

Introducing the Zirkonium MK2 System for Spatial Composition

David Wagner
david@zkm.de

Ludger Brümmer
lb@zkm.de

Götz Dipper
dipper@zkm.de

Jochen Arne Otto
otto@zkm.de

ZKM | Institute for Music and Acoustics
Lorenzstr. 19
D-76135 Karlsruhe

ABSTRACT

The Institute for Music and Acoustics is a production and research facility of the ZKM | Center for Art and Media Karlsruhe. It is well known for the "Klangdom", a multi-loudspeaker facility for spatial sound diffusion with the aim to provide artists and composers with new possibilities. In this paper we present the overall revised and extended software solution for controlling the Klangdom, the Zirkonium MK2. Its origins in the previous version are briefly outlined and the advances are thoroughly described. Due to a very flexible client-server architecture, a hybrid spatial rendering engine and a very gestural trajectory editor it is already a useful toolkit for the institute's guest composers.

1. INTRODUCTION

In his utopian text "New Atlantis", published in 1627, Francis Bacon describes the prototype of a research institute and museum which comprises a department occupied with the processing and display of sound: "We have also sound-houses, where we practise and demonstrate all sounds, and their generation. [...] We represent small sounds as great and deep; likewise great sounds extenuate and sharp; we make divers tremblings and warblings of sounds, which in their original are entire. [...] We have certain helps which set to the ear do further the hearing greatly. We have also divers strange and artificial echos [...]. We have also means to convey sounds in trunks and pipes, in strange lines and distances." [1]

However, it was not before the 1950s that the technical means for the creative use of virtual sonic space in music were actively developed, with Pierre Henry and Jaques Poullin's "Potentiomètre d'Espace", the Poème Electronique of Le Corbusier, Edgar Varse and Iannis Xenakis, Jordan Belson and Henry Jacob's Vortex concerts from 1957-1959, and Stockhausen's premiere of Gesang der Jünglinge in 1956, which in a 1958 speech he conceived of as the attempt to include sound direction and movement as a new dimension of musical experience [2]. In this historically

prominent speech, Stockhausen likewise expressed the vision of spherically shaped music venues specifically designed for the presentation of spatial music, all around equipped with loudspeakers for an immersive presentation of sound. This conception was actually put into practice about ten years later as part of the German pavilion at the Expo '70 in Osaka, Japan. The "Kugelauditorium" spherical concert hall featured 50 loudspeakers together with equipment for the spatial positioning and movement of sound sources: a tape recorder carrying a control signal, and the so-called "rotary mill", a device for the live routing of a source signal along up to ten channels by the turning of a crank.

The immersive approach of the Kugelauditorium remained relevant for the further development of performance venues for spatial music, when in the early 2000s Ludger Brümmer and his research group at the Institute for Music and Acoustics (IMA) at the ZKM | Center for Art and Media conceived and implemented their Klangdom [3], extending the IMA performance and production space within the prominent ZKM Cube. This dome-shaped setup of 43 loudspeakers and four subwoofers however enables the direct positioning and movement of virtual sound sources by way of its dedicated software Zirkonium, making the Klangdom an advanced concert and production instrument for spatial music.

Moreover, the Klangdom and Zirkonium have been designed following an open strategy, bearing a number of practical issues in mind. Particular emphasis has been put upon basic functionality to enable both adaptability and extendability, but also to facilitate the creative process for users that aren't at the same time programmers. In particular, a fusion of the levels of sound and spatial composition should be scaffolded.

This general strategy has been further developed in the new, revised and extended version of Zirkonium, Zirkonium MK2, which is the subject of the remainder of this paper. For a recent overview of the Klangdom, see also [4].



Figure 1. The Klangdom in the ZKM Cube concert hall, with 43 loudspeakers and 4 subwoofers (©ZKM | Zentrum für Kunst und Medientechnologie Karlsruhe, Photo: Fabry)

2. ZIRKONIUM

For controlling the ZKM Klangdom the IMA is developing the free software Zirkonium since 2004. Its central aim is to simplify the use of space as a compositional parameter. Therefore positions and movements of sounds can be created and arranged in an event-based timeline or remote controlled using Open Sound Control (OSC). It is designed as a standalone application for Apple OS X and handles multichannel sound files or live audio. For the spatial rendering of virtual sound sources it uses Vector Base Amplitude Panning (VBAP) within a user-defined loudspeaker setup, or binaural filtering for headphones. When working with real speakers it is moreover possible to modify the size of a sound source by using a technique called Sound Surface Panning [3]. To avoid comb-filter effects a source can optionally be snapped to the nearest speaker. Zirkonium has been a reliable tool in a wide variety of productions and performances. It has been constantly enhanced with different developers involved which to a certain extent results in a patch-like source package.

2.1 Zirkonium MK2

In 2012 the IMA started reengineering the system taking into account the experience of the staff and guest composers. The result is a more stringent modular client-server based toolkit which comes along as a bundle of several applications and plugins, an extensive text and video documentation as well as a steadily growing collection of example patches and code snippets for creating custom control units in programming environments like Max or SuperCollider.

2.1.1 Motivation

The general necessity of a flexible system in composing spatial music is thoroughly discussed in [5] and [6]. The following text highlights certain aspects that coincide with experiences made with the original Zirkonium serving as a background for a description of new modules and features.

The variety of compositional and technical needs, especially in the area of computer music, requires a system

which can be easily modified by a programmer or the composer himself. This can be achieved by a more immediate access to the logical components of the software which in turn encourages the use of creative performance techniques including mobile devices, controllers and sensors.

These either permanent or project-based extensions of the software can cause a huge effort when being forced to break into someone else's code and understand the internal infrastructure. The client-server paradigm provides a remedy for this by enabling new ZKM or third party developers to write extending modules in their familiar programming languages and development environments which access the core functionality of the remaining system by networking communication. This was already considered in the original Zirkonium by means of the possibility to receive and send OSC messages for a fully externalized control of the spatialization. In Zirkonium MK2 this rather distributed approach is pursued even more severe which results in an isolated entity: the Spatialization Server (section 4). It is a Max-based standalone application which does all the audio related tasks like the handling of live inputs / soundfiles and the spatial rendering according to a desired speaker setup. When working with Max, which to a certain extent is a modular framework itself, it is much easier to re-use already existing software and libraries since its community is built on sharing. This also encouraged the development of a hybrid spatial rendering engine by use of modified versions of current VBAP and Higher Order Ambisonics (HOA) implementations. Simultaneous combinations of these techniques can be chosen according to their aesthetic properties. Furthermore it is optimized for the integration of several third party clients like the ZirkOSC plugin [7] or the Spatium interfaces ([5]). New project-based extensions can easily be linked with the system due to its distributed and open architecture. Another positive side effect is the simplified maintainability. The audio engine and spatial rendering system in the original Zirkonium is implemented with a version of the Apple Core Audio library that is becoming more and more outdated. To remain compatible with contemporary OS X updates it would be necessary to constantly revise a set of rather low-level functions. By using Max the audio functionality can be easily accessed with a few clicks while the respective bindings to the operating system and hardware updates are maintained by the creators of Max.

The original Zirkonium breakpoint editor pursues a relatively rudimentary approach for the positioning and movement of virtual sound sources: in an event-based timeline one can create spherical rotations with a fixed speed. Zirkonium MK2 contains a graphical Trajectory Editor (section 5) which uses quadratic Bézier splines for the definition of movements around the speaker setup but also for a modification of speed and acceleration along time. Furthermore it is capable of recording live panning instructions in the underlying representation. This has been a strongly demanded feature by many guest composers since it is a very intuitive and natural way of describing movements which can also be exported as compositional patterns. By maintaining the event-based data structure from

the classic Zirkonium old pieces can easily be imported by applying a resampling of the text-based rotational figures as Bézier splines which can be extended or modified just like newly created paths.

2.1.2 Architecture

Figure 2 shows the elementary architecture of the system and the possible combinations of the individual components that are represented by rectangular boxes. Each box is subdivided into two sections: the In and Out section. The different entities that are listed in the Out section of one component can be sent to another component if they are contained in the respective In section. Since the specific combinations strongly depend on the composer's intention and workflow the following text gives a more general overview of how the components are usually applied.

The Spatialization Server (section 4) receives live audio streams delivered by another software like a DAW by using Jack or Soundflower or an external source through the hardware audio interface or the network by services like netJack and the streaming audio project [8]. Furthermore it can be instructed to play back certain sound files by a custom controller client or the Trajectory Editor (section 5). The real-time spatialization is done with respect to specified loudspeaker setup (section 3) while the Server's output goes directly to the audio interface. The instructions of how to position and move certain sound sources can also be transmitted from an external client, from the featured Trajectory Editor or forwarded and exchanged between them to complement each other's functionality (section 8). A time stamp that keeps track of the ongoing piece can be exchanged between all clients to fire up certain spatial events or update positions according to a timeline.

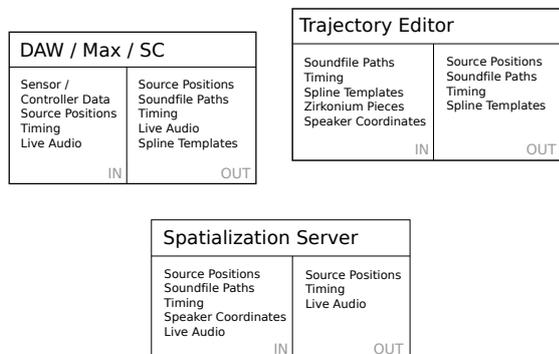


Figure 2. Possible combinations in a Zirkonium MK2 Setup

3. SPEAKER SETUP

The Speaker Setup application emerged from the original Zirkonium where it existed in a subwindow for explicit internal use. It enables the user to create and maintain different loudspeaker configurations which are corresponding to the studios where one is planning to work. At this point the advantage of an object-oriented encoding based on spherical coordinates becomes clear which is employed both in VBAP and Ambisonics. The spatial score which implies

sounds, their positions and movements is decoded in regard to a desired speaker setup which can be exchanged by one click. Thus the composer can present his piece in a different studio without having to prepare a new mixdown. Certain aspects like the individual dimensioning and room acoustics of a studio can make a piece sound different from the initial composing space (see [9] for a more detailed consideration) and should always be kept in mind when intending to relocate the presentation.

Speaker Setup (Figure 3) provides both a three-dimensional interface for a free-handed positioning of speakers as well as numerical tables to register either spherical or rectangular coordinates. By default it contains several sets of loudspeaker coordinates for basic layouts like a quad or octophonic and the studios in the ZKM including the Klangdom. These presets can either be modified or a new one can be started from scratch. The tool also maintains output patching presets which are associated with one or several speaker setups. Once a setup has been successfully created it can be exported to XML to be re-used in the previously mentioned Spatialization Server or the Trajectory Editor. Such a setup can also be created with a regular text editor since the XML-Scheme is also part of the documentation.

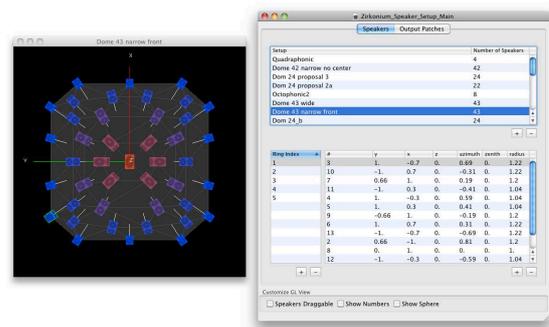


Figure 3. User Interface of the Speaker Setup application

4. SPATIALIZATION SERVER

The Spatialization Server is a standalone application based on the Max runtime that can be considered as the core of the MK2 system. It is responsible for every realtime-audio related task such as soundfile playback or recording, handling live audio streams and applying the spatial rendering of virtual sound sources. It is designed to be remote controlled by means of the MIDI-, OSC or TCP/IP protocols and provides an adjustable threedimensional visualization of the ongoing piece.

4.1 Everything at a glance

The front end to the Spatialization Server (see Figure 4) enables the user to organize the entire infrastructure of the respective piece in one window. A tweakable 3D-visualization in a secondary window can be used to monitor the moving sound sources in relation to the used speaker setup. The main window is subdivided into three

sections providing level meters for all incoming and outgoing audio signals concluding with a collection of buttons and fields which provide access to every configurable feature of the Spatialization Server.



Figure 4. Main window of the Spatialization Server

4.2 Generic Presets

The User Interface design of the Server including the overall color scheme and types of the respective buttons, sliders and level meters is inspired by the very appealing set of spatium renderers which are thoroughly introduced in [5] and can be accessed in the open source domain [10]. These renderers consist of different OSC- or MIDI controllable standalone applications for Ambisonics rendering or Amplitude-Panning of live inputs, sound field recordings or stereo sound files respectively. The spatium-ambi renderer for instance provides configurable soundfile- and Ambisonics related settings and a fixed amount of 16 live inputs and different pre-declared exchangeable loudspeaker setups.

To extend this limitation of pre-defined in- and outputs the Spatialization Server makes use of the JavaScript capabilities of Max which are provided by the *js* object. Thus the amount of inputs can be freely chosen without any logical limit. Depending on the power capacity of the user's computer this maximum supported number is limited physically and can vary in each scenario. The amount of outputs is determined by the created speaker setup and thus is also assigned dynamically. The same counts for direct outs. This in- and output configuration as well as the spatial rendering assignment can be saved into a preset which might refer to one or several pieces.

4.3 Hybrid Spatial Rendering

One of the key features of the Server is the possibility to simultaneously process incoming sound sources with different spatial rendering algorithms. Each channel of the Input section can be spatialized with VBAP according to [11] with the Sound Surface Panning extension described in [12], a mixed order Ambisonics proposed in [5] or sent to a dedicated direct out right next to the Output section. Furthermore a channel can be declared as one of the four B-format channels coming from a soundfield microphone recording. The idea behind this mixed approach is to enable the composer to use the desired rendering algorithm and its acoustic properties as a stylistic device. Concrete aesthetic qualities of spatial technologies are described by

Marije A.J. Baalman in [13] where she applies Ambisonics for giving the recording a rather "spatial and diffuse impression" or Wave Field Synthesis (WFS) for projecting sound inside the listening area and thus creating "a very intimate effect".

Advances have also been made in the sound quality of the binaural filtering for headphone use. It applies the FFT-based fast convolution with impulse responses from the CIPIC HRTF database [14] as proposed in [15].

4.3.1 Performance

Pieces for the Klangdom typically comprise about 16-32 sound sources. Of course this number strongly depends on the composer's intention, workflow and base material. For our eight core Mac Pro model from early 2008 a number of 50 sound sources is a realistic upper limit for VBAP / Ambisonics rendering.

Before actually working with the system composers tend to be worried about the performance limitation of the spatial rendering although it rarely remains an issue further on. Experience has shown that an increased amount of different sound sources can not only complicate the composer's work but also make it harder for the listener to perceive and distinguish individual sound sources and spatial gestures.

The binaural filtering is more demanding in this regard with a maximum of about 10-12 simultaneous sources. For efficiently deploying the available CPU power an arbitrary number of input channels can be excluded from the binaural processing.

4.3.2 Scalability

The system is not primarily designed to pursue a distributed processing approach. However, due to its network oriented communication and synchronization the performance limitations can be exceeded by deploying the software on separate computers. The outgoing audio signals which represent the real-time result of the spatial rendering merely have to be joined before going to the speakers. This can be achieved on the hardware side by means of an appropriate mixing interface or on the software side by applying a networked audio streaming.

5. TRAJECTORY EDITOR

The Trajectory Editor is a document-based application client which provides a very intuitive and gestural user interface for the spatial distribution of sound along time. It can either be for the live recording or construction of automations as cubic Bézier splines with a set of graphical tools. It provides a very powerful remote access to the the Spatialization Server and can be used in numerous scenarios (also see section 8).

5.1 Terminology

The underlying data structure and the corresponding terminology was mostly inherited from the original Zirkonium. Virtual sound sources are considered as IDs which can be gathered in Groups. Both can be the target of spatial Events which are arranged in a timeline and determine

movements or modifications of a parameter called Span which describes the extension of a sound source along the surface of the underlying speaker setup.

Keeping this data structure simplifies the import of original Zirkonium pieces which can be extended with the new set of tools or partially re-used in a new context.

5.2 The Loudspeaker Canvas

Figure 5 shows the user interface of a Trajectory Editor document. On the right side the previously mentioned data entities and their relationships are configured in text-based table entries while on the left side the user mainly operates on an interactive canvas while creating a spatial score. This canvas is a two-dimensional plain with a desired loudspeaker configuration as its background. It shows the speakers from above which enables the user to access every point of the semispherical surface of the dome. This view can also be altered between a spherical or planar representation while the first one implies a spherical distortion of a camera which would be positioned directly in front of the uppermost speaker. The second representation shows the speakers as if they were flattened in one plain which keeps more realistic proportions of the distances between the speakers.

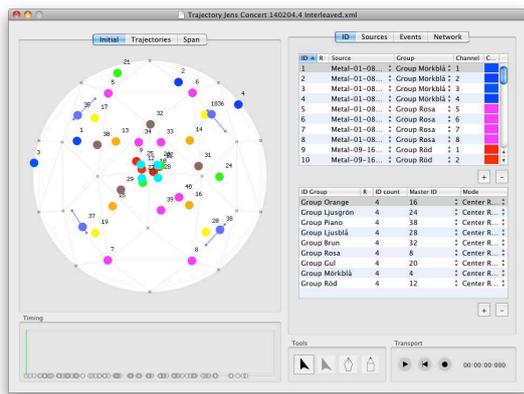


Figure 5. User Interface of a Trajectory Editor document

5.3 Global vs. local Event Context

The canvas view is subdivided into three different tabs which also set the lower timing representation into an individual context. The Initial tab is a more global overview of the entire piece - each ID and group can be monitored and highlighted while the individual splines of the actually processing events are hidden. It is also used to define the initial position for each ID or freely play with a certain target. If this tab is active the lower timing view is scaled onto the length of the entire piece and can be used to observe the progress which is represented by a running cursor or step into a certain point in time.

The Trajectory as well as the Span- Tab are both associated with a selected Event. They display the corresponding splines and the target as shown in Figure 6. The lower timing view switches to a local timing mode - it displays the

progress and acceleration within the current event along the time axis.

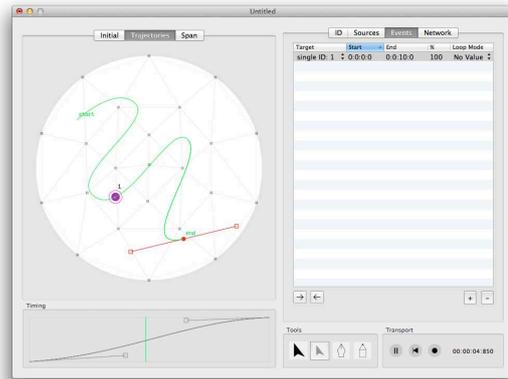


Figure 6. A Trajectory Editor document in the Event-based context

5.4 Create and modify Automations

For each time-based automation the Trajectory Editor uses cubic Bézier-Splines which can easily be created and modified with the toolbar on the lower right side of the document. These tools are inspired by contemporary software solutions for scalable vector graphics like Adobe Illustrator or Inkscape. The resulting graphical lines can freely be modified by accessing the individual spline elements or globally transformed by operations like shifting, x/y-scaling and rotation of all elements. By default each spline has a linear timing which means a constant speed determined by the duration of the event and its own length. This timing can be modified in the timing view right below the canvas. The x-axis of this view represents the duration of the event while the y-axis represents the amount of the spline which has been passed. Each automation can also be looped by an arbitrary percentage referring to the overall path. A further loop mode specifies the direction of the loop and the way its acceleration curve is considered. Hereby the acceleration can be applied for each loop segment or once across the entire looped event. Loops are a very easy and effective way of giving the piece a subtle vividness without the necessity of programming many individual movements. Besides constructing or painting trajectories it is also possible to record incoming live data into an underlying spline representation. The resulting trajectory including its acceleration curve can also be modified by the graphical tools described above. Splines can be exported and imported as Templates by a given XML interface.

5.5 External Synchronization

The Trajectory Editor has its own internal master timing which gets controlled by the transport section in the lower right corner of the User Interface. This timing however is only relevant if the Trajectory Editor is used to spatialize the playback of audio source files. If it is used for live audio streams it is possible to synchronize its timing to MIDI Time Code (MTC) and OSC-based timing messages.

6. PLUGINS AND CONTROLLERS

A very convenient and more and more established way of recording spatial movements in a direct relation to sounds and an underlying waveform representation is by the use of a DAW and the respective plugin interface ([5], [7], [16] and [17]). This way the variation of spatial coordinates either in spherical or rectangular representation can easily be recorded and played as an internal automation curve. Hereby each dimension of the coordinate (X, Y and Z vs. Azimuth, Elevation and Radius) gets its own parameter in the plugin and therefore its own curve which is corresponding to the desired track's timeline while the resolution of these curves and their type of interpolation strongly depend on the individual DAW. The advantage of this approach is that the spatial movements can be arranged just like the equivalent sound snippets and become part of the typical workflow in a DAW-based composition. A disadvantage lies in the fact that most DAWs and their respective plugin SDKs don't provide a time-independent access to the automation data which can only be modified with the internal tools curve- and parameter wise. Another disadvantage refers to the archiving of the piece - each piece and its possibilities of further modifications and access to the spatial score is tied to the specific DAW it was created in. Using the Trajectory Editor with external MTC synchronization provides a workaround for keeping the typical DAW workflow for the soundwise composition and having a full DAW-independent dataset for the spatialization. To remain flexible the open source Audio Unit plugin introduced in [7] is included in the Zirkonium MK2 bundle or can be downloaded under the given url. Other contents are example patches for Max, SupperCollider, the Lemur and TouchOSC as well as an Audio Unit / VST Plugin that hands the DAW's current timestamp to the Server if it doesn't support MTC sending.

7. EXCURSION: OPTIMIZING SOUNDS FOR SPATIAL PERCEPTION

A critical point in the field of spatial audio can be found in the complex interaction between sound qualities and spatial appearance in terms of ambiguity, localisation and distance. Our perceptive system is used to work with indifferant cues in the way that it creates an ambiguous impression. We are used to accept these ambiguities in real life, but composers seem, when working with spatial audio, to be obsessed with the fact that every sound should have a clear position in space. Especially when using low frequency signals it is important to learn that the spatialization technique in use cannot deliver a more distinguished outcome than we are able to perceive. Extending this observation it becomes clear that other qualities of sounds might influence their spatial perception. A sound without a percussive timbre envelope will always spread in space as opposed to sounds containing one or more high frequency impulses. The latter will clearly be perceived as point sources. The impression of distance is influenced by the amount and loudness of higher partials. Sounds with loud high frequency partials seem to be closer to the

listener than sounds with fewer high frequency partials. Listeners and composers seem to accept these facts when sounds with a real reference are used. But using artificial sound information these perceptive strategies of the ear and brain interact unintentionally with the different timbres and create an interpretation of a spatial scenery sometimes contradicting the initial intention of the composer. As a conclusion composers should know whether to blame the spatial reproduction system for those artefacts or the regularities of how our perceptual system localizes sounds differently based on their physical characteristics.

8. EXAMPLES

The field of application for MK2 reaches from tape music to interactive live performances. Within this context both the audio as well as the spatialization can either be prepared files or generatively created data streams. This leads to a variety of combinations which are benefitting from the flexibility of the system.

In context of the 50 year anniversary of the Elektronmusikstudion in Stockholm five Swedish musicians and composers Helene Hedsund, Lars Åkerlund, Lise-Lotte Norelius, Jens Hedmann and Eva Sidén played a concert in the ZKM Kubus in March 2014 which was also part of a live broadcast for a Swedish radio station. The pieces were prepared and played with the MK2 system, each in a different context. Hedsund pursued the pure tapemachine approach and worked on a Pro Tools session sending the audio streams to the Spatialization Server with Soundflower. The entire spatialization was done in the Trajectory Editor which was synchronized to Pro Tools by MTC. Åkerlund worked in a similar way with the difference that he also included a live electronics performance in the setup. As well as the IDs for the tape part the live inputs had representations in the Trajectory Editor and were included in the previously programmed spatial score. Hedmann and Sidén did the same thing with microphone inputs which were capturing sound from a piano and several metal objects. For an easier handling of the different sessions during the concert the Pro Tools tracks were consolidated and deployed as multichannel soundfiles which were played by the Spatialization Server. The pieces consisted from 12 up to 44 individual sound sources. Norelius used MK2 in her residency in a different way. A Max patch is generating live audio which is handed to the Spatialization Server by Soundflower. A part of the spatialization is created from pre-recorded data of movement sensors while the other part is created in the Trajectory Editor. The timing is also generated in the patch and sent to the Spatialization Server by OSC which synchronizes the Trajectory Editor with the progress of the patch. Norelius' piece will be presented at a concert in June 2014.

9. CONCLUSIONS

The MK2 system emerged from a rather conceptual state towards a set of tools that are actually used to play concerts on a regular basis in the ZKM Kubus. Since the software is free to use many composers take it along after their residency at the IMA and apply it in their private studios or institutions. Upcoming developments will focus on improving the current spatial rendering techniques and also incorporate an external Wave Field Synthesis rendering module. Thus the composers will be enabled to combine the rather surface-oriented VBAP and Ambisonics spatialization techniques with the aesthetics of sounds that are physically departing from the surface of the dome. One major challenge will be the revision of the compositional tools in regard to a distance encoding that also conforms with the actual acoustical domain. In context of the "European Art-Science-Technology Network" project new strategies to generate spatial gestures with the help of physical modelling systems like "Genesis" or "Mimesis" [18] will be implemented in the near future. Other efforts will lie in building synergies with other non-commercial parties in the field of spatial audio. By establishing compatibilities to environments with a different focus like Matthias Kronlachner's Ambisonics Plugin Suite [16] and its non-realtime capabilities a different workflow and set of aesthetics can be proposed to the guest composer without losing the advantages of the flexibility in the Zirkonium MK2. This will also be encouraged by taking into account established interchange formats for the description of spatial audio scenes like the SpatDif described in [19].

10. REFERENCES

- [1] B. Vickers, Ed., *Francis Bacon: The Major Works*. Oxford University Press, 2008.
- [2] K. Stockhausen, *Musik im Raum*, 2nd ed., ser. Texte Zur Elektronischen Und Instrumentalen Musik, 1. Aufsätze 1952:–1962 zur Theorie des Komponierens:152–75, D. Schnebel, Ed. DuMont Buchverlag Köln, 1988.
- [3] C. Ramakrishnan, J. Goßmann, L. Brümmer, and B. Sturm, "The ZKM Klangdom," *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, pp. 140–143, 2006.
- [4] L. Brümmer, G. Dipper, D. Wagner, H. Stenschke, and J. A. Otto, "New Developments for Spatial Music in the Context of the ZKM Klangdom: A Review of Technologies and Recent Productions," *Divergence Press*, vol. 3, 2014 (accepted).
- [5] R. Peñha and J. Oliveira, "Spatium, Tools for Sound Spatialization," *SMC*, 2013.
- [6] N. Peters, T. Lossius, J. Schacher, P. Baltazar, C. Bascou, and T. Place, "A Stratified Approach for Sound Spatialization," *Proceedings of the 6th Sound and Music Computing Conference*, pp. 2019–224, July 2009.
- [7] R. Normandeau, "ZirkOSC. Audio Unit plug-in to control the Zirkonium," 2013. [Online]. Available: <http://code.google.com/p/zirkosc/>
- [8] "A streaming audio system from the Sonic Arts R&D Group at CaliT2 UCSD." [Online]. Available: <https://code.google.com/p/streaming-audio>
- [9] G. S. Kendall and A. Cabrera, "Why things don't work: what you need to know about spatial audio," *ICMC*, 2011.
- [10] "Spatium, Tools for Sound Spatialization." [Online]. Available: <http://spatium.ruipenha.pt>
- [11] V. Pulkki, "Virtual Source Positioning Using Vector Base Amplitude Panning," *Journal of the Audio Engineering Society*, vol. 45, pp. 456–466, 1997.
- [12] C. Ramakrishnan, "Zirkonium: Noninvasive Software for Sound Spatialisation," *Organised Sound*, vol. 14, pp. 269–276, 2009.
- [13] M. A. Baalman, "Spatial Composition Techniques and Sound Spatialization Technologies," *Organised Sound*, vol. 15, no. 3, pp. 209–218, 2010.
- [14] "The CIPIC HRTF database." [Online]. Available: <http://interface.cipic.ucdavis.edu/sound/hrtf.html>
- [15] J. H. Andersen, "FFT-based binaural panner," 2014. [Online]. Available: <http://cycling74.com/toolbox/fft-based-binaural-panner/>
- [16] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," *Linux Audio Conference*, pp. 49–54, 2013.
- [17] F. Melchior, U. Michaelis, and R. Steffens, "Spatial Mastering - a new concept for spatial sound design in object-based audio scenes," *ICMC*, 2011.
- [18] C. Cadoz, A. Luciani, J.-L. Florens, and N. Castagné, "ACROE - ICA: artistic creation and computer interactive multisensory simulation force feedback gesture transducers," *NIME '03 Proceedings of the 2003 conference on New interfaces for musical expression*, pp. 235–246, 2003.
- [19] C. Miyama, J. C. Schacher, and N. Peters, "Spatdif Library – Implementing the Spatial Sound Descriptor Interchange Format," *Journal of the Japanese Society for Sonic Arts*, vol. 5, no. 3, pp. 1–5, 2013.