Metrological reverse engineering of Microsoft Kinect's microphone Array

F. Mayer[a], M. Meissnitzer[a], M. Stadtschnitzer[b], B. Rettenbacher[b] und M. Fellner[b]

[a] *Technische Universität Graz, 8010 Graz, E-Mail: mayer@student.tugraz.at*

[a] *Technische Universität Graz, 8010 Graz, E-Mail: marlon.meissnitzer@student.tugraz.at*

[b] *Joanneum Research, 8010 Graz, E-Mail: michael.stadtschnitzer@joanneum.at*

Abstract

In the last few years sound and audio has become more and more important for developers and consumers in the gaming sector. Aware of that, developers are not only putting more effort and resources into the design and reproduction of sound, but also alternative approaches such as voice control are coming increasingly used.

Due to this, new gaming devices with voice and motion control (without the use of classic hand-controllers) offer a new, unprecedented gaming experience, most notably Microsoft's Kinect for XBOX 360.

With the release of Microsoft's SDK (Source Development Kit) for Kinect [1], developers are now able to use the Kinect and the contained microphone array for a huge variety of different applications [2] on the computer.

In this paper we concentrate on reverse engineering of Kinect's microphone array. Therefore metrological measurements are used to examine the microphone array's acoustic characteristics such as beam accuracy, impulse responses, transfer functions and the directivity pattern in order to grasp the possibilities and potential for new innovations and further projects.

Keywords

video games, gaming, microphone array, kinect

Introduction

The first electronic video games in the 1950s had no sound. In 1971 the first mass-produced video arcade game with sound *Computer Space* (Nutting Associates) was released. Since then audio in video games is very much developed, not least because of better hardware. Quantity and quality of sound increased and new formats and technologies come to use (see also Collins [3]). Nevertheless sound and audio in games played a subordinate role. Only in recent years it is becoming increasingly important, not only for developers but also for consumers. Therefore, more and more alternative approaches such as voice control are used. With Kinect for XBOX360 Microsoft also provides the player with the ability of voice control and therefore opens up an unprecedented gaming experience.

Thus it is exciting to see what has happened during the development of game sounds and which audio hardware is pretty cheap to buy nowadays. In this paper Microsoft's Kinect for XBOX360 will be analysed to show what is possible with current hardware. Therefore metrological measurements are used to examine the characteristics of the contained microphone array.

Measuring setup

In order to perform the measurement, Microsoft's Kinect was mounted onto a tripod. In the distance of 180 cm a drip-shaped speaker was placed in front of the Kinect.

For the following measurements the angle of incidence of the excitation signal had to be changed in steps of 10 degrees within the range of ±90 degree. To ensure the exact rotation of the microphone array the tripod was placed onto a rotary disc. To compare the measured signal and to equalize the influence of the loudspeaker, a reference microphone was placed above the array. Furthermore the reference microphone was aligned with the exact same angle as the Kinect consists of.

The center of the speaker membrane and the lower edge of the testing device were positioned 166,5 cm above the floor and the reference microphone was adjusted at a height of 172 cm. These measurement settings were determined with the use of a laser assisted level air. To reduce early reflections, caused by short signal distances to the floor, a molleton was spread out on the floor covering the testing area.

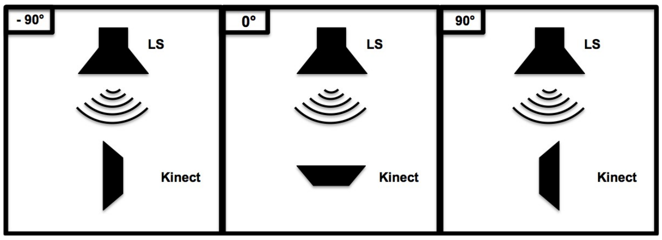

Figure 1: Shows the principal alignment of the test device to the fixed-positioned loudspeaker.

Determination of the beam accuracy

For determination of the Kinect's beam accuracy, three different types of acoustic signals (pink noise, square wave signal and speech signal) were used and played back via the loudspeaker. With these in use, the behaviour of the beam was monitored within three use-cases and different bandwidths of the input signals. The beam direction was determined and recorded using Kinect's SDK. Within this measurement the three types of input signals were used once each.

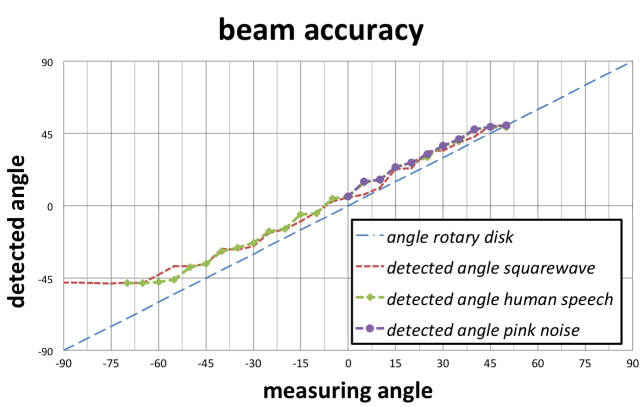

Figure 2: Beam accuracy of the microphone array; (squarewave signal, speech signal and pink noise).

Figure 2 shows a tolerably behaviour of all three excitation signals within the range of -50 to +50 degree arrangement to the speaker.
Comparing the results, pink noise and the speech signal are nearly even within the specific beam area, both having a constant offset to the angle of the rotary disk on which the test device was fixed. It turns out, that using speech signals obtain the best results, and therefore Microsoft's Kinect is suitable for voice-control. In the area within ±50 degrees, the detected beam of the array is directly hooked to the angle of the rotary disc, but leaving this specific beam area will cause a static beam angle at the beginning.
If the angle of the disk gets changed furthermore, the determined beam angle jumps through all defined values, which causes stochastic and not useable results.

Impulse responses, frequency response functions and directivity pattern

A logarithmic sweep with a frequency range from 20 Hz to 22 kHz was used to determine the performance of the impulse response. The signal was recorded once with use of the SDK and once with Audacity [4], both over the build-in microphone array and the reference microphone. The measurement was performed in 10 degree increments for angles of ±90 degree. The analysis and processing of the recorded signals was performed with Matlab [5].
The SDK saves the recorded data in the extensible wave file format. Due to this, the stored audio had to be encoded into the PCM file format, in order to work with Matlab. Thereafter the recordings were resampled to a uniform sampling rate of 16 kHz. Then the excitation sweep was windowed from the recordings. After this steps the frequency response functions (FRF's) were determined. From the resulting FRF's the impulse responses were calculated. The room reflections can now be filtered out of the impulse responses with an adapted HANN-window. Depending on the angle of incidence the shift of the individual impulse responses can be seen. This can be explained by the microphone positions and the time differences.

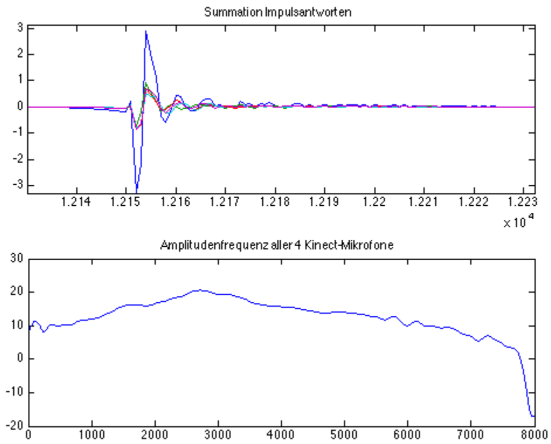

Figure 3: Summed impulse response and frequency response function at 0 degree.

The windowed impulse responses were then re-transformed into the frequency domain to obtain the windowed FRF's. To determine the directivity pattern of the microphone array, the 4 individual impulse responses were summed (Figure 3, above). After that, the sum of the impulse responses was transformed back into the frequency domain in order to obtain the total FRF (Figure 3, below).

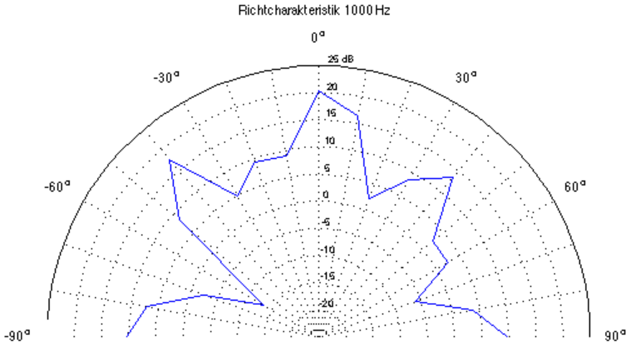

Figure 4: Directivity pattern at a frequency of 1000 Hz.

In the next step the FRF's were equalized by the influence of the loudspeaker. With the resulting FRF's the directivity pattern (Figure 4) was determined.

Conclusions

Gaming consoles and games, which can be controlled with human speech are in great demand these days. Microsoft's Kinect is a very innovative kind of gaming peripheral and can be used as a voice controller as well. For this reason, the Kinect has been reverse engineered in order to monitor its behaviour and to show up its primary use case. A measurement setup was built up which allowed changing the position of the testing device in steps of 10 degree each single measurement. At first, the accuracy of the beam detection has been determined with several signals played back over the loudspeaker. Furthermore the response of the exciting signal was recorded via the microphone array and the SDK. With the help of signal processing the signals were summarized and a directivity pattern was calculated.

References

[1] Kinect for Windows SDK, URL: http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx

[2] Rettenbacher, B. et al. 'Interactive Sound Source Localization using Natural User Interfaces and Mobile Devices', 4. Forum Medientechnik, Fachhochschule St. Pölten, 2011

[3] Collins, K. Game Sound (An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design), 2008

[4] Audacity, URL: http://audacity.sourceforge.net/

[5] Matlab, 1994-2012, The MathWorks, Inc., URL: http://www.mathworks.de/products/matlab