

# Zirkonium 3.1 - a toolkit for spatial composition and performance

Chikashi Miyama

miyama@zkm.de

Götz Dipper

ZKM | Institute for Music and Acoustics

dipper@zkm.de

Ludger Brümmer

lb@zkm.de

## ABSTRACT

*Zirkonium is a set of Mac OSX software tools to aid the composition and live performance of spatial music; the software allows composers to design multiple spatial trajectories with an intuitive GUI and facilitates arranging them in time. According to the provided trajectory information, the actual audio signals can then be rendered in realtime for 2D or 3D loudspeaker systems. For developing the latest version of Zirkonium, we focused on improving the aspects of usability, visualization and live-control capability. Consequently, a number of functionalities, such as parametric trajectory creation, additional grouping modes, and auto-event creation are implemented. Furthermore, ZirkPad, a newly developed iOS application, enables multiple performers to control Zirkonium remotely with a multi-touch interface and spatialize sounds in live performances.*

## 1. INTRODUCTION

The Institute for Music and Acoustics (IMA) at ZKM Karlsruhe, Germany, is dedicated to electroacoustic music, its production, and performances. The heart of the institute is the *ZKM\_Klangdom*. It is a 3D surround loudspeaker system, comprising 43 loudspeakers arranged in the form of a hemisphere. A crucial part of the Klangdom project is the 3D spatialization software *Zirkonium*.

Other major spatialization software, notably IRCAM Spat[1] or ambiX[2], places the primary emphasis upon spatial rendering, and most of them are provided as plug-ins or external objects for Max or Pd. By contrast, the main focus of Zirkonium is spatial notation and composition; Zirkonium provides electroacoustic composers with dedicated editors and tools that enables them to write scores of spatial movements precisely, intuitively, and flexibly. In order to pursue this end, Zirkonium is implemented not as a plug-in but as a standalone Mac OSX application.

Although the primal focus of Zirkonium is fixed-media composition, realtime capabilities of the software have been continuously extended from the beginning of the development in 2004. Zirkonium can be controlled by other software remotely via OSC. In this way it is possible to employ Zirkonium also for realtime applications. Moreover, the software is able to send “composed” spatial trajectories as OSC messages in realtime. It enables us to employ

Copyright: ©2016 ZKM | Center for Art and Media Karlsruhe. This is an open-access article distributed under the terms of the [Creative Commons Attribution License 3.0 Unported](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

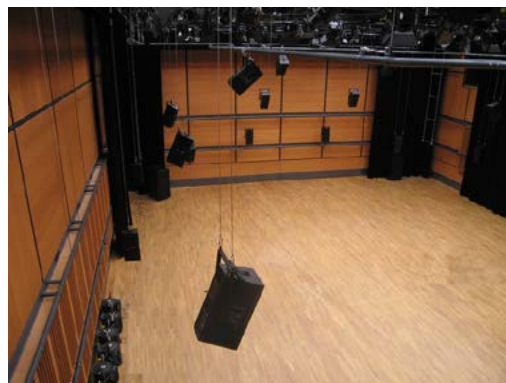


Figure 1. ZKM\_Klangdom

external software or hardware for the spatial rendering in addition to its internal rendering algorithms.

For the release of ver. 3.0 in November 2015[3], we improved the aspects of usability, visualization and live-control capability. After the release, multiple additional functionalities were implemented in the software. This paper briefly introduces the overview of Zirkonium ver. 3.0 and new features implemented for ver. 3.1.

## 2. ZIRKONIUM 3.1

In order to enhance the usability and the visualization, the software structure of the previous version was reassessed and redesigned. The latest version of Zirkonium, version 3.1 consists of three independent applications: *Trajectory editor*, *Speaker setup*, and *ZirkPad*. The *Trajectory editor* is the main application of the software package. It allows to draw and compose spatial trajectories and deliver actual audio signals to the hardware output. The *Speaker Setup* is a utility software that enables the user to define custom speaker arrangements and export them to XML files<sup>1</sup>. The *Trajectory editor* then imports the XML file and adjusts its spatial rendering algorithms to the speaker arrangement defined in the file. *ZirkPad* is a newly developed iOS App for the release of ver. 3.1. It allows the user to control the *Trajectory editor* remotely with the multi-touch interface.

### 2.1 Trajectory editor

The new *Trajectory editor* provides a superior GUI for designing audio trajectories and arranging spatial events in time (Fig. 2). Unlike the previous version, the *Trajectory editor* in ver. 3.0 is also responsible for processing actual audio signals from sound files and physical inputs in order

<sup>1</sup> Refer [3] for the detailed description of the *Speaker Setup Application* in ver 3.0.

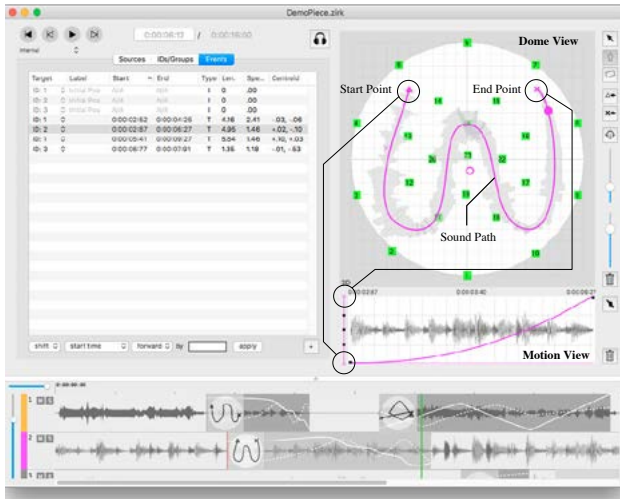


Figure 2. Zirkonium ver. 3.1 Trajectory editor

to execute spatial rendering for a maximum of 64 loudspeakers.

Figure 3 depicts the software architecture of the Trajectory editor. The Trajectory editor consists of three components: GUI, data management, and spatial rendering engine. For the new version, most of the GUI components are reimplemented with OpenGL and GLSL in order to conserve the CPU resources for the execution of the spatial rendering algorithms. The data management component processes all the data regarding spatial compositions. It has functionalities of importing Speaker-Setup XML files, and exporting spatial events to SpatDIF-XML files [4]. In the spatial rendering engine, the Spatialization server executes the spatial rendering algorithms and distributes the actual audio signals to each output channel. The spatialization server is entirely programmed with Pd (Pure Data)[5] and integrated into the Trajectory editor with the aid of *libPd*, a C Library that turns Pd into an embeddable library. This integration of the Trajectory editor and the Spatialization server does not prevent users from accessing the Spatialization server, running internally in the Trajectory editor. Advanced users with experiences in Pd programming are able to access the patch, modify the core spatial rendering algorithms, and apply arbitrary custom effects (e.g. reverb or doppler) to sound sources. Moreover, thanks to this integration the Trajectory editor is capable of accessing audio content more efficiently than the previous version. The new Trajectory editor offers various new features that take full advantage of this improved efficiency.

The following subsections introduce the most important additional functionalities, implemented in the Trajectory editor.

### 2.1.1 Trajectory Creation in ver. 3.0

In ver. 3.0, the graphical approach of trajectory creation, inherited from the previous version, is further enhanced.

In the Trajectory editor, a single trajectory (i.e. a movement of a virtual sound source in a specific time frame) is determined by a pair of paths; a *Sound path* and a *Motion path*. These two paths are drawn in two different views: the *Dome view* and the *Motion view*. The *Dome view* displays a space for spatialization, observed orthographically from

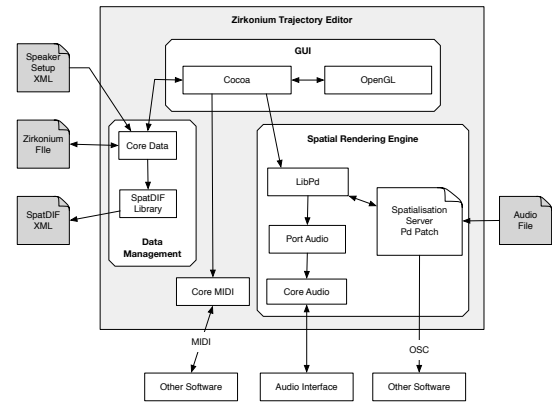


Figure 3. Software architecture of the Trajectory editor

the zenith. In this view, a Sound path, a geometrical route, that a virtual sound source moves along, can be drawn with Bèzier curves. The Motion view and the Motion paths, on the other hand, visualize how a sound source moves along a Sound path in a specific period of time. In ver. 3.0, the Motion path can be drawn with a multi-segment curve. The steepness of each segment is independently configurable by simple mouse operations and it controls the acceleration and deceleration of spatial movements.

Figure 2 shows a possible combination of a Sound path and a Motion path in the Dome view and the Motion view. In the Dome view, a meandering Sound path is defined. A virtual sound source moves along this Sound path from the start point to the end point, marked by a triangle and a cross symbol respectively. The X-axis of the Motion view indicates the time line of an event, and the Y-axis represents the relative position between the start and the end point of the Sound path. The start point (triangle symbol) coincides with the bottom and the end point (cross symbol) coincides with the top of the Motion view. In this way, the Motion view displays the relationship between time and relative position of a sound source, moving along the Sound path. In figure 2, an exponential curve is used as a Motion path. Thus, the sound source accelerates its speed towards the end point (cross). Moreover the waveform of the respective audio file is rendered along the Sound path and behind the Motion path in ver. 3.0. This feature enables users to grasp the relationship between the audio content and its position in space, and to adjust a certain audio content to a specific position in the Dome view.

### 2.1.2 Parameter-based Trajectory Creation

In addition to the manual drawing method with Bèzier curves, the software provides an algorithmic approach for Sound path creation. By entering a few parameters to the “add circle/spiral” pop-over panel, the software automatically draws a circular or spiral Sound path in the Dome view, employing the minimum amount of Bèzier curves required. These algorithmically drawn Sound paths can be further modified by mouse operations (Fig. 4).

### 2.1.3 Variety of Grouping modes

We speak of groups in Zirkonium, when several sound objects are moving together. This is a quite efficient way of working, since the movement has to be defined just once

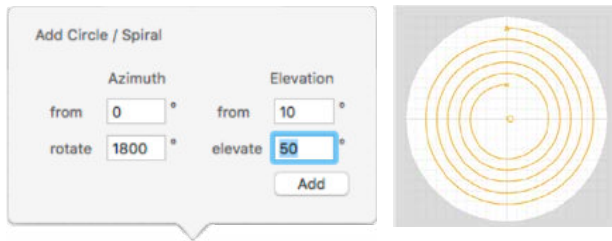


Figure 4. “add circle/spiral” pop-over panel and the result of parameter-based trajectory creation in the Dome view

for the group, and not for each group member separately. The concept of groups has been proven to be very useful especially if some spatial information is already included in the sound material, as it would for example be the case with stereo, quad or 5.0 material. For instance, we might position four sound objects in the form of a virtual quad in Zirkonium, define the quad as a group and move the group in Zirkonium as a whole, elevating or rotating it, while keeping the spatial relationship between the individual group members fixed. Now, if we have movement already included in the original quad source, we will get a superposition of the original movement and the movement created in Zirkonium [6].

It is easy for composers to generate quad or 5.0 source material with other software, since these formats are of course omnipresent and most traditional audio software delivers audio in these or similar formats. Thus, the described approach is a pragmatic way to create spatial complexity in the Klangdom and comparable systems. Another reason for choosing this approach could be the fact that it is not recommended to use more than about 30 sound objects in Zirkonium, partly because of performance issues, partly because of the risk to cause an unnecessary confusion for the user. There are, however, moments, where more than 30 different movements or positions of sounds might be desirable, for example when clouds of sound grains are to be distributed in the space. In these cases, the described approach, possibly using more than one group, yields typically very good results.

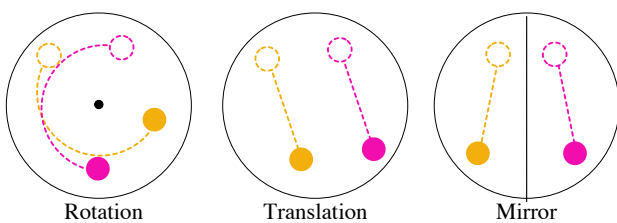


Figure 5. Three Group Modes

The notion of groups has been present in Zirkonium since the very first version. However, the treatment of groups has been considerably refined in ver. 3.1. There are three main group modes, the rotation mode, translation mode and mirror mode (Fig.5). The rotation mode has been available since the early days of Zirkonium. In the rotation mode all sound objects keep the same azimuth and elevation offset among each other. It results in rotational movements

around the Z-axis<sup>2</sup>, the line between the zenith and the nadir. This is a very effective way of moving a group within the Klangdom – it suits the Klangdom quite well since it has a spherical shape. It is especially suited for surround groups like a virtual quad, because the group stays centered automatically. A special characteristic of the rotation mode is the fact that the spherical distance between the sound objects decreases by increasing their elevation. If, for instance, a virtual quad group is elevated up to the top of the Klangdom, it will eventually end up as a mono source in the zenith. If this is not the desired behavior of a group, the composer can use the translation mode instead. Here the group is moved in parallel translation, which means, that the spherical distance between all sound objects belonging to the group stays always fixed. The third mode, the mirror mode, is especially useful for stereo sources, if their left/right orientation should be kept stable. In the mirror mode, all movements are mirrored against the Klangdom’s Y-axis, which is running from the front center to the rear center point.

For the rotation and translation modes there are two minor modes, so we get five different group modes in total. We call the minor modes “fixed” and “free” mode. They define how group movement and individual movement alternate. Generally a group can be deactivated and re-activated again at any moment in a piece. As long as a group is deactivated, its group members can be moved individually as separate sound objects. When the group is re-activated, the members of the group react differently depending on the minor mode. If it is defined as fixed, it regains its initial formation. If it is defined as free, it updates its formation to the current one.

#### 2.1.4 Improved Event handling functionalities

The more a spatial composition project evolves, the more spatial events are used. A powerful event handling tool would be indispensable for increasing productivity. In ver. 3.0, multiple new functionalities are implemented to enhance the efficiency of event handling.

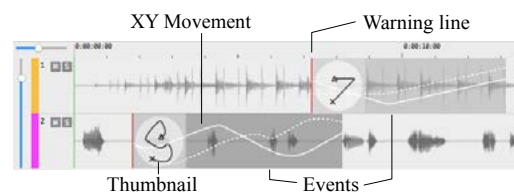


Figure 6. Event view

The Event view is a newly introduced GUI component in ver. 3.0. As shown in figure 6, this view visualizes the waveform of imported sound files and the spatial events assigned to each virtual sound source in the manner of a typical DAW software. In the figure, the waveforms of two sound files are displayed and a few spatial events are represented as transparent dark rectangles, superimposed on the waveforms. On top of these rectangles, snapshots of Sound paths, edited in the Dome view, are shown as thumbnails, and the chronological position of the sound objects is rendered as a solid and a dotted line, either in

<sup>2</sup> Zirkonium adopts the Spherical Navigational System defined in Spat-DIF v0.3 specification [7]

cartesian or in polar coordinates. This graphical representation enables the user to quickly grasp the distribution of trajectories in time. Besides, the Trajectory editor also offers the Event filtering panel (Fig. 7). This panel provides a way to automatically select specific events that match particular conditions. All selected events can then be shifted, scaled, copied or erased at once with mouse or keyboard operations.



Figure 7. Event filtering panel

### 2.1.5 Auto-Event Creation

As the name suggests, this functionality allows the user to create spatial events automatically based on the audio content of imported sound files (Fig. 8). Once this function is executed, the Trajectory editor analyzes the amplitude envelope of the selected sound file and estimates the start and end time of sounds in the file. Based on this estimation, the Trajectory editor creates spatial events for a specific ID or group. This functionality may significantly reduce the amount of time necessary for the event creation by mouse.



Figure 8. Automatically created events based on the amplitude envelope

### 2.1.6 Other features

For the release of ver 3.0, further functionalities, such as automatic event interpolation, event filter, reprogrammable spatialization server, HOA rendering algorithms, and Spat-DIF export functions are implemented. Refer [3] for details.

## 2.2 ZirkPad

ZirkPad is an iOS application for iPad, that remotely controls the Trajectory editor in realtime and allows users to control the movement of sound objects with its intuitive multi-touch-based user interface.

### 2.2.1 Motivation

There is a long tradition of sophisticated live-diffusion methods or systems, notably Acousmonium[8] or BEAST[9]. Most of these approaches are primarily specialized in the diffusion of a stereo signal to a large number of loudspeakers. However, they are not perfectly suited to the diffusion of a larger number of input channels. ZirkPad attempts to overcome this limitation by utilizing Zirkonium's traditional "object-based" approach, which gives the performer more intuitive control over the positions and movements of sound sources, as opposed to the conventional "channel-based" approach.

This is not the initial attempt to spatialize sound in real-time with the Zirkonium. In fact, live-diffusion has been possible and exercised since the first release of Zirkonium in 2006[10]; tentative live diffusion tools, that control the Trajectory editor remotely, were occasionally implemented with Max/MSP, Pd, or SuperCollider for each specific composition. However, the main focus of Zirkonium development always lied on the production of fixed-media composition.

Along with the release of ver. 3.1, we now attempt to provide a universal tool for live-diffusion beyond one-time usage, so that the user can get accustomed to the interface by practicing and working with it on a long-term basis.

Another significant advantage of the iPad is its mobility; the user can freely move around in the performance space. This is especially valuable during sound checks and rehearsals, where the performer can test how specific sound movements are perceived in different parts of the listening space.

### 2.2.2 Interface

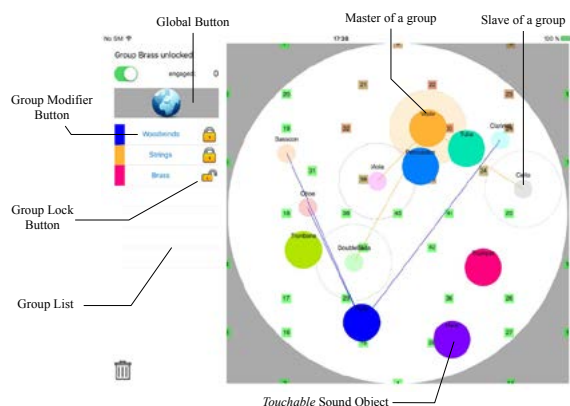


Figure 9. ZirkPad

Figure 9 shows the main interface of ZirkPad. On the right side of the screen, the multi-touch-enabled Dome view is displayed. It synchronizes with the Dome view of the Trajectory editor and visualizes the position of speakers and sound objects as well as the levels of the audio signals that each sound object generates and each speaker receives.

Unlike the Dome view in the Trajectory editor, the sound objects in the Dome view in ZirkPad are touchable; users are able to move single or multiple sound objects by dragging them with the fingers. As soon as ZirkPad recognizes touches on sound objects, it sends OSC messages to the Trajectory editor. In response to the received OSC messages, the Trajectory editor updates the position of the sound objects immediately.

A fundamental problem of live-diffusion with multiple sound objects is that we have got only 10 fingers and we are not able to move them completely independently, although we might want to control the position of more than 10 sound objects simultaneously.

An obvious solution would be to arrange the objects into musically meaningful groups. For example, we could assign 32 sound objects to four groups, each comprising

eight objects. In general, two to four groups seem to be a good quantity for live diffusion, being a good compromise between possible density and complexity on one hand, and clarity and ease of use on the other hand. ZirkPad offers two approaches to move a group of sound objects. The first approach is called *direct group mode*, the second *parametric group mode*.

In the direct group mode the user has to specify one master sound object per group. All remaining sound objects of that group are slave objects. When the master of a group is dragged by a finger, the slave objects follow the master, keeping the same azimuth and elevation offset to the master, in other words, they adopt the rotation mode as described above (2.1.3). The slave objects are displayed smaller in size, so the interface is clearly arranged and not overcrowded. The slave objects are not draggable themselves in order to prevent the user from accidentally moving them. However the user can unlock a group by tapping the respective lock button in the group list. Once a group is unlocked, its master and slave objects can be controlled independently from each other. When the group is locked again, it updates the spatial relationship or the formation of all group members to the current one. In other words, it follows the rule of “free” group mode as described in section 2.1.3. Thus the shape of a group can be changed in an easy way. This gives the user utmost flexibility in handling a group during a performance, while not suppressing the straightforward usage.

In the parametric group mode there is no direct control over the position of a sound object. Instead, the movement of the hand is mapped more indirectly to a movement of the group. With four fingers, the movement of the hand along the Y-axis of the screen controls the span or size of each sound object; with three fingers, the group is rotated; with two fingers, the elevation is changed; with one finger, the group is translated in parallel along the X- and Y-axis.

The parametric group controls can be applied to a single group as well as to several groups simultaneously. There is also a “global” button that activates the parametric controls for all sound objects at once.

It is also possible to establish one to many connections between the Trajectory editor and multiple ZirkPad running on several iPads, so that multiple performers can control a single Trajectory editor simultaneously. This feature is another means to control a large number of sound objects in live situations.

### 2.2.3 Use Cases

There were already several opportunities where we successfully employed ZirkPad in real-life situations.

The first concert took place on Dec 11th, 2015, in the ZKM.Cube concert hall, where the piece “Next City Sounds Hörstück” was being diffused by ZirkPad. It was a improvisational piece by the IMA, with five stereo channels coming from electronic instruments on stage. We used a single iPad with 10 individual sound objects without group mode.

The second occasion was a workshop at ZKM for high school students. It was especially interesting to see how quickly the students started to feel at ease with the interface and that they were able to use it in a convincing manner, even though those students did not have much experience

in musical performances previously.

Another very valuable opportunity was a concert of the Australian collective “Tralala Blip” on Jan 30th, 2016, in the ZKM.Cube. In this concert we successfully utilized ZirkPad with two performers, each controlling one group of sound objects.

## 3. CONCLUSION AND FUTURE PLANS

For the new release of Zirkonium, we focused on improving usability and live-control capability. We attempted to attain these objectives by revising the software structure, introducing the new GUI components, and implementing a novel iOS App for live spatialization. For the next development phase, enhancements in archive functionality, backward compatibility, and configurability of the hardware settings are planned. In addition, more algorithmic and audio-based approaches for trajectory creation will be further explored. Zirkonium ver. 3.1 public beta is available for free on the website of ZKM | IMA. <http://zkm.de/zirkonium/>. ZirkPad will be soon available from the App store.

## 4. REFERENCES

- [1] T. Carpentier, M. Noisternig, and O. Warusfel, “Twenty Years of Ircam Spat: Looking Back, Looking Forward,” in *Proceedings of ICMC*, North Texas, 2015.
- [2] AmbiX. [Online]. Available: <http://www.matthiaskronlachner.com/?p=2015>
- [3] C. Miyama, G. Dipper, and L. Brümmer, “Zirkonium MKIII - a toolkit for spatial composition,” *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59, 2015.
- [4] J. C. Schacher, C. Miyama, and T. Lossius, “The SpatDIF library – Concepts and Practical Applications in Audio Software,” in *Proceedings of ICMC*, Athens, 2014, pp. 861–868.
- [5] M. Puckette, *The Theory and Technique of Electronic Music*. World Scientific, 2007.
- [6] C. Ramakrishnan, “Zirkonium: Noninvasive software for sound spatialisation,” *Organised Sound*, vol. 14, no. 3, pp. 268–276, 2009.
- [7] N. Papters, J. Schacher, and T. Lossius. (2012) SpatDIF specification V0.3. [Online]. Available: <http://spatdif.org/specifications.html>
- [8] F. Bayle, “À propos de l’Acousmonium, Recherche Musicale au GRM,” in *La Revue Musicale*, vol. 394–397, 1986, pp. 144–146.
- [9] S. Wilson and J. Harrison, “Rethinking the BEAST: Recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre,” *Organised Sound*, vol. 15, no. 3, pp. 239–250, October 2010.
- [10] C. Ramakrishnan, J. Goßmann, and L. Brümmer, “The ZKM Klangdom,” in *Proceedings of NIME*, Paris, 2006, pp. 140–143.