

Tangible user interface for sound field control

Bachelor Thesis

Max Haider
Timon Rönisch

Supervisor: Ao.Univ.Prof. DI Winfried Ritsch

Graz, 2019



institut für elektronische musik und akustik



Abstract

Controlling the origin of sound sources in a three-dimensional audio field is a complex task, particularly in a live environment. Since multiple sources can be involved, a good overview, fast access to every source and simple automation commands are required. A user interface should facilitate straightforward handling and situational awareness. The following thesis presents the further development of “A Tangible User Interface for Playing Virtual Acoustics” by Birgit Gasteiger in 2010, with an emphasis on improved usability and advanced Ambisonic-controls and automation. Furthermore, it strives to define a field of application for the “Tangible User Interface for Auditory Virtual Environments” (TUI-AVE). Several controls for the spatialization of sound sources, the respective development as well as designs of the corresponding visual presentation are outlined.

Contents

1	Application Field	1
2	Auditory Room Control	2
2.1	Auditory Virtual Environment (AVE)	2
2.1.1	Historical References	2
2.1.2	Implementation	2
2.2	Sound Field Reproduction	3
2.2.1	Wave Field Synthesis (WFS)	3
2.2.2	Vector Base Amplitude Panning (VBAP)	3
2.2.3	Ambisonics	4
3	Tangible User Interface	5
3.1	IR-Tracking	5
3.2	Tracking Setup	5
4	Development Environment	6
4.1	Tableless Development	6
4.2	Experimentalstudio	6
5	Hardware Setup	8
5.1	Detection	8
5.2	Graphical Feedback	8
6	Software Setup	9
6.1	Components	9
6.2	ReacTIVision	9
6.3	TuioController	11
6.3.1	Coordinate Transformation	11
6.3.2	Notable Objects	12
6.4	TuiGraph	13
6.4.1	Graphics	13
6.4.2	Functionality and Usage	13
7	Controls	16
7.1	Solo	16
7.2	Volume	16
7.3	Trajectory Recording	17
8	Possible Improvements	18
8.1	Hardware	18
8.1.1	Lighting	18
8.1.2	Table	18
8.1.3	Camera	19
8.1.4	Projection	19

8.2	Additional Controls	19
8.2.1	Circling Sound Sources	19
8.2.2	Couple	19
8.2.3	Spatial EQ	19
8.2.4	Ambisonics Blur	20
8.2.5	Spatial Pre-Listening	20
9	Conclusion	21

1 Application Field

Apart from musicians, the most common actors when playing music for a live audience are the disc jockey and the video jockey.

The **DJ** (disc jockey) takes care of the selection, rearrangement and presentation of conserved audio [Bun05][Dud19]. The related field of application ranges from radio over studio-production to live performances at electronic music festivals or solo concerts.

The **VJ** (video jockey) introduces a visual dimension to acoustic presentation [Wik19]. This audiovisual connection varies between a support of the sound and the main attraction, which in turn is supported by the sound.

3D-audio can add a new artistic dimension to the performance or presentation of sound: the spatialization. Using 3D audio approaches, sound fields can be re-/created and manipulated [Art15] in order to achieve an auditory virtual environment (AVE). Since this task can require a lot of attention, it seems reasonable to dedicate it to an additional person.

In the current project the term space jockey, or short **SJ**, refers to the individual responsible for the spatialization of sound. The SJ contributes to the experience of the performance by including the three-dimensional space surrounding the audience, emphasizing an existing visual or acoustic impression, or forming a new way of artistic expression.

2 Auditory Room Control

2.1 Auditory Virtual Environment (AVE)

Auditory scenes generating acoustic input signals to the listeners' ears, that are creating the perception of pre-described auditory events (parametric controlled), are called auditory virtual environments. Not only perception at a single position but also interactivity (in the sweetspot) by the listener can be provided. [Bla05]

AVEs focus on the acoustic domain, nevertheless its output can be combined with visual and haptic domains to form an extended virtual environment.

2.1.1 Historical References

The invention of the radio at the turn of the 20th century can be seen as a step towards the evolution of AVEs, since it enables the listener to participate in events without physical presence but solely in an auditory way, yet without spatialization.

The first device able to control the spatial position of sound in a 3 dimensional room was introduced by Pierre Schaeffer in 1951. The "pupitre d'espace" (space device) enabled the spatialization of a virtual sound source generated of a mono signal within the space defined by 4 speakers. [Bre15]

Following inventions like the Rotationtable and the Halaphon represent the further development of sound spatialization. [Bre15]

Multi-channel loudspeaker reproductions systems in current consumer electronics (TVs, CD-, DVD-players) can be seen as pre-steps to virtual environments [Bla05]. Thereby, a trend towards an extensive presence of AVEs is shown.

Current technical possibilities, e.g. increasing computational power and advanced loudspeaker technologies, facilitate the development of a new generation of AVEs with impressively authentic 3D spatialization.

2.1.2 Implementation

Depending on the application there are mainly three different approaches of implementing AVEs [Nov05]:

- **Authentic reproduction of real existing environments** The virtual environment should evoke in the listener the same percepts that would have been evoked by the corresponding real environment. The audience should have the same spatial impression of their own movement inside the environment as well as the movements of sound sources.

- **Reproduction of plausible auditory events** This approach tries to evoke auditory events which the listener perceives as having occurred in a real environment. Here only those features are implemented which are needed for a specific simulation situation, respectively certain aspects of the environment that are intended to be emphasized.
- **Creation of non-authentic and non-plausible auditory events or environments** The virtual scene doesn't evoke percepts in the listener which are related to a real acoustic environment. For instance, paradoxical information about the virtual environment can be provided.

2.2 Sound Field Reproduction

To reproduce sound fields, several technologies have been developed. Currently the most prominent are wave field synthesis, vector based audio panning and Ambisonics.

2.2.1 Wave Field Synthesis (WFS)

Based on simplifications of the Kirchhoff-Helmholtz integral the WFS aims at the recreation of a sound field at the hole surface of its boundaries. As a result of this method a very high number of loudspeakers is needed for its implementation. Due to this expensive hardware setup, WFS is mostly implemented for 2D sound fields, since less loudspeakers than in 3D are necessary. [Spo04]

The recording- and playback setup of WFS are highly depended on each other. Therefore, sound field recordings cannot be used without the scene data to reproduce an auditory environment [Spo04]. As a result WFS is mainly employed in the synthesis of AVEs.

2.2.2 Vector Base Amplitude Panning (VBAP)

The amplitude panning is the most frequently used panning technique, e.g. stereophonie. In a stereophonic speaker setup, a virtual sound source, perceived by the listener, can be placed on the line between the speakers using amplitude panning on the stereo signals. [Pul01]

VBAP realizes this technique in a 2D and 3D sound field reproduction. In 3D sound fields, triangles defined by three speakers are used to create a virtual sound sources at any position in this triangle. An equal change of amplitude on all three signals can simulate the perception of a change in distance. The panning data for the positioning has to be generated in advance. There is no native abstract layer for VBAB's sound field representation. [Pul01]

In comparison to the other approaches explained in this section, more power on each speaker is needed, since less are used in parallel.

2.2.3 Ambisonics

The Ambisonics technique is based on spherical decomposition of sound fields. According to this decomposition, the Ambisonics channels contain directional information of the sound field. [Art15]

These channels can be decoded for various loudspeaker setups, since the decoder accounts for the amount of loudspeakers and their position. As a result, the encoding and decoding of Ambisonics can be performed independently.

The Ambisonics order defines the precision of the sound field reproduction and therefore the amount of available Ambisonics channels. The higher order Ambisonics format always includes all channels of all lower orders. As a deduction, all speakers are used parallel in the reproduction, since the omni-directional channel of the 0th order is always included. The reproduction is downwards compatible, e.g. a 5th order Ambisonics recording can be performed on a 3rd order playback setup accepting a loss in precision. [Art15]

This allows Ambisonics to provide a widely flexible sound field reproduction.

3 Tangible User Interface

3.1 IR-Tracking

The tracking method used in this project is based on the reception of infrared (IR) light. When working with optical tracking methods, which use visible spectra [Eva74], there is an evident interference with the surrounding ambient or working light. The IR-spectrum therefore provides more stability and flexibility in the surrounding light setup [Tuf17]. Although, it should be considered that there are common light sources like neon light, which still can interfere with tracking [UsH12]. IR-tracking requires relatively affordable hardware, since the current market for technical equipment provides many low-cost cameras with acceptable latency and video quality.

3.2 Tracking Setup

Fiducials are specially designed symbols allowing fast recognition of identity (ID), position and orientation [KB05]. The fiducial shown in figure 1 is part of the Amoeba set [Gas10] employed in this project.

The aim of this setup is to provide good conditions for the reception of the IR-light reflected by the fiducial on the object which is being tracked. Hence the IR-flashlights point to the lower side of the glass. A deflection layer prevents tracking of objects not touching the table. Subsequently, the camera receives the reflected IR-light and forwards it to the tracking software via USB 2. The tracking software processes the video stream of the camera. Objects are then compared with either the selected set of fiducial symbols or the adjusted finger size and forwarded as status messages via OSC [Gas10]. For a more detailed explanation of the tracking software see subsection ReacTIVision on page 9.

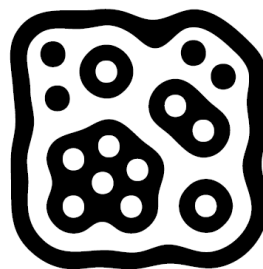


Figure 1: Amoeba fiducial (ID: 0)

4 Development Environment

4.1 Tableless Development

As a first stage of the development, a setup was established without most of the hardware. Table, projector, camera and Ambisonics sound system were not used in this setup. The *TuioSimulator*¹, based on the TUIO framework, provides a simulated input of the physical interface. It allows to send a fiducial's ID, position and orientation as well as a finger position in the same format as the subsequently utilized TUIO tracker *ReacTIVision*². This setup was used for developing the basic control structure and the graphics for the visual feedback.

4.2 Experimentalstudio

In order to further develop the sound control of the TUI-AVE table, it was necessary to work in a 3D audio environment. A location for this purpose was found in the *Experimentalstudio*³ of the IEM. Figure 2 shows the table in the *Experimentalstudio* during the development. Advantages mentioned in section 2 and its numerous implementations at the IEM led to the choice of Ambisonics for the realisation of the 3D audio.

The concept of the loudspeaker layout was based on the loudspeaker layout of the IEM Produktionsstudio. The Ambisonics hemisphere was arranged in three rings with a total of 13 loudspeakers, as shown in figure 3. Due to the scaffolding in the studio on which the loudspeakers were mounted, they were not equally distanced to the middle of the hemisphere. The resulting volume difference was compensated by adapting the gain of each speaker. The relatively small room size of the *Experimentalstudio* (27m²) results in near field effects within the Ambisonics hemisphere. A near field equalization was not performed, due to the early state of prototyping.

Software Configuration

Controlling the spatialization of several sound sources simultaneously was a major goal of the project. A software for encoding multiple mono signals in a sound field was needed. The *MultiEncoder* Plugin of the IEM Plug-in Suite provided this functionality. Additionally, this plugin facilitates remote control via network and was thus perfectly suitable to interact with the *TuioController*.

The Ambisonics decoder was designed using the *AllRADecoder* of the IEM Plug-in Suite⁴. For this calculation the loudspeaker positioning as shown in figure 3 was used.

¹https://github.com/mkalten/TUIO11_Simulator

²<https://github.com/mkalten/reactIVision>

³Studio of the IEM (<https://iem.kug.ac.at/services/raeume.html>)

⁴<https://plugins.iem.at>

Via *Jack* a routing between these plugins was performed before sending the output signals to the loudspeakers. For managing the plugins' settings the software *Carla*⁵ was used as a plugin container. This software provides the means to configure several plugins, save the configuration to a file and provide *Jack*⁶ interoperability.

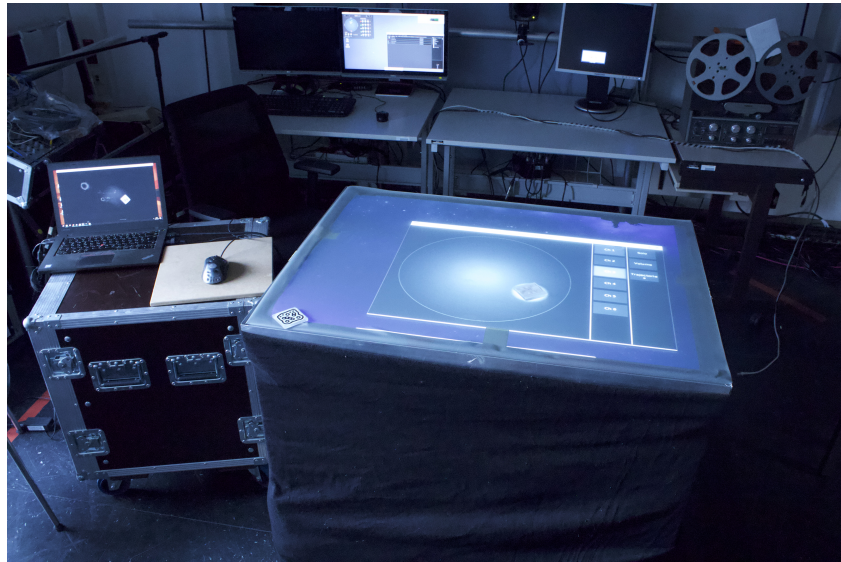


Figure 2: Overview of the Experimentalstudio

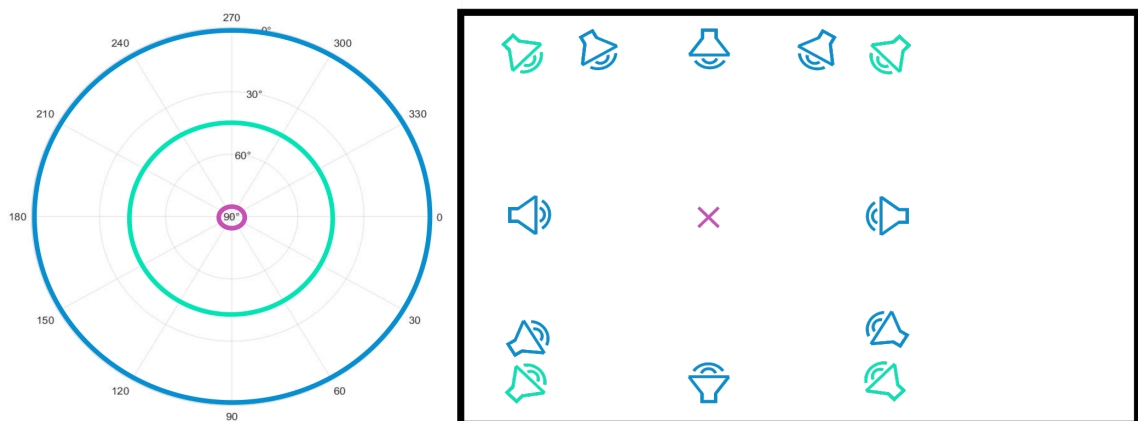


Figure 3: Left: Polar plot with elevation angle of the loudspeaker rings. Right: Spatial distribution of the loudspeakers in the Experimentalstudio from a bird's eye view.

⁵<https://kxstudio.linuxaudio.org/Applications:Carla>

⁶<http://jackaudio.org/>

5 Hardware Setup

The TUI-AVE project's hardware setup started with a basic wooden table from the project by Gabriele Gasteiger [Gas10]. Along the rebuild and further development, the table was extended with an updated input detection and elements for graphical feedback.

Since the targeted hardware environment was an embedded device, VirtualBox was used to run a virtual machine with a Debian Linux distribution.

5.1 Detection

The IR-reception was performed by a Logitech C920 camera, an IR-pass filter, two IR flashlights and a deflection layer on top of the glass plate. Additionally a wooden chassis was designed, to keep the camera in the calibrated position. The IR-pass filter lens was placed in front of the camera to remove the light created by the projector or in the surrounding room. The described setup is shown in figure 4. In order to increase the quality of the IR tracking, the camera was disassembled, removing the built-in IR-cut filter lens and to adapting the hardware preset focus range.

5.2 Graphical Feedback

The graphical feedback was displayed via a Benq projector, a reflection mirror, a wooden mount and a frosted glass plate. The mount is used to fix the projector to a calibrated position, as shown in figure 4. Furthermore, the mirror allows a smaller distance between the projector and the glass, since a greater image distance can be created by optical reflection. To serve as a projection surface, the glass plate is frosted.



Figure 4: Hardware setup

6 Software Setup

6.1 Components

The issue of controlling, automating and displaying a position by tracking a fiducial was split into three separated components which were addressed in the TUI-AVE project:

- Tracking
- Processing
- Graphics

To preserve the independence and the possibility to outsource parts to a single board computer (SBC), each of the three components were implemented in separate software applications.

A comparison of several tracking softwares led to the choice of *ReacTIVision*, given its support of fiducial and multi-touch finger tracking. For processing, the patch *TuioController* was created with PureData⁷. To be able to display a graphical representation of the acoustic situation, the application *TuiGraph* was created.

Communication

To keep a high level of interoperability, interchangeability and consistency all communication between the software components was implemented using the protocol Open Sound Control (OSC)⁸. *ReacTIVision* uses OSC by default to communicate the status messages of the tracking. For *PureData* a library⁹ was added to allow for receiving OSC data from *ReacTIVision* and sending data to *TuiGraph*. An overview of the communication is shown in the signal flow graph in figure 5.

6.2 ReacTIVision

ReacTIVision is an open source, cross-platform computer vision framework for the fast and robust tracking of fiducial markers, as well as for multi-touch finger tracking. It was mainly designed as a toolkit for the rapid development of table-top tangible user interfaces (TUI) and multi-touch interactive surfaces, based on the TUIO protocol [KB07]. This framework has been developed by Martin Kaltenbrunner and Ross Bencina as part of the *Reactable* project, a tangible modular synthesizer¹⁰. With these possibilities, *ReacTIVision* met all requirements of a tracking software in this project.

⁷PureData is an open source visual programming environment developed by Miller Puckett (<https://puredata.info/>)

⁸https://de.wikipedia.org/wiki/Open_Sound_Control

⁹<https://github.com/reduzent/pd-mrpeach>

¹⁰<https://github.com/mkalten/reactIVision>

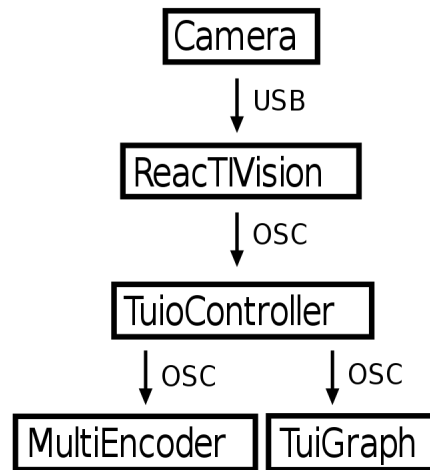


Figure 5: Signalflow graph

ReacTIVision outputs three kinds of messages for finger and for fiducial tracking: **add**, **update** / **set** and **delete** / **remove**. This is contrary to the original implementation of the TUIO protocol, in which **set** and **alive** messages were used to prevent possible tracking loss [KB07].

Fiducial tracking messages:

- add: fiducial ID, initial position and orientation.
- set: fiducial ID, position, orientation, several calculated values such as differential speed etc.
- remove: fiducial ID

Finger (“cursor”) tracking messages:

- add: cursor ID
- update: cursor ID, position
- delete: cursor ID

Fingers are detected as dots of specified size. Since the tracking software is thereby unable to distinguish between different fingers, each detection is numbered in ascending order. The expected approximate size for a finger needs to be configured.

The fact that the fiducial’s ID is fixed to a physical object is one of the main advantages of this tracking method. Therefore, a constant connection between a localized position of a certain audio channel and the physical object can be defined.

During the calibration of *ReacTIVision* in the presented tracking setup, a trade-off be-

tween the stability of finger and fiducial tracking unexpectedly had to be made. The calibration of the video signal from the camera only allowed for stable processing without package loss for either fiducials or fingers. Due to the aspired linkage between the fiducials and the spatialization the fiducial tracking was prioritized and a compromising calibration for sufficient finger tracking was found. The finger tracking was therefore used for navigation, whereby errors by tracking loss seemed acceptable.

The employed *ReacTIVision* Version 1.5.1 was transmitting 31 frames per second on average.

6.3 TuioController

The developed *PureData* patch *TuioController* receives and processes the user input sent by *ReacTIVision*. Afterwards the resulting position data is sent to the Ambisonics system for the spatialization and the data concerning the visual feedback to *TuiGraph*. In the patch all the elements responsible for the behavior of the interface are centralized to facilitate clear and fast adjustments.

PureData was chosen, because it allows for a high degree of flexibility, which is necessary for the implementation of the controls and the prototyping. The development was performed in *PureData* version 0.48-1. All used Pd libraries are part of the project repository.

The input stream from the tracking software is first split up into data of fiducial and finger tracking. To simplify the processing of position data, a coordinate system transformation is performed. The resulting coordinates value from 0 to 1.

The data of the finger tracking contains an event ID and the position coordinates. The position is used for the selection of an input sound channel and the associated controls. To select a specific channel, the position data is compared to the calibrated coordinates of the control fields. If the position matches a control field, the assigned message is sent. The object *stateControl* provides the necessary stability and overview of the selection.

The tracked fiducial data is routed according to the fiducial ID and linked to the processing channels, similar to channels in a mixing console. A certain fiducial is responsible for one processing channel. The relation is hard-coded. These processing channels provide clear scheduling and structure. Along the channels the data passes through the control objects *write*, *read*, *volume* and *solo*. The functionality of these objects is described in section 7. Subsequently the position data is sent to the graphics application and, after another coordinate system transformation, to the Ambisonics encoding software.

6.3.1 Coordinate Transformation

The fiducial input data is given by the user putting the fiducial on the two dimensional representation of the hemisphere displayed by *TuiGraph* (see figure 6). The basic idea of this representation is a view from the aerial perspective on the hemisphere. In that

sense, the *elevation* angle in the spheric coordinate system is mapped onto the radius in the polar coordinate system. The azimuth is consistent in both systems. The areas with smaller *elevation* angle are of significant importance in the spatial perception of sound [Bla08]. But these areas would be displayed at the outer areas of the representation with a lower resolution than the areas with bigger *elevation* angle. Therefore, the representation has to be distorted towards its middle point. The distortion function has to preserve the coordinates of the radius at 0 and 1 and compress all other values towards 0. A Matlab¹¹ script was written to simulate and animate this mapping. After testing trigonometric functions, a root function was used which provides the possibility of smooth adjustment by changing its argument.

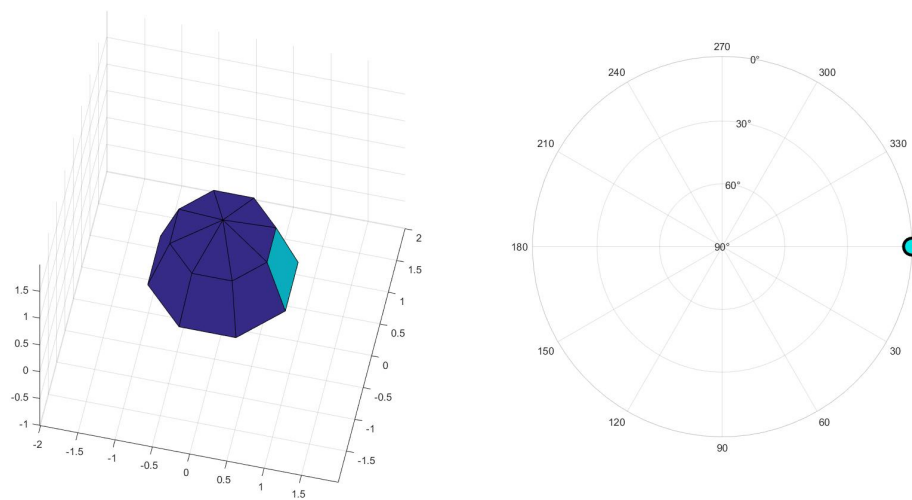


Figure 6: spheric-polar mapping

6.3.2 Notable Objects

The previously mentioned robustness against tracking loss of the original implementation of the TUIO protocol, was modeled in the fiducial data processing of the *TuioController* in the object *suppressLoss*. In this object a **remove** message and the following **add** message are blocked for a timeframe of 100 ms, if the **add** message is received in that same time frame.

Also, a set of debug and test objects were developed to simplify maintenance and further development.

¹¹Matlab is a software for matrix-based numeric calculation (<https://de.mathworks.com/products/matlab.html>)

6.4 TuiGraph

TuiGraph was written in Java, because this programming language is a high-level language and can be run on most of the common operating systems. The current graphical implementation library in Java is the *JavaFX*¹² library. It allowed for creating graphical elements using current software technologies. The Spring Framework¹³ was selected for receiving data from the *TuioController* and to facilitate dependency injection¹⁴.

6.4.1 Graphics

JavaFX allows to use a graphical designer to create the styling via CSS. The basic layout and visible elements can be defined using XML.¹⁵ Those elements can be manipulated while the program is running and also additional elements can be added.

This capability was used, to create a basic layout of three columns, one large column containing the circle and two smaller columns, one containing the signal sources and one the effects.

The defined layout can then be styled using CSS¹⁶, so layout and look are decoupled. As a result of the decoupling, the look of the UI can be changed without affecting the basic functionality. CSS is a well known and simple format, which allows to easily change the colors, shades and color transitions. This would allow for further development and experimenting with different themes.

6.4.2 Functionality and Usage

To be able to dynamically and easily change the behavior of the whole TUI-AVE, all logical decisions and computation remain in the *TuioController*. *TuiGraph* is only used as a simplified displaying tool. So the functionality of *TuiGraph* was reduced to:

- Display the position of a sound source
- Highlight a specific channel
- Record and display a trace of a specific sound source

Position

The main functionality of the graphical component *TuiGraph* was to display the position of a certain sound source. A hemisphere is therefore displayed as the main element,

¹²“JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms.” (<https://www.tutorialspoint.com/javafx/>)

¹³<https://spring.io/>

¹⁴https://en.wikipedia.org/wiki/Dependency_injection

¹⁵<https://en.wikipedia.org/wiki/XML>

¹⁶https://de.wikipedia.org/wiki/Cascading_Style_Sheets

and for each sound source a small circle is placed on the corresponding position of each source.



Figure 7: TuiGraph

Regarding communication, the coordinate system was kept identical to the cartesian coordinate system in *TuioController*, thus *TuiGraph* converts the coordinates to the display coordinates. To display a sound source on a specific position the following OSC message needs to be sent to *TuiGraph*:

```
\pos[id] ,ff [x] [y]
```

Between the address path `\pos[id]` and the x position is the type tag of the OSC syntax. The position address only accepts two floating point arguments.

As an example to display the sound source with the ID 0 on the coordinates (-0.1, 0.5) the message would look like this:

```
\pos0 ,ff -0.1 0.5
```

```
/newpage
```

Highlighting

A possibility to highlight a channel was added, to supply a reference, which channel is currently controlled or selected for adding an effect. Each channel can be highlighted in 3 different stages:

- 0: no highlighting
- 1: subtle highlighting
- 2: well visible highlighting
- 3: full highlighting

To highlight a channel, an OSC message to the address `\highlight[id]` needs to be sent, containing the desired level as an integer argument:

```
\highlight[id] ,i [level]
```

E.g. highlighting the channel 0 would be achieved with the message

```
\highlight0 ,i 2
```

Recording

TUI-AVE is able to record a trace of a signal position and replay this position, see Trajectory Recording in the section Controls. This record is also displayed as a line on the table. To keep the communication between *TuiGraph* and *TuioController* lightweight, the recorded data was not sent from *TuioController* to *TuiGraph* but instead also recorded also at *TuiGraph*. The duplication of this data could therefore lead to inconsistency, which was an accepted risk and might have to be reconsidered if problems in a future usage of the table occur.

To start recording a path for a channel with a specific ID, the recording address was created:

```
\record[id] ,i [0/1]
```

7 Controls

In general, the implemented controls aim for a flexible and versatile sound field control by the space jockey, while simultaneously preventing complicated usage. The controls are based on the possibilities of the previously described interface setup and the software performing the Ambisonics decoding.

According to the head-related coordinate system, the directional information is expressed by the angles *azimuth* and *elevation*. A third dimension is the radius as a representation of the loudness, as used in the *AllraDecoder* ¹⁷ to compensate unequally distanced loudspeakers. Therefore, possible controls in the present setup will relate to the positioning regarding the spatialization and the volume of the applied audio signals.

As shown in section 3 and section 6, the control quantities are the **fiducial's ID**, **position** and **orientation** next to the **finger position**, which is used only for selecting.

To simplify the assignment of the audio signals, the **fiducial IDs** were fixed to the input mono channels. This also seemed to provide a more intuitive handling of the controller.

The **fiducial's position** relative to the circle representing the hemisphere (see chapter 6.4) is mapped onto the position of the mono channel in the hemisphere of the sound field. Therefore the **fiducial's orientation** is a free control quantity.

The assignment of a control to an input channel during operation follows the scheme:

- Select active input channel, by touching the labeled field in the channel list (on the right side in figure 8, left list)
- Select control, by touching the labeled field in the control list (on the right side in picture 8, right list)

7.1 Solo

The control Solo mutes all other input channels, leaving the selected one as the only one playing. In the implementation of this control, there is no control quantity necessary, since its states are only *on* and *off*. For this reason, the state of the *solo* object in the *TuioController* toggles after selection, triggering the built-in solo function of the *MultiEncoder*. The default state is *off*.

7.2 Volume

To enable the adjustment of the related input channel's volume, this control can be used. In the *volume* object the change of the fiducial's orientation is linked to the volume control of the related audio channel in the *MultiEncoder*, if active.

¹⁷<https://plugins.iem.at>

The 360° of the fiducials' orientation are mapped onto the volume range of -60dB to +10dB of the *MultiEncoder*. The default is 0dB.

The degree and therefore the values for the volume are increasing with clockwise rotation. To prevent the output volume from jumping from -60dB to +10dB (when rotating the fiducial more than 360°) the object *diff_suppress* blocks output values with a difference of more than 20 to the last one that was not blocked. Thereby it enables the possibility to recapture the control by rotate the fiducial to its last processed orientation.

7.3 Trajectory Recording

The trajectory recording control supports the automation of the channel's spatialization and volume control. When selecting this control, the control list changes to the trajectory list. If a trajectory storage is selected, the values of the current position and orientation are sampled and stored in tables in the *write* object of the *TuioController*. These tables can be read out afterwards with the same sample frequency either in single or loop mode. The loop mode can be selected before or during the active reading. If the loop mode is switched off during reading, the last loop is performed till its end. This control enables the dynamic spatialization of several input audio channels simultaneously.



Figure 8: Usage of the interface

8 Possible Improvements

For the development of this prototype a lot of problem solving was involved. Some ideas and insights could although not yet be realized. For instance a noticeable latency between user input and graphical feedback was discovered. An investigation did not reveal the exact cause, but reducing the latency would improve the user experience. These ideas can be categorized into improvements of the hardware and additional controls.

8.1 Hardware

At this state of prototyping some improvements on the hardware setup did not seem reasonable yet, since a proof of concept was paramount. The following suggestions are mentioned to support any continuation.

8.1.1 Lighting

An advanced IR-light concept could be a starting point to provide more stability and equally distributed precision of the tracking. As described, there are currently two IR lamps illuminating the fiducials from the bottom of the table. This arrangement resulted in two major problems:

- Due to the angle of radiation of these lamps, some areas of the glass were either too bright or too dark to allow a reliable detection.
- The reflection of the lamps from the glass were very bright and created two spots, where no detection was possible.

Experimenting with the placement of the lamps, two possible (and combinable) solutions seemed possible. All solutions would require additional lamps.

- A shallower lighting angle resulted in less bright reflections, but also in a smaller illuminated area per lamp.
- Dispersing the emitted light with a milky sheet positioned directly at the lamp would result in a more even lighting and less reflections.

Additionally, the latency of the tracking process by the tracking software is depending on a stable and equally distributed IR illumination.

8.1.2 Table

For the continuation of this project, it could be desirable to design the table in a more flexible and transportable way. Length and width of the table, determined by the size of the glass surface, provide comfortable handling, as experienced during the development. The height of the interface could be shortened and a more precise folding mechanism could be implemented to provide an easy switch between transport and performance.

For a shortened height a camera with an appropriate wide angle lens would be necessary, as well as a short distance projector.

8.1.3 Camera

The reception angle of the Logitech C920 camera is too small to cover the whole area of the glass. Additionally, fiducials placed in the corners of the detected field of the camera were out of focus. A well selected wide angle lens for the camera could also solve this issue.

8.1.4 Projection

It would also be desirable to use a projector able to project a rectified image on the glass from the side of the interface. The glass and the deflection layer could be unified to one component, by choosing a frosted glass with the required properties of both components. Thereby, a less complex hardware setup could be realized.

8.2 Additional Controls

8.2.1 Circling Sound Sources

One idea, which was not fully implemented, was to perform a automated circular movement (or other preset movements) of a sound source around a given center point. Per default the center point could be the center of the hemisphere or a second fiducial. For initiating such a movement, the distance and speed could be measured and continued at the removal of the fiducial.

8.2.2 Couple

To be able to control the position of a stereo signal, a single fiducial could be used to define the spatialization of the left and the right channel. The position of the fiducial would define the center of the stereo couple and its orientation the rotation of the channels around it. The orientation of the fiducial could alternatively be used to modify the distance between both channel. This method could also be extended for a higher number of channels.

8.2.3 Spatial EQ

All spectral parts of a sound signal do not necessarily need to be on the same position in the AVE. With a series of bandpass filters a sound signal could be split up into multiple signals. Using the rotation of the fiducial, the position of these signals could

be increasingly spread out, e. g. in a linear way along the azimuth angle, or in a circle around the position of the fiducial. This implementation could support the synthesis of non-authentic and non-plausible auditory events or environments mentioned in section 2.

8.2.4 Ambisonics Blur

The Ambisonics order specifies the precision of the spatialization and thus how precise a source can be located, as mentioned in subsection 2.2. A lower Ambisonics order in the encoding of a sound channel increases the blur of its localization. The orientation of a fiducial could be used to alter the order. With this control the SJ has the possibility to change the expression from *blurry* to *clearly defined*.

8.2.5 Spatial Pre-Listening

A binaural decoder could enable the SJ to listen to a spatial arrangement of selected sound channels on headphones. Thereby, the possibility of experimenting before playing to the audience by a spatial pre-listening could be implemented. Requirements on flexibility and spontaneity during the performance could be met.

9 Conclusion

The general intention of the TUI-AVE project was to develop a user-friendly tool for the space jockey. Controlling the spatialization of sound in a live environment by this actor should be simplified by an interface.

The TUI-AVE controller prototype is a functioning 3D audio controller. Basic features for controlling the audio spatialization and its automation were implemented. Also, a clearly laid out graphical feedback was provided. The decisions concerning the software components and the hardware setup are a major part of this work.

Besides the considerations on hardware improvements mentioned in section 8 some concepts could be reconsidered.

The method of optical fiducial tracking as user input detection offers the major advantage of providing the fiducial ID as a fixed linkage to the control channels and the fiducials rotation as a flexible control quantity. On the downside, the latency experienced on the described setup did not convey the feeling of being fully in control of the auditory virtual environment. For the usage after any change on the hardware setup, a rather complex calibration remains necessary. Different approaches are shown by the *Reactable* product *Rotor*¹⁸ and by *Madrona Labs*' product *Soundplane*¹⁹. These could be alternative bases for the TUI-AVE controller.

To adapt any advancement on the controls or the graphical feedback, a profound evaluation would be necessary, preferably by individuals familiar with the subjects of spatialization or live performance.

All things considered, the controller cannot yet be described as a finished tool, but as a prototype with a lot of potential.

¹⁸<http://reactable.com/rotor/>

¹⁹<https://madronalabs.com/soundplane>

References

- [Art15] Daniel Arteaga. *Introduction to Ambisonics. Audio 3D - Grau en Enginyeria de Sistemes Audiovisuals*. June 2015. URL: <https://upfa3d.files.wordpress.com/2015/05/ambisonics.pdf>.
- [Bla05] Jens Blauert. *Communication Acoustics*. 2005. ISBN: 978-3-540-22162-3.
- [Bla08] Jens Blauert. *Räumliches Hören*. In: Weinzierl S. (eds) *Handbuch der Audiotechnik*. Springer-Verlag Berlin Heidelberg, 2008. ISBN: 978-3-540-34301-1. DOI: https://doi.org/10.1007/978-3-540-34301-1_3.
- [Bre15] Martha Brech. *Der hörbare Raum*. Transcript Verlag, 2015. ISBN: 978-3-8376-3096-1.
- [Bun05] Gericht: Bundesfinanzhof. *Urteil verkündet am 18.08.2005. Aktenzeichen: V R 50/04*. Rechtsgebiete: UStG 1993, Richtlinie 77/388/EWG. 2005. URL: https://judicialis.de/Bundesfinanzhof_V-R-50-04_Urteil_18.08.2005.html.
- [Dud19] Duden. *Duden: Discjockey, Diskjockey, der*. 2019. URL: <https://www.duden.de/rechtschreibung/Discjockey>.
- [Eva74] Ralph M. Evans. *The Perception of Colour*. 1974. ISBN: 978-0-471-24785-2.
- [Gas10] Birgit Gasteiger. "A Tangible User Interface for Playing Virtual Acoustics". Thesis. Institut für Elektronische Musik und Akustik, Kunst Universität Graz, Technical University Graz, 2010.
- [KB05] Martin Kaltenbrunner and Ross Bencina. "The Design and Evolution of Fiducials for the reacTIVision System". In: 2005.
- [KB07] Martin Kaltenbrunner and Ross Bencina. "reacTIVision: a computer-vision framework for table-based tangible interaction". In: Jan. 2007. DOI: 10.1145/1226969.1226983.
- [Nov05] Pedro Novo. *Auditory Virtual Environments*. Ed. by Jens Blauert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 277–297. ISBN: 978-3-540-27437-7. DOI: 10.1007/3-540-27437-5_11. URL: https://doi.org/10.1007/3-540-27437-5_11.
- [Pul01] Ville Pulkki. "Spatial sound generation and perception by amplitude panning techniques". PhD thesis. Helsinki University of Technology, 2001. URL: <http://lib.tkk.fi/Diss/2001/isbn9512255324/isbn9512255324.pdf>.
- [Spo04] *Wave Field Synthesis - Generation and Reproduction of Natural Sound Environments*. Naples, Italy, Oct. 2004. ISBN: 88-901479-0-3.
- [Tuf17] Terence Tufuor. *Using Active IR for eye detection and tracking*. ECE Senior Capstone Project. 2017. URL: https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/Violet_Tufuor.pdf.
- [UsH12] Eranga Ukwatta, Jagath samarabandu, and Mike Hall. "Machine vision system for automated spectroscopy". In: *Machine Vision and Applications* 23 (May 2012), pp. 111–121. DOI: 10.1007/s00138-011-0338-8. URL: https://www.researchgate.net/publication/233885124_Machine_vision_system_for_automated_spectroscopy.
- [Wik19] Wikipedia. *Visual Jockey*. 2019. URL: https://de.wikipedia.org/wiki/Visual_Jockey.