker set-up is simulated over headphones, consumers no longer need to worry about buying or installing complex loudspeaker setups of 5.1, 6.1, 7.1, 10.2, Quad, and Ambisonics, required for example by ITU [5] see figure 1.

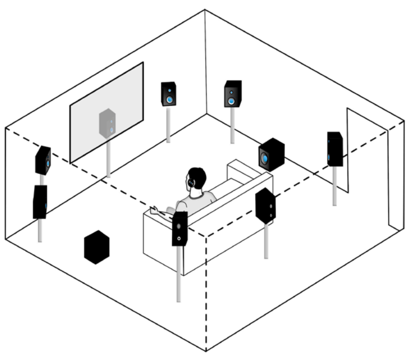

Figure 1. Surround setups and binaural audio

However, it has been demonstrated in [6, 7] that a binaural system with HRTF alone is sometimes not enough to convince the listeners that they are hearing real sound-sources arriving from outside the headphones. In many cases, HRTF filters are also insufficient to produce the illusion of directional sound from the rear or front. In the real world, when we turn our heads slightly to the left, the front sound source seems to move a little to the right. If we fail to emulate this effect, our brain is forced to conclude that the sound comes from inside our head. This phenomenon is so common with headphones, that most listeners have become accustom to this unnatural sonic experience. In the same manner, if a real sound source was playing from behind us, then in turning our head to the left, the sound would get closer to the left ear. When this effect is not reproduced, we experience a phenomenon called "front-back confusion", described by [6, 7] and others, where listeners attribute frontal sound sources to originate from behind and vice versa.

To reproduce the head rotation effect correctly, head-tracking devices can be used. If we know the three-dimensional orientation of the head, and the exact location of the virtual sound sources we wish to emulate, we can compensate for any head movement in digital audio processing, and make it seem like the sound sources correspond to the head orientation in the correct manner. This also solves the front-back confusion problem [7, 8]. Recently, head-tracking devices have become commercially available, whether as additional wearables or hidden inside the headphones arc. Wearable sensors may use Inertial Measurement Units (IMUs), magnetic field detection, or track an external beacon emitting an infra-red or a radio-frequency signal. Video cameras placed in front of the listener or in a mobile device, are also capable of tracking the head and reporting its pose. The head tracker developed by the authors has two versions: one using an IMU sensor (figure 2) and another using a video camera with a face-tracking algorithm [3].

In this paper we use azimuth and elevation angles to describe source position, while yaw, pitch, and roll refer to three-dimensional head rotations.

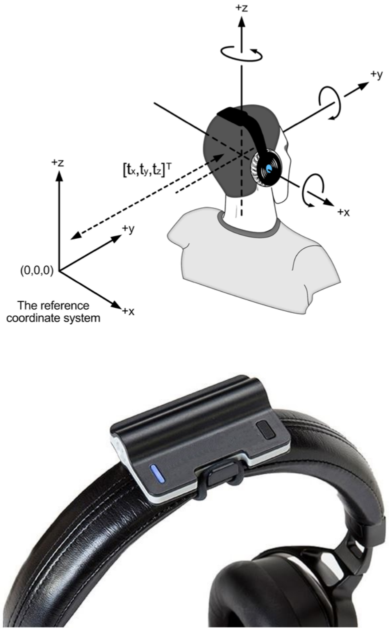

Figure 2. Head tracking: coordinate system (above), our experimental IMU sensor for tracking (below)

Real-time audio processing with head tracking is challenging: the audio must be adapted in real-time to the head rotation in yaw, pitch and roll, and in some cases, also to the three-dimensional position in the room. This requires heavy real-time processing, which modifies the spatial filter several tens of times per second, and which applies the up-to-date filter at very low latency to the audio signal. If the rate of update drops below about 30fps, or if the tracking delay increases, the listener might notice a mismatch between their head movements to the sound [3].

Since the listener may be travelling, flying, train or car riding, walking, jogging, cycling, or just walking around at home, a major challenge comes up: it is not clear how to adapt the head tracking to the movements of the listener. Does the listener, while moving, expect the sound sources to stay in their absolute original (virtual) position? For example (see figure 3), if a pop song places a vocalist right in front of the listener, and the listener is on a train, making a slow left turn, does the listener expect the vocalist to be heard from the right after the train completes the turn?

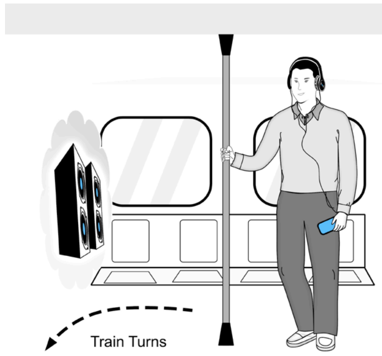

Figure 3. A listener in motion

The following sections of this paper will discuss the ideal listening set-up for music and its derivative for headphones, describe the NX-tracker algorithms for processing data from a head-mounted IMU sensor, and explain how to address the listener's movements in a variety of music listening use-cases.

2 Listening to Music

Music is an integral part of our lives and is consumed everywhere and played over a variety of reproduction systems. Surprisingly, we can enjoy a musical track just anywhere, despite of a big gap in quality and in spatial properties of different playback systems.

It is nevertheless very clear what recorded music is intended to sound like. Music production is an involved procedure, including usually a process of "mixing". In mixing, the musical creation gets its final shape, and the people behind the musical creation determine what the mix should sound like. In most cases, this is done over loudspeakers, in an acoustically treated professional studio. Mixing thus sets the golden reference for how a musical creation should sound. Unfortunately, only few people get to hear the music this way.

In order to bring the mix-room experience to the listener wearing headphones, a possible approach, called "virtual loudspeakers", is to emulate, in signal processing, the acoustics of a virtual mix-room, with a virtual ITU multi-channel set-up of loudspeakers. Then, in an adaptive manner to the listener's head pose, we approximate the sound wave that would have arrived at the listener's ears in a real room. As mentioned in the previous section, for each virtual loudspeaker as a point source, we need to emulate the room reverberation resulting from that loudspeaker, as well as the ITD and ILD for a sound wave arriving from the loudspeaker to the listener's ears (see figure 4).

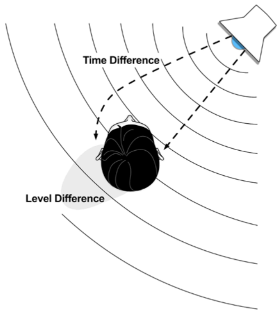

Figure 4. ITD and ILD

We notice that, with the virtual loudspeakers approach, two room reverberations are reproduced, one on top of the other. The first belongs to the acoustics of the room captured by the microphone, during the music recording session, and the second belongs to the emulation of the mix-room acoustics. While this combination is against intuition, the authors believe that this is the correct approach, because it represents what the artists heard in the mixing studio while producing the music.

With this approach, we follow the intents of the original artists, in whether they want us to feel like "being there in a concert hall", or like bringing the musicians to our home (known as the "they are here" experience, see figure 5). The latter is only possible because, with headphones on, we are not exposed to the acoustics of the physical space (room or vehicle) we are in, but to the virtual acoustics of the mix-room.

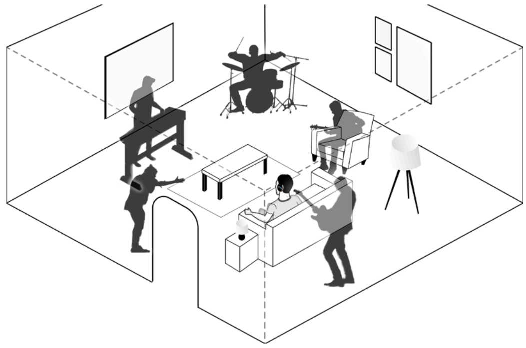

Figure 5. The "they are here" experience

In an earlier work [2], the authors developed NX, a virtual mix-room for audio-engineers and a virtual music-listening room for consumers. NX places virtual loudspeakers in a virtual room (as in figure 1), using ITU 5.1/7.1 set-ups optimized to music listening [5].

Using real-time digital filtering, the proposed method is to apply a set of linear filters, simulating the linear combination of HRTFs, room acoustics, and directional frequency-responses of the loudspeakers. In a mixing room with N loudspeakers, if each loudspeaker X_i is considered a point-source, then to simulate the complete listening setup over headphones, including a specific seating position of the listener and their head pose, the signals $Y_{L,R}$ at the left and right ears respectively are given by:

$$Y_L(t) = \sum_{i=1}^{N} \left(HRTF_{L,i} * HA_{L,i} * HS_i * X_i(t) \right) \tag{1}$$

$$Y_R(t) = \sum_{i=1}^{N} \left(HRTF_{R,i} * HA_{R,i} * HS_i * X_i(t) \right)$$

Where * denotes convolution, $HRTF_L$ and $HRTF_R$ are the left/right HRTFs, HA_L and HA_R are the room acoustics filters to left and right ears respectively, and HS is the filter of the loudspeaker, equal in both ears. For a listener in motion we need to apply these filters in real-time, adaptively to the head pose.

For a listener far enough from the mix-room loudspeakers (see figure 6), HS can be measured in advance. Since the listener may be in a moving vehicle, it becomes also useless to adapt HA_L and HA_R to the listener's position, and it is enough to adapt the room reflections and the HRTF in real-time to the head orientation in yaw, pitch and roll.

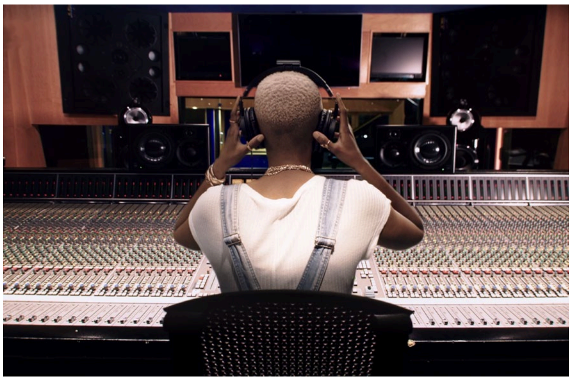

Figure 6. A typical mixing studio setup

We now consider the reproduction of a virtual ITU 5.1 surround loudspeakers system [5] over headphones. In the ITU 5.1, loudspeakers are at equal distance from the listener and at zero elevation. The center loudspeaker is positioned right in front (azimuth=0). The front left and right loudspeakers are positioned in azimuth +/-30 degrees respectively, and the rear left and right loudspeakers are positioned in azimuth +/-110 degrees. The additional subwoofer (the ".1" channel) can be positioned where the low frequency response in the virtual room is mostly flat.

Consider a listener sitting still in a living room, wearing headphones and a head-mounted tracker, measuring the instantaneous head orientation. When looking forward, the listener hears a center-channel virtual-loudspeaker in front. To keep the center channel in the center, if the listener's head turns D degrees to the left or to the right, we need to compensate our processing by rotating the virtual loudspeakers and the simulated room reflections by D degrees in the opposite direction.

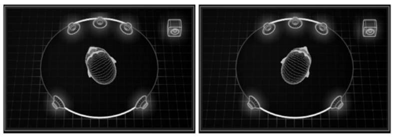

Figure 7. virtual ITU 5.1

But what if the listener changes a seat in the real living room, now facing away from the center channel source? If the head tracker works in absolute coordinates (not relative to the listener), then we need to re-calibrate our reference coordinate system. Otherwise, the listener would now be hearing the virtual rear-surround loudspeakers as playing from the front. One way to solve this is to ask the listener to activate a calibration process once switching seats. The head tracker would then register the head orientation in the moment of calibration as angle zero. But what if, while performing the calibration, the listener looks to the left? The calibration would then go wrong for the rest of the listening session. We thus prefer an algorithm to perform the calibration in an automatic manner. For example: the algorithm could detect the moment when the listener looks forward, or the moment when the head is at rest.

According to [8], the approach is to estimate, from the head-tracking data, which pose changes are due to head motion and which changes are due to torso motion. This estimation is described as particularly complex, given the large number of use-cases.

We propose an alternative method to decide which direction to calibrate to, by using a second sensor. Such an extra sensor can be attached to the body of the listener, or a part of their smartphone, tracking the orientation of the listener's body or torso. If we could sync perfectly between the head tracker and the body tracker, we could subtract the body measurement from the head tracker's measurement. Unfortunately, wearing a body sensor adds some inconvenience to the music-listening experience. Such a second sensor may be conveniently found in the listener's smartphone, but we do not know how the phone is positioned with respect to the body.

In some setups we could use the mobile camera to track the head and body orientations (see figure 8), but a camera is only useful if:
1. The camera's FOV is covering the body
2. We know that the listener is not watching a movie, where the smartphone may move together with the head movements.

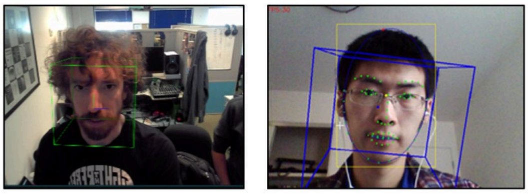

Figure 8. External camera tracker

A smartphone equipped with a user-facing camera and a built-in IMU sensor can become such an external camera tracker, if the IMU is used to compensate for the movements of the phone itself, and if the listener is kept within the camera's field of view. However, we suggest that, if we connect our processing to a smartphone with a built-in IMU, the camera may not be required at all. As will be shown in the next section, it is possible to use the mobile's IMU sensor alone as a second sensor, even without assuming anything about the orientation of the smartphone itself.

For computing the reference direction to calibrate to, it is useful to apply a long-term averaging to the head tracker's orientation, representing the average look-direction (see figure 9), and to calibrate to that average. In most cases this average will represent the direction where the listener expects to hear the front stereo channels in the music.

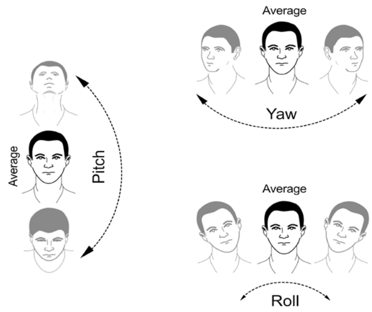

Figure 9. Average look direction.

Consider now a listener on a moving vehicle, with a head-mounted tracker to detect the head orientation. As long as the vehicle moves in a straight line, the listener's experience is similar to that of the listener in the living room. Once the vehicle starts turning, we need to re-calibrate the tracker's coordinate system. One of the questions is thus when to re-calibrate the reference? In a sharp turn, the moment to re-calibrate is clear, but it is less clear if the vehicle is following a very slow curve. The head-tracking algorithm does not know in advance which kind of turn it is going to be. It cannot distinguish between head motion and vehicle motion. The proposed solution in this paper involves a continuous re-calibration. This solution keeps calibrating as long as the vehicle keeps turning.

To measure the turning curve, the proposed solution uses a second sensor, coupled to the vehicle itself or on the person's smartphone, measuring the movements of the vehicle in order to adapt the reference system. The second sensor alone, however, does not account for other types of movements during travelling, such as looking through the window.

Consequently, for the moving vehicle use-case (see figure 10), our tracking algorithm should calibrate simultaneously to the vehicle movements and to the average look direction of the head-mounted head tracker. The look direction averaging time must be long (tens of seconds) when the vehicle moves straight, and short during the sharp turns. This concept is behind the proposed NX tracker algorithms, as given in detail in the next section.

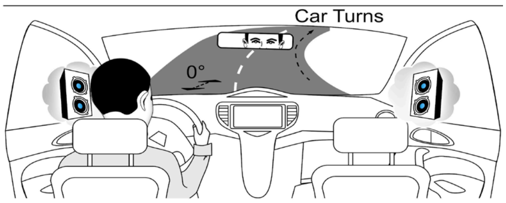

Figure 10. A listener in a moving car

Consider further a person listening to music while jogging. When running, the body moves up and down, and the head moves in all directions - not only horizontally (yaw), but also vertically (pitch). Sudden head movements are frequent, and the feet hitting the ground add further instantaneous acceleration to the head sensors, resulting in an increased drift error to IMU sensors. It is essential to re-calibrate the reference system frequently and continuously, as could be concluded by examining the rapidly moving mobile sensor in the runner's pocket. We don't turn off the head-tracking completely. For better realism during small head movements, it may still be worthwhile to use head tracking for binaural rendering even while jogging.

There are extreme cases where we prefer to calibrate the reference direction immediately and repeatedly. This is equivalent to the head tracking turned off momentarily. For example, when the listener turns their head rapidly to look back, or when a car makes a U-turn. In these cases, binaural audio with head tracking becomes a liability more than an advantage.

3 Algorithms for Motion Compensation

Camera Head Tracking
It is possible to use a video camera with image-processing to track the position and orientation of the listener's head.

While it is many times impractical to use a camera during music listening, a camera can be useful during movie watching or computer-game playing, when the listener is in front of a screen, looking mostly forward. This use-case allows an integrated camera to capture the head movements.

A camera tracker can provide more information than is usually provided a head-mounted IMU sensors:
- Position relative to the room and to the camera
- Complete head-pose information in six degrees of freedom, including orientation in yaw, pitch, and roll, and position in X, Y, and Z.

With a frontal camera, aligned with the screen being watched, there is no need to calibrate for a reference direction more than once. Angle zero is always when the listener looks right into the camera. X and Z offsets do not need calibration either, while Y-axis, representing forward and backward movements, does need calibration to the camera's field of view and to the listener's head diameter. In other cases where the camera is installed off-axis, one can still calibrate the angle differences in advance.

The additional information of X,Y,Z head position relative to the camera, can be used to enhance the music listening experience over headphones. If the camera is attached to a smartphone, tablet, or notebook, we can place two virtual frontal loudspeakers in the same position of the device's physical loudspeakers (for example on the sides of the screen). This can be done by controlling, as a function of Y, the gain and the delay of the simulated direct audio path with respect to the gain and delay of the simulated reflected sound.

In a similar manner, we can use X,Y,Z positional information relative to the room to place virtual objects in the physical room. For example, we can place virtual loudspeakers in preferred positions in the listener's living room.

The authors implemented a camera head tracker (figures 7,8) on a laptop computer, tracking in six degrees of freedom. Head pose was estimated at 30 frames per second by fitting a 3D head model to a detected face, and tracking the movement of the fitted points. In an experiment, we placed two virtual loudspeakers in the same directions (with respect to the listener) of the laptop's loudspeakers. We used Y data to adapt the room filters to the distance between the head and the loudspeakers. In an informal test it was demonstrated that, some listeners wearing headphones were unable to tell whether the music was coming from their headphones or from the built-in loudspeakers of the computer in front of them.

NX-Tracker – Dual Sensor Algorithms

As previously discussed, for a listener in motion, it is important to keep track of the listener's frame of reference, so that music is played from the listener's front. This can be achieved by using tracking data from an additional sensor located on the listener's body, such as the IMU sensor in a smartphone.

As we have already concluded in previous sections, the desired reference direction is computed as a time-average of the head orientations (average look direction). This idea stems from the fact that, most of the time, the listener would be looking in the same direction. The listener expects the music to be played around that average direction. With an average look direction, listeners can still look around and move their heads freely, and these short movements all cancel out in the averaging, not affecting the reference direction.

So how do we combine the averaging of the head tracking with the data from a second sensor? In our solution, the second sensor is used to figure out the duration of averaging, hence how far back in time we should look at to compute the average. For example, for a listener sitting on the couch and placing the smartphone on the table, the second sensor inside the smartphone would be very stable, so the averaging will be taken over a very long time. On the other hand, if a listener is walking and taking a turn, the head orientations from before the turn are irrelevant to the average reference direction. In this case, the second sensor can indicate that an abrupt direction change occurred, and the algorithm will compute the averaging over a shorter time. Once the turn is complete and the sensor stabilizes, the averaging time becomes longer again (figure 11).

For averaging the first sensor's direction measurements, we use a spherical linear interpolation function (slerp), described in [9], in the following recursive manner:

$$Q_{ref}[n] = \mathrm{Slerp}\,(\,Q_{ref}[n\text{-}1], Q_{head}[n], \mathrm{Adjust}[n]\,) \quad (2)$$

Where Qref[n] is the new reference direction, expressed in quaternions, computed as a function of the previous reference Qref[n-1] and the current sensor data Qhead[n], and where Adjust[n] is the rate of adaptation of Qref by the recursive slerp, from 0 to 1, where closer to 1 is faster. Adjust[n] is a function of the stability of the second sensor, the less stable – the faster we adapt the reference.

To make this work in different scenarios a statistical measure of the sensor data can be constantly learned. The averaging time can then be proportional to the amount of deviation from the learned measure (figure 12). For example, one such measure can be the variance of the IMU acceleration data in each axis. This way, when the second sensor is a smartphone in the pocket, the back and forth movements of the leg are learned and therefore not considered as instability. When the listener turns, leg movements also change the pattern, which no longer matches the original statistical measure. This causes the averaging time to shorten until the statistical measure is updated and the averaging becomes longer again. Similarly, on a moving train, constant vibrations can be learned and accounted for, so only if the train makes a turn the averaging time shortens.

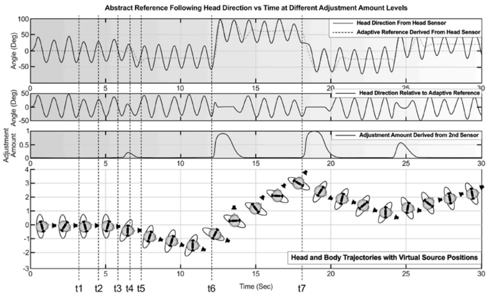

Figure 11. Dual-sensor moving reference behavior for a walking person

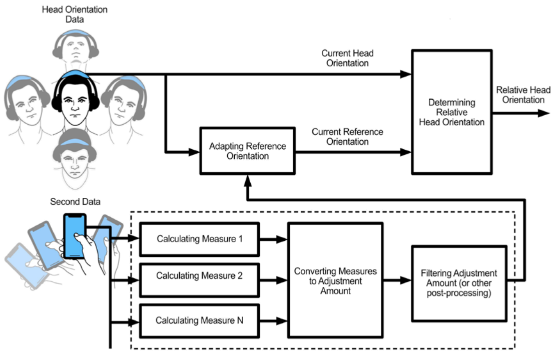

Figure 12. Dual-sensor algorithm diagram

To evaluate the proposed dual-sensor algorithm, the authors implemented the diagram of figure 12 in a smartphone application, connecting to an IMU head tracker (figure 2) via Bluetooth LE, and to the phone's built-in IMU sensor as a second sensor. We handed head trackers and software packages to five listeners, who were asked to complete a week of music listening on their mobiles at home, work, commuting, and sports. The listeners were asked to report their activities, as well as any events in which stereo or surround music did not play in the expected direction. All listeners reported the directions to match their expectations, except during activities with very frequent turns (walking the dog, dancing). To amend this, we recommend that whenever the first sensor detects excessive movements over time, the slerp's 'adjust' parameter is set to 1, so that the refence will follow the head tightly, effectively cancelling the head tracking effect.

NX-Tracker – Single Sensor Algorithms

It is possible to base a single sensor algorithm on the dual sensor algorithm, by using a single head-mounted sensor to replace both sensors in the dual-sensor algorithm in figure 12. The single sensor is firstly averaged as in formula (2), to be used as the first sensor, and the same data is then analyzed to derive the statistical measures for the reference system, instead of the second sensor.

As discussed, the dual-sensor paradigm described above is not based on the pure orientation of the second sensor, but on extracting statistical information from it. We use the data as indication for the need to adapt the reference by shortening or prolonging the averaging time. Due to this flexibility, this paradigm can be employed using only a single sensor, which is placed on the head.

On the other hand, the feature extraction for a single sensor is more difficult and error-prone because it must distinguish between a head movement and a body+head movement. To improve this, some heuristics may be added. For example: if the head is significantly far from the center for a prolonged time, it is reasonable to assume that a new reference should be adapted.

4 Conclusions

This paper presents a variety of real-word music-listening and movie-watching use cases, and analyzes the optimal experience in each, considering the professional studio mix as the reference. It explains how to deal with a listener in motion, and how to choose the reference direction for the music in binaural rendering.

Algorithms are proposed for calibration and adaptation of the reference coordinate system in response to head and body movements, with one and two tracking sensors. The proposed algorithms were implemented by the authors in software and hardware, as part of the NX and NX-Tracker. It is demonstrated that for practical use cases, two sensors can present an advantage in the music listening experience. In the absence of a second sensor, the algorithm can be adapted to a single sensor.

Further needed is a systematic subjective evaluation and blind testing of the proposed algorithms in this paper. In future researches, smart artificial intelligence applied to the motion data can be used to analyze inputs from several sensors. This would allow a classification of the exact use case, and to apply optimal tracking parameters to each case.

References

[1] Begault, D. R., "3-D Sound for Virtual Reality and Multimedia". AP Professional, Chestnut Hill, MA, 1994.

[2] Ben-Asher M., "3D Audio on Headphones: How Does It Work?", online paper, https://www.waves.com/3d-audio-on-headphones-how-does-it-work. June 2016.

[3] Shabtai N., Nehoran I., Ben-Asher M., Rafaely B., "Intelligibility of speech in noise under diotic and dichotic binaural listening". Applied Acoustics, Vol. 125, October 2017, pp. 173-175

[4] Ben-Kiki A., Ben-Asher M., Neoran I., "Improving Voice Intelligibility using Waves Nx". Online paper. https://maxx.com/wp-content/uploads/2019/03/Improving_Voice_Intelligibility_Using_Waves_Nx-Technical_Paper.pdf, September 2016

[5] ITU-R standard BS.2159-7 – "Multichannel sound technology in home and broadcasting applications"

[6] Carlile, S., "Virtual Auditory Space: Generation and Applications". R. G. Landes, Austin, TX, 1996

[7] Begault D., Wenzel E., Anderson M., "Direct Comparison of the Impact of Head Tracking, Reverberation, and Individualized Head-Related Transfer Functions on the Spatial Perception of a Virtual Speech Source". JAES Vol. 49 issue 10 pp. 904-916, Oct 2001

[8] Algazi, V.R., Dalton Jr, R.J., Duda, R.O. and Thompson, D.M., "Motion-tracked binaural sound for personal music players". AES Convention 119, October 2005

[9] Erik B. Dam, Koch M., Lillholm M., "Quaternions, Interpolation and Animation". Technical Report DIKU-TR-98/5, Department of Computer Science, University of Copenhagen, July 1998