Visualization Research Center of the University of Stuttgart (VISUS)

Bachelorarbeit

Immersive Visual Analysis of Cello Bow Movements

Yannik Kohler

Course of Study: Informatik

Examiner: Prof. Dr. Michael Sedlmair

Supervisor: Frank Heyen, M.Sc.,
Xingyao Yu, M.Sc.,
Sebastian Rigling, M.Sc.

Commenced: January 21, 2022

Completed: July 21, 2022

Abstract

To support cellists in analyzing their interpretation of a musical piece and comparing it to others, this work presents and explores possibilities to visualize musical data and motion-captured bow movements in virtual reality. The presented prototype utilizes consumer VR technology and, optionally, the OptiTrack motion capture system to record a cellist's playing, providing different visualizations that can be viewed in real time as well as played back from recorded data. In this thesis, we propose a one-by-one mapping of the 3D path of the bow, supported by various synchronized 2D line charts that are placed in virtual reality. We map motion features, audio features, and computed metrics to the color and width of the respective segments of the 2D and 3D lines. To evaluate the proposed design a user study with two domain experts was conducted, showing potential use of the design and providing us with ideas for future improvements, such as an extension to augmented reality.

Contents

1 Introduction **11**

2 Related Work **13**
2.1 Musical Visualization . . . 13
2.2 Motion Tracking and Guidance Visualization . . . 13
2.3 VR/AR for Musical Tasks . . . 14
2.4 Motion Tracking of Violin Bowing . . . 14

3 Design **17**
3.1 Use Cases . . . 17
3.2 Tracking and Data . . . 17
3.3 Color Schemes . . . 18
3.4 3D Visualization . . . 19
3.5 2D Visualization . . . 22
3.6 Playback Two and Play Along . . . 26
3.7 User Interaction and Additional UI Elements . . . 27

4 Implementation **33**

5 Evaluation **37**
5.1 Methods and Participants . . . 37
5.2 Part 1: Live Visualization . . . 38
5.3 Part 2: Analysis . . . 39

6 Limitations and Discussion **41**

7 Conclusion and Outlook **43**

Bibliography **45**

List of Figures

3.1 Blue-Yellow-Red color legends. 19
3.2 Color legends for notes and line index. 19
3.3 Plain 3D motion path of the bow. 20
3.4 Plain 3D line, Filtered By Time. 20
3.5 Plain 3D line, Translated By Time. 21
3.6 3D line with different visual encodings. 21
3.7 3D line with different metrics in secondary line. 22
3.8 Assortment of multiple 2D line charts in VR space. 23
3.9 2D line chart with different spatial directions on the y-axis. 24
3.10 2D illustrations provided by our domain expert and corresponding visualization . 25
3.11 2D line charts showing velocity against time 26
3.12 Two 3D lines played back next to each other. 27
3.13 2D line charts with two recordings per chart. 27
3.14 Color legend and sheet music shown in VR. 28
3.15 Images showing our desktop UI. 29
3.16 Images showing our VR UI. 30
3.17 Image series demonstrating *moving and scaling UI components*. 30

4.1 Cello bow prepared for different tracking approaches. 34

List of Tables

3.1 Data summary from our previous paper [HKT+22] 18

1 Introduction

The violoncello (cello) is a four-string musical instrument that is played using a cello bow. The bow is usually held in the right hand, while the left hand presses down certain strings to play different notes. The precise motion of the right hand plays an important role in learning and mastering the cello, as unintended movement of the bowing hand and poor posture do not only harm the quality of the resulting music but can also potentially cause injury to the musician. Sheet music does not usually contain instructions on the exact bow motion for the piece, which leaves musicians with their own ideas on the bow motion, resulting in different musicians having different interpretations of the same musical piece. By analyzing their motion and comparing it to that of other musicians, cellists can work on their own interpretation, to potentially find new variations and improve their playing. Being aware of and recognizing the own precise movements can be difficult, and discussing and comparing them to other musicians' motions is usually even harder.

In related work, different visualizations have been used to provide additional information to musicians by augmenting sheet music with rhythmic and harmonic information [FMK+20; MBE19], and by showing self-similarity within a musical piece [Foo99]. The use of motion tracking and motion guidance is being explored for different domains such as physiotherapy [TYB+15], as well as to support violinists in their practice [BTR21; DVO+20; LSBJ11; PWG03]. While the related work suggests interest in utilizing motion tracking to provide information about musical data of bowed string instruments, none of the above provides immersive 3D visualizations similar to what we propose.

This work presents a design to support musicians in analyzing their play, by visualizing their motion as a 3D path in virtual reality (VR), supported by different 2D line charts placed in VR space. To do so, we capture the position and orientation of the cello bow and record this data over time along with the produced audio. We can track the bow either by attaching a VR controller to the bow or using the OptiTrack[1] motion capture system with small, lightweight infrared markers attached to the bow. Using our design, musicians can view real-time visualizations of their playing, play back the data of one or multiple recordings to analyze and compare, or play back data while recording simultaneously.

We evaluated the proposed design in a qualitative user study with two experienced cellists, who showed interest to use the design for self-analysis, as well as to guide the student's focus in a teacher-student scenario. The user study also revealed different limitations and provided us with new ideas for further improvements to the design. We consider extending our prototype to support augmented reality (AR) in the future.

[1] https://optitrack.com/

In a paper [HKT+22] released during the research for this bachelor thesis, we presented the concept of the prototype. The bachelor thesis adds information on the different visualizations and functions of the prototype, implemented before and after the release of the paper. Additionally, we conducted a user study for this thesis, evaluating the design with experienced cellists.

In summary, we propose an immersive visualization of cello bow motion in 3D space, by displaying an animated 3D path trailing the cello bow, with the user taking control over the playback speed, playback position, and the length of the time interval that is displayed. As the motion of the cello bow usually happens in a shallow, near two-dimensional volume, we can use the third dimension to encode time by translating the recorded positions by a distance depending on their time stamp. This way, our 3D visualization gives detailed views on relatively long periods of time, which would usually clutter up 2D visualizations. To provide additional information and undistorted perspectives to the 3D lines, we propose different 2D charts that can be viewed alongside and synchronized to the 3D visualization, by placing them in virtual space. To encode data such as velocity or different metrics, we propose using different visual channels, such as the color and thickness of the line.

In Chapter 2, we discuss different related works on visualization of musical and motion data and the use of motion tracking to assist violinists. Chapter 3 provides further information on the proposed ideas and use cases, as well as explanations on the design we derived from those ideas. Implementation details on our current prototype for that design are explained in Chapter 4. We evaluate our design by conducting a user study, which is described in Chapter 5. In Chapter 6, we discuss different limitations and results revealed during design and by the user study. Finally, Chapter 7 presents our conclusion, as well as ideas and plans for future work.

2 Related Work

In the following sections, we summarize research that inspired us or is similar to our work and research that forms the foundation of our design.

2.1 Musical Visualization

Khulusi et al. presented a survey [KKM+20] in 2020 in which they classify and analyze work related to music visualization from different scientific sources, commercial software, and public websites. The survey shows that there has been relatively little work in this domain so far, despite a range of applications for musical visualizations, some of which are discussed in the following.

The paper *Visualizing Music and Audio using Self-Similarity* [Foo99] presents a 2D visualization to show the structural and rhythmical characteristics of an audio recording.

By augmenting sheet music with harmonic information, Miller et al. [MBE19] proposed a design to help novices identify harmonic patterns they would usually not recognize. In a follow-up publication [FMK+20], they showed that, by adding rhythmic information to sheet music, novice users could recognize rhythmic structures that only experts used to see.

The above work indicates that there are applications for using visualization to provide musicians with additional meaningful information beyond sheet music, which we extend by visualizing bow motion.

2.2 Motion Tracking and Guidance Visualization

The fields of motion tracking and motion guidance visualization are closely related to our work, as we are looking to visualize motion data to provide means of analyzing and improving motions.

In a paper on motion guidance [YAM+20], Yu et al. explore and compare different perspectives, characteristics of motions, and visual encodings. They conclude that first person is the best-performing perspective for the domain, as the three different user studies that they conducted indicate.

With *Physio@Home* [TYB+15], Tang et al. explore solutions for using motion guidance to support physiotherapy patients when exercising at home. By displaying on-screen motion guides from different views, they aim to compensate for the lack of feedback and guidance from an expert when exercising at home.

Ware et al.'s work on visualizing the movements and behavior of whales [WAPW06] inspired us in our more analytical approach. By mapping whale movements to a 3D path and utilizing different textures and glyphs to encode different behaviors, they show long paths of whale behavior without animation. This is particularly interesting for us, as we intend to display entire motion sequences. Similar to their visualization of the whale roll behavior, we might use a ribbon instead of a line in the future to show the rotation of the cello bow. Also, an implementation of their space-time note, a note that can be placed at a specific position in space and time, might be interesting for us, as teachers could write notes for their students on interesting parts to work with at home.

With *AvatAR* [RBD+22], Reipschläger et al. present a system for immersive analysis of human motion data using augmented reality. They propose different visualizations to show human motion and posture with 3D trajectories, as well as exploring options to embed visualizations into the environment by highlighting footprints and objects that the user interacted with. If tracking of the cello is added to our design in the future, similar techniques might be used to show the positions at which the bow interacts with the cello.

We extend the above approaches by combining motion tracking with musical data to create new unique visualizations, utilizing different visual features of the trailing path to provide insight on musical data features.

2.3 VR/AR for Musical Tasks

A recent study [PLB+22] presents a system for musicians, that simulates different visual and acoustic environments in VR. The system was tested and evaluated with 19 musicians and was found to potentially help musicians in different ways, such as stimulating creativity and supporting them in their practice for rehearsal. By combining the results of this study with the approach of visualizing and analyzing musical data in VR, an interesting environment beneficial for musicians' creativity, practice, and self-analysis could be created. This will be further discussed in Chapter 7.

AR Hero [SSM+22] by Skreinig et al. utilizes AR technology to display guitar instructions directly on the instrument in real time. The software can present instructions generated from digital guitar tablature, as well as from actual captured performances.

While the above form an interesting foundation for using VR/AR while playing an instrument, none of them propose motion visualizations similar to what we present in this work.

2.4 Motion Tracking of Violin Bowing

The closest work to this thesis has been done in the domain of violin bowing, where the use of motion tracking to support musicians has been explored throughout multiple related works.

In 2003, Peiper et al. presented a system [PWG03] that utilizes motion tracking to capture violin bow movements, from which they can then extract motion features to classify bow strokes. As this paper focuses on the classification process, it was not relevant for our visualizations. However, for future work, similar classifications could be implemented to add information on top of our given visualizations.

With *MusicJacket* [LSBJ11], Van der Linden et al. present a wearable system to guide a violinist's bowing technique through vibrotactile feedback. It compares the musician's bowing to a target trajectory, providing feedback about the deviation as well as incorrect posture of the player.

Recently, D'Amato et al. presented a data-driven approach [DVO+20] for analyzing the motion data of violinists to classify the player's skill level. They aim to identify which motion features are the most suitable to distinguish between musicians of different skill levels. While the current results of that approach were not used for this thesis, future developments in this work could be interesting for exploring new data features to visualize.

Blanco, Tassani, and Ramirez present and evaluate a system [BTR21] to analyze sound quality and motion features of violinists. They propose real-time 2D visualizations of violin bow motions and sound quality measurements, to support musicians in practicing good sound generation.

Some of the data features proposed for violinists might be useful in our visualization for cellists as well, which we will evaluate further in the future. In addition or contrast to the related works in this section, our design uses VR to provide an immersive, direct spatial mapping of bow movements to a 3D line. Furthermore, we present different visualizations that are mainly built around utilizing the inherent visual features of 2D and 3D lines such as color and width, to represent different data features.

3 Design

In this chapter, we explain the requirements and design of our prototype. Inspired by Munzner's nested model [Mun09], we divided this chapter into use cases, tracking and data, and visualizations, as well as an additional section where we explain our user interfaces.

3.1 Use Cases

Following guidelines from design study methodology [SMM12], we worked in close cooperation with a domain expert. We aimed to create a design that can support cellists in improving their practice, by providing means of analyzing their playing.

For music schools, a teacher could record a music piece that the student would then play back, trying to follow the teacher's motion. When playing back the student's recording, the teacher could point out interesting aspects for the student to focus on, such as potential flaws, differences from the recommended motion, and even just sloppy motions.

Experts' motion data on a piece of music could be made available online. Self-taught musicians could then import and play back the data on their own device, attempt to follow the motion, and analyze differences and potential flaws in their own motion.

In general, cellists could record their motion, then watch it in VR or on desktop, visualizing and re-watching interesting details as often as necessary. To support the cellist in finding such details, we explore different metrics and visualizations to highlight different aspects of the motion.

In a more artistic approach, live visualizations of the cello bow motion could be shown on screens during cello concerts to additionally provide the viewer with a visual experience of the musicians' playing.

3.2 Tracking and Data

We can log different data features over time, some of which are explained in this section or summarized in Table 3.1 from our previously published paper [HKT+22]. Table 3.1 shows which features we can extract from the different sources and which visual channels can be used to encode the respective feature. For example, from audio, we can extract the loudness, which could be represented by the color, thickness, or texture of the line.

We track the cello bow's spatial position and orientation and log them over time. To be precise, we record the position of the bow's adjuster screw as a point in space, as it is an easily recognizable feature close to the musician's hand. Reasons and alternatives to this are further explained in Chapter 4.

Table 3.1: Data summary from our previous paper [HKT+22]

Feature	Source	Example visual encodings
time stamp	time	1D space, animation, label
3D position	motion	1D, 2D, or 3D space
3D orientation	motion	texture, material, shape
speed of movement	motion	color, thickness, texture
speed of rotation	motion	color, thickness, texture
pitch / played note	audio	color, label
loudness	audio	color, thickness, texture
expected note	sheet music	color, label
expected loudness	sheet music	color, thickness, texture

We also record audio, from which we can detect pitches (frequencies) using CREPE [KSLB18]. CREPE also provides us with a confidence value ranging from 0 to 1, representing the confidence that the given pitch is present at the given time sample.

From the collected data, we can compute additional useful data features, some of which are discussed in the following. The speed of movement, also referred to as velocity, is computed by measuring the distance between the position of two consecutive samples and dividing it by the time difference between the two samples. In analogy to the speed of movement, we can compute the speed of rotation by dividing the angle of rotation between two consecutive samples by the respective time difference. From the pitch, we can compute the closest corresponding MIDI note. Additionally, we implemented an example function to highlight potentially unclean bow strokes. For that, we compute the smallest angle between the line of movement and the pointing direction of the bow and divide the result by 90 degrees, resulting in a number between zero and one. A clean bow stroke without side movements would score best (zero) in this function, whereas movements perpendicular to the bow's long side score worst.

3.3 Color Schemes

We currently use three different color schemes to encode our different data types. An important consideration for all of the color schemes used is their contrast to the dark background that is used in our current design.

Blue-Yellow-Red (BuYlRd) from D3's chromatic scales[1] is used as a diverging color scheme for velocity, where slow is blue and fast is red. We divide the velocity by the user-selected maximum speed to map the velocity to an interval between 0 and 1. We also use BuYlRd to highlight potentially more interesting data in red. For example, with our stroke-angle function, alleged flaws are displayed in red, while smooth strokes are shown in blue. Color legends of the two examples can be seen in

[1] https://github.com/d3/d3-scale-chromatic

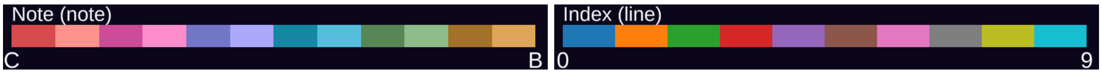
Figure 3.1. We chose BuYlRd mostly because it works relatively well with the most common color vision deficiencies, while also using common mappings: blue for good or slow and red for bad or fast.

(a) Color legend for velocity. **(b)** Color legend for stroke-angle metric.

Figure 3.1: Blue-Yellow-Red color legend for the two cases we currently use it for.

The current prototype uses the alternating dark and light color scheme proposed by Bunks et al. [BWSW22] to map musical notes, as shown in Figure 3.2a. This color scheme features twelve different colors, grouped in six pairs of adjacent colors. Two colors in a pair are similar in hue but differ in their saturation and lightness.

To distinguish between different lines when multiple recordings are shown, we have the option to show each line in its own color, using the *category10* scheme from D3. The *category10* scheme consists of ten categorical colors across the color spectrum, shown in Figure 3.2b. One requirement for this color map is that the different colors are clearly distinguishable, especially for the first few colors as we usually only display two lines.



(a) Color legend used for notes. **(b)** Color legend for line index used to distinguish different lines.

Figure 3.2: Color legends for notes and line index.

3.4 3D Visualization

Using 3D visualization in VR, we can show motion exactly as it was recorded, to give an accurate and intuitive representation of the player's movements. For our design, we draw a line in 3D space, simply following the path of the cello bow as it was tracked. As shown in Figure 3.3, this line will quickly clutter the small volume in 3D space within which the cello bow moves, so we consider different approaches to unclutter.

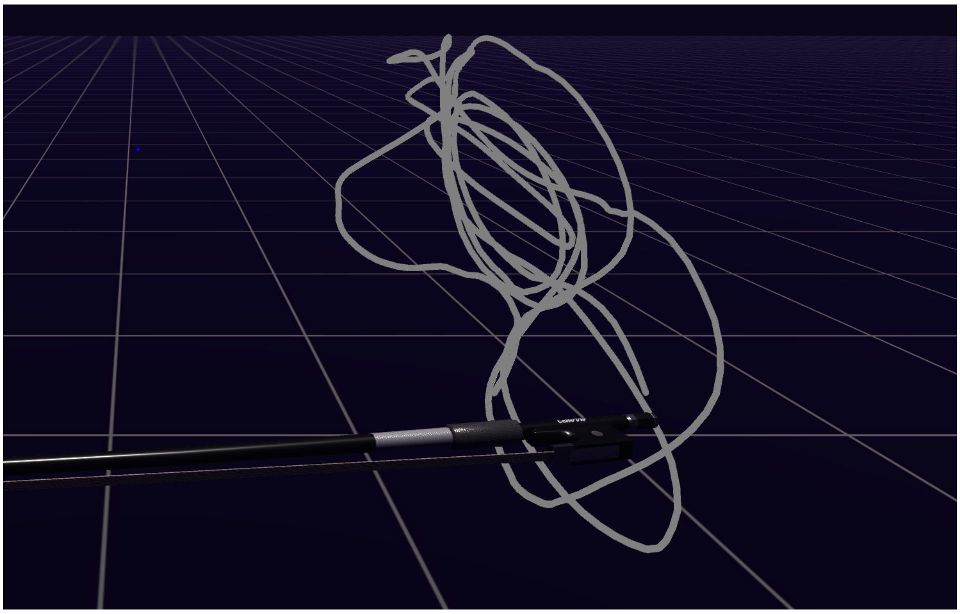
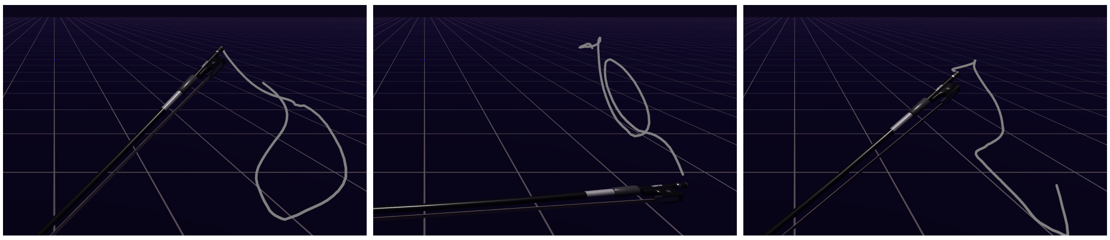


Figure 3.3: Plain 3D motion path the bow traveled within five seconds. Even for this relatively short amount of time it is already difficult to recognize distinct movements.

3.4.1 Filtered By Time

By displaying only points that lie within a given time interval, for example the most recent two seconds, we can animate a short, clear line segment that represents the current motion of the bow. Figure 3.4 contains a sequence of images, demonstrating how this can be used. As this approach completely hides most of the otherwise displayed data, it also helps with performance issues described in Chapter 6. The length and position of the displayed time interval can be easily adjusted by the user during runtime in different ways, granting the user control over which data is being displayed. However, if we are interested in viewing longer time intervals, another approach is necessary, such as *translating by time* as described below.

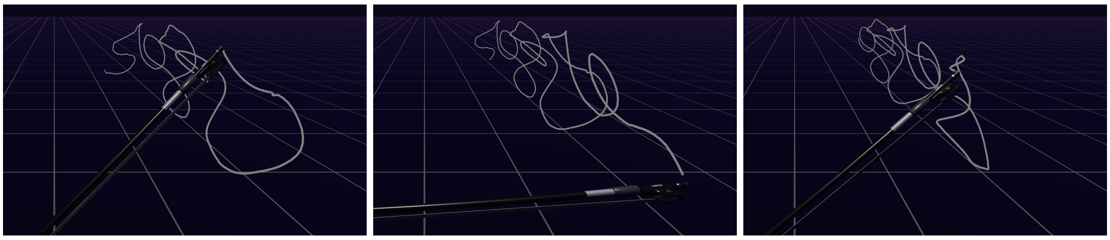


(a) Time: 0 to 1 second. **(b)** Time: 1 to 2 seconds. **(c)** Time: 2 to 3 seconds.

Figure 3.4: Plain 3D motion path that the bow traveled, but cropped to only show the last second. The three subfigures show three consecutive seconds of motion.

3.4.2 Translated By Time

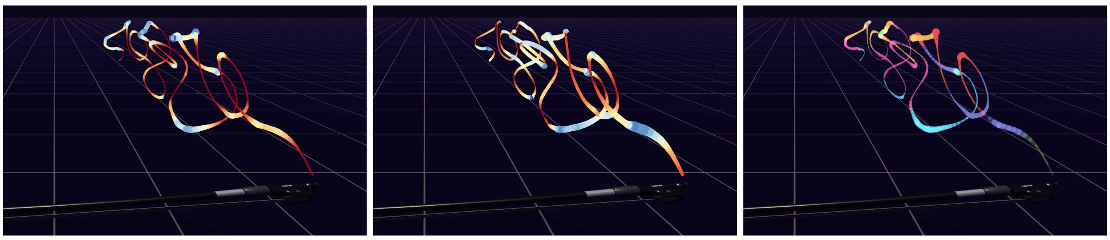
As most of the movement we are interested in happens in a shallow, almost two-dimensional volume, we can use the third dimension to display the time component. By transforming the created lines by a value based on their timestamp in the approximate direction the player is facing, we can make the line effectively “fly away” from the player. That way, the users can always see all of the displayed data in front of themselves. With this approach, points that are further back in time are displayed further away, as they are usually less interesting, whereas the most recent points are closest to the user, where new lines are still being created. The speed of the translation can be adjusted by the user during runtime, where setting the speed to zero practically disables this function. For the following, we usually apply this approach with a translation speed of $0.2\frac{m}{s}$, which is also what was used in Figure 3.5.



(a) Time: 0 to 1 second. **(b)** Time: 1 to 2 seconds. **(c)** Time: 2 to 3 seconds.

Figure 3.5: Plain 3D motion path that the bow traveled, showing a five-second interval, flying away 0.2 meters per second. The three subfigures show three consecutive seconds of motion.

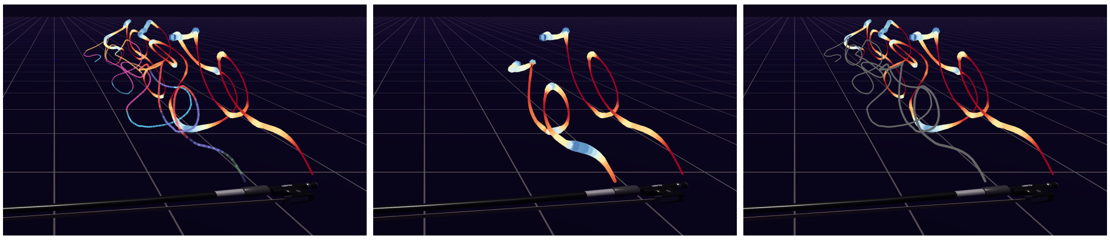
We can adjust the color, opacity, and thickness of the line according to metrics that we further discuss in Section 3.2. Some of the possible encodings are displayed in Figure 3.6. The encodings shown in Figure 3.6a will be used as the default for the 3D visualization in the following. For color legends, see Section 3.3. For line thickness, the line becomes thinner as the encoded number increases. For example, the maximum velocity would lead to a line width of 0.003 m, while a velocity of zero leads to a line width of 0.02 m.



(a) Encoding the velocity in the color and width of the 3D line. **(b)** Encoding the stroke-angle metric in the color and width of the 3D line. **(c)** Encoding the played note in the color of the 3D line, with opacity according to the confidence for the computed pitch at that time. Additionally, the width indicates the velocity.

Figure 3.6: 3D motion of the bow, encoding different metrics in line color, width, and opacity.

In addition to the main path described above, we draw a secondary path from a second point on the bow at a distance of a few centimeters from the primary point. This secondary line is meant to intuitively visualize the bow's orientation. It can also be used to visualize any of the same data as the main line (Figure 3.7b) but for the most part, we will leave it gray to keep the focus on the main line and reduce clutter.



(a) Encoding the played note in the color of the secondary line, with opacity according to the confidence for the computed pitch at that time.

(b) Encoding the stroke-angle metric in the color and width of the secondary line.

(c) Default main line, with gray secondary line.

Figure 3.7: 3D motion of the bow, using the secondary line to encode different metrics.

To give additional context on the currently displayed time interval, we add a label to the line at every fifth second, displaying the time stamp for the given point. This also helps to keep track of the current time when fast-forwarding, rewinding, or playing back at reduced speed. We further considered displaying sheet music information such as the current bar or beat. As this information might not be known and would require alignment when users do not (want to) follow exact timings, we left this feature for future work.

3.5 2D Visualization

To provide additional information that can be viewed simultaneously, we also present 2D visualizations showing different perspectives of the data, as well as some additional data that is difficult to display in 3D. Since the 3D visualization is distorted due to perspective, 2D charts can help by providing a view on parts of the data without distortion.

For 2D visualization, we can place multiple 2D line charts in 3D space. We designed 2D line charts that are synchronized with the 3D visualization and player inputs and can update in real time. The user can add as many line charts as necessary and move them to a good position, via the move-and-scale feature described in Section 3.7

Via a drop-down menu on each individual line chart, the user can select the color scheme and color-encoded data described in Section 3.2. Another drop-down menu allows selecting the type of line chart and the values that are presented, as described in the following. By using the two drop-downs, the user can create multiple customized line charts and arrange them in 3D space, as shown in Figure 3.8.

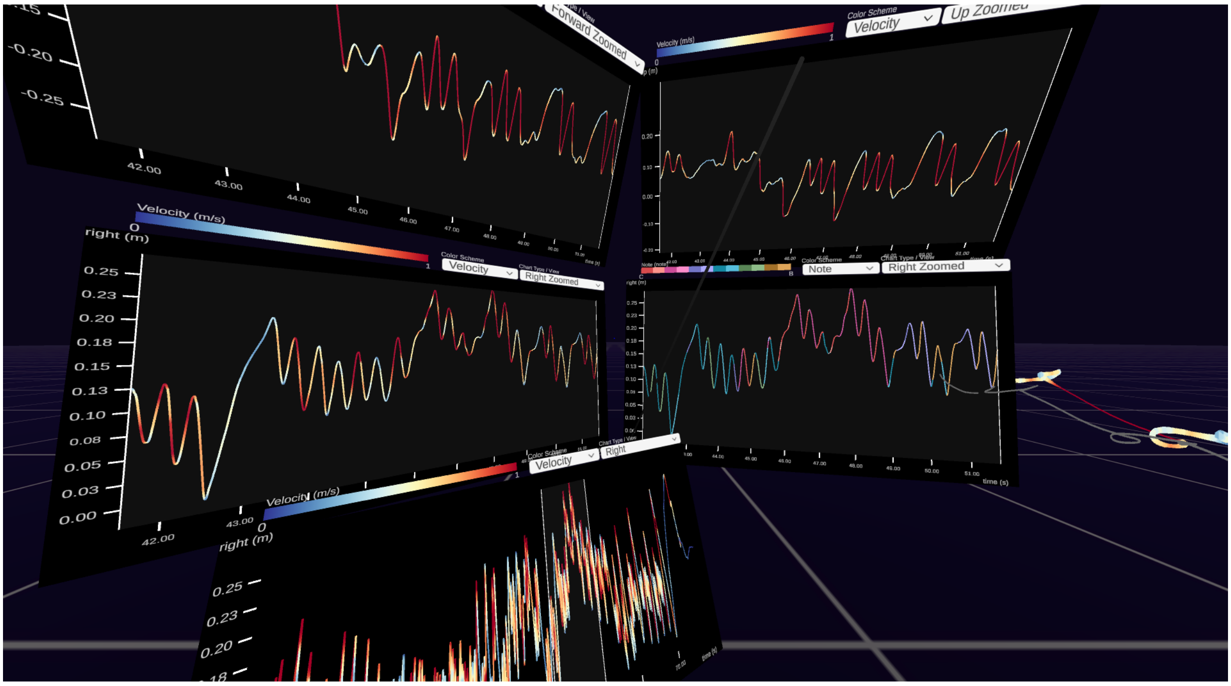


Figure 3.8: Assortment of multiple 2D line charts positioned in space to be viewed in VR. Line charts are individually customized using the two drop-down menus, to show different views and metrics.

3.5.1 Line Chart Types

Our line charts give different views on the same data, to follow the common concept of *overview first, zoom-and-filter, details on demand* [Shn03].

Full Overview In our *full overview* mode, we show the entire line from start to end, or while recording, from start to current time. This gives an unfocused overview packed with details of the movements during an entire music piece. Due to the length of a music piece, the 2D line in *full overview* is usually quite compressed on the x-axis, making it mostly suitable for a quick search for potentially interesting sections, which can then be selected and viewed as described below.

Zoomed View In the *zoomed view*, we only present the time interval of the line that is currently selected, potentially giving a far more detailed visualization of exactly what the user selected to see and/or what is currently played.

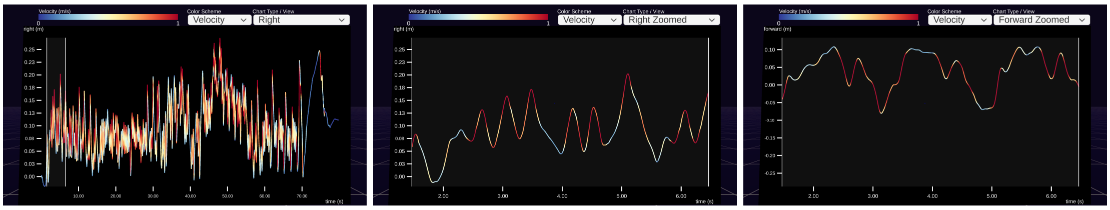
Filtered By Time In the *filtered by time* view, we only show points that are within the given time interval. This is used in conjunction with a mapping that shows a spatial axis on each of the axes, as that would usually clutter up the line chart.

3.5.2 Line Chart Axes Values

In the following, we explain some of the different value pairs that the axis of the line chart can represent.

Time vs. Position

In the *time vs. position* view, the chart displays time on the x-axis and one of the three spatial axes on the y-axis, to give an impression of left-right, up-down or forward-backward movements. This can be combined with *full overview* or *zoomed view* mode, to create the examples shown in Figure 3.9.



(a) Showing the left-right motion on the y-axis. *Full overview* mode.

(b) Showing the left-right motion on the y-axis. *Zoomed view* mode.

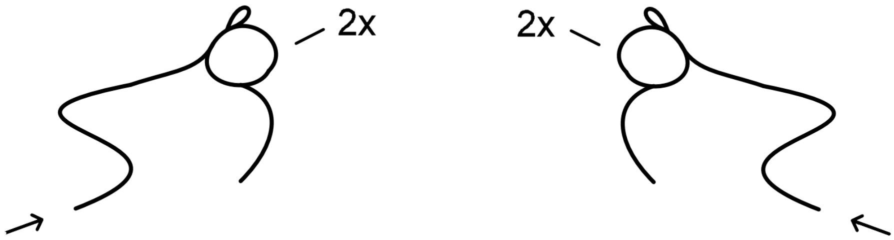
(c) Showing the backward-forward motion on the y-axis. *Zoomed view* mode.

Figure 3.9: 2D line charts, with time shown on the x-axis, and one of the three spatial directions (as seen from the user's perspective) on the y-axis. Color encodes velocity.

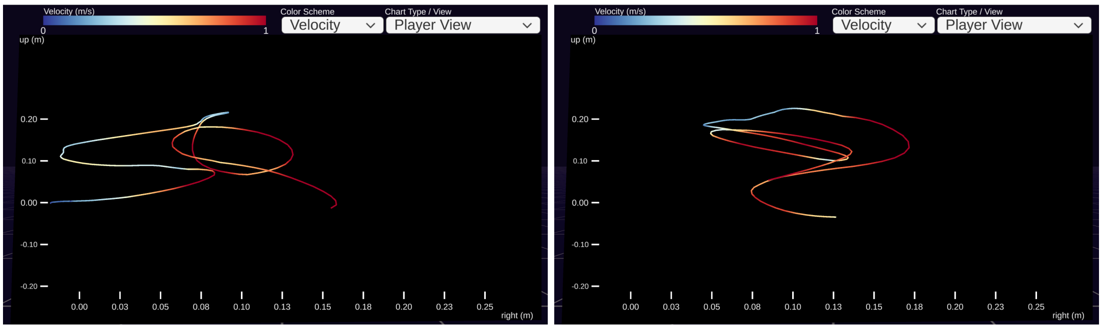
Position vs. Position

In *position vs. position* view, the chart shows one of the three spatial axes on each 2D axis. This mode should only be used for very short time intervals, as longer lines will quickly intersect and clutter up the 2D space. Thus, we only use this mode in conjunction with the *filtered by time* mode, which allows users to select the time interval they want to view.

The *position vs. position* view is mostly used to display the x-y-plane, which shows the up-down and left-right movements of the bow. This visualization is inspired by an illustration that the domain expert we worked with showed us. Such illustrations are scarcely used by some cellists for practice, augmenting sheet music to add the recommended bow motion. A comparison of the visualization of a recording and the corresponding illustration is shown in Figure 3.10. When created with this purpose in mind, these illustrations could also be used to generate and display reference motions for the user to follow.



(a) Illustration 1. Compare to Figure 3.10c. **(b)** Illustration 2. Compare to Figure 3.10d.

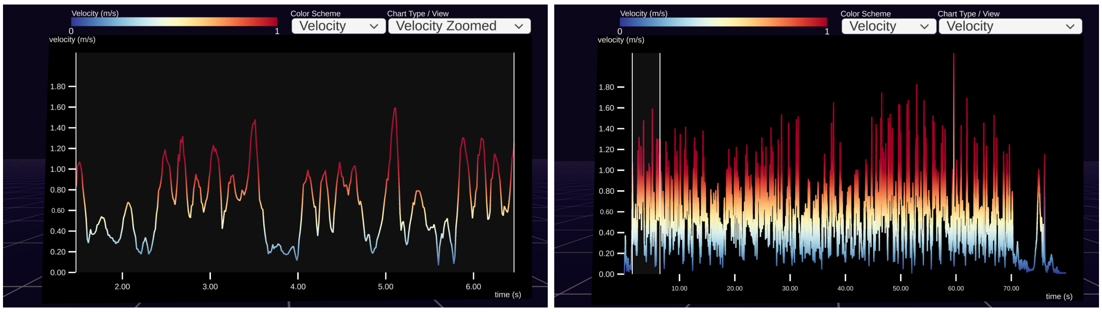


(c) First snippet from playing Bach Cello Suite No. 1 Prelude. *Filtered by time* mode. **(d)** Second snippet from playing Bach Cello Suite No. 1 Prelude. *Filtered by time* mode.

Figure 3.10: 2D illustrations previously created by our domain expert (top), which inspired us to the *position vs. position* view, along with their corresponding snippets (bottom) from playing Bach Cello Suite No. 1 Prelude.

Time vs. Metric

The 2D line charts can also be used to display any implemented metric on the y-axis against time on the x-axis. When combined with more advanced metrics, where flaws and other interesting details produce clear spikes in the data, this could be used to more easily identify and navigate to such details. An example of using this mode to view velocity over time is shown in Figure 3.11.



(a) *Zoomed view* mode. **(b)** *Full overview* mode.

Figure 3.11: 2D line charts, with time on the x-axis and velocity on the y-axis. Color encodes velocity.

3.5.3 Features

For our 2D visualization, the only visual feature we use besides the axes to encode our data is the color of each line segment. The line thickness as used in the 3D visualization was intentionally left out, as it makes the chart harder to read and does not convey the information well.

3.5.4 Time Control

To show what the currently selected time interval is and to allow the user to adjust that time interval, we add two vertical lines to each line chart that has time as its x-axis. These vertical lines represent the start and end time of the currently selected interval. That time interval is the same that controls the shown interval in the zoomed 2D line chart, as well as the part of the 3D line chart that is currently shown. By clicking and dragging the vertical lines the user can move them to select the desired time interval. This gives the user more control over what is displayed, in addition to the *time manipulation* described in Section 3.7.5.

3.6 Playback Two and Play Along

In addition to the default modes *record* and *playback*, we implemented a mode to play back two recordings simultaneously (*playback two*), as well as a mode to record alongside a playback of a previous recording (*play along*).

The *play along* mode is currently only a simple proof-of-concept, playing back the recorded track just like in *playback* mode, while also recording like in *record* mode. There are, however, more options that could be explored in the future, such as showing the playback line from a different perspective, to provide the musician with better motion guidance.

In our *playback two* mode, the two recordings are played next to each other at a given distance as shown in Figure 3.12, with both audio recordings being played from the respective bow. The user can choose to mute or hear either of the recordings to focus on the sound of only one of them. By offsetting either of the recordings, similar parts between the recordings can be synchronized. To do so in our current prototype, the user simply presses a button to make one of the recordings pause, effectively waiting for the other track to catch up. Automatically offsetting and aligning two tracks is planned for future work and discussed in Chapter 7, as it is especially relevant for recordings with different or even changing tempi.

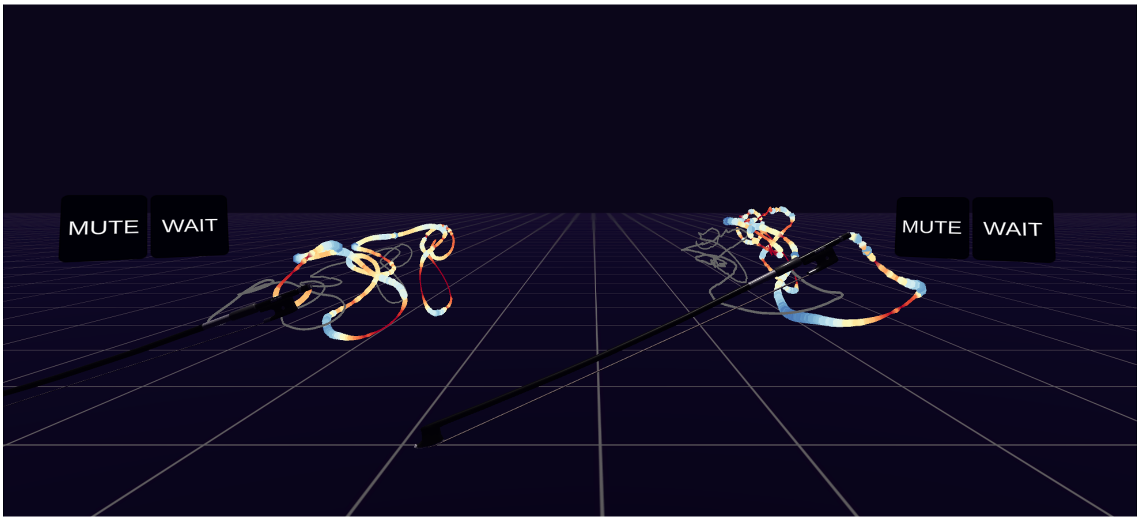
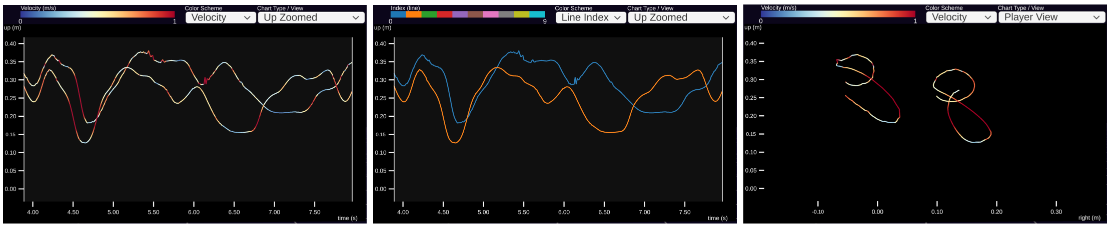


Figure 3.12: Two 3D lines played back next to each other. The *mute* buttons above the lines can be used to mute the audio of the respective line. The respective *wait* button can be used to pause the playback of one line, offsetting it in respect to the other line.

When using either of the two modes *playback two* and *play along*, the 2D line charts each display both lines in one chart, as shown in Figure 3.13. To make the two lines more easily distinguishable in 2D, the user can switch to a categorical color scheme, where each recording has its own color.



(a) Showing the up direction component of two recordings. Color encodes velocity.

(b) Showing the up direction component of two recordings. Each recording has its own color.

(c) Showing the player view for two recordings. Color encodes velocity.

Figure 3.13: 2D line charts with two recordings per chart.

3.7 User Interaction and Additional UI Elements

In this section, we explain our user interface design and present some additional UI elements placed in VR space.

3.7.1 Sheet Music

We created a sheet of music to float in VR space, such that the musician can play sheet music when recording while also viewing live visualizations, as displayed in Figure 3.14. Our current prototype allows the user to load one or multiple images containing sheet music. Pages can be turned back and forth by pressing the *page up*/*page down* buttons on the keyboard. This feature also supports page flipping pedals that are commercially available and used by some musicians for controlling sheet music apps on PC or tablets.

3.7.2 Floating Color Legend

Our floating color legend dynamically updates to show the legend corresponding to the color scheme and metric currently used by the 3D lines. Figure 3.14 shows the color legend arranged next to sheet music and the 3D line.

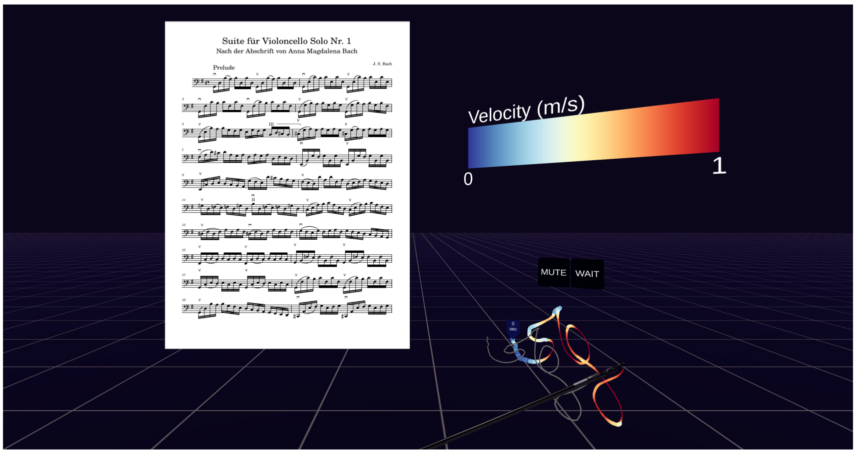
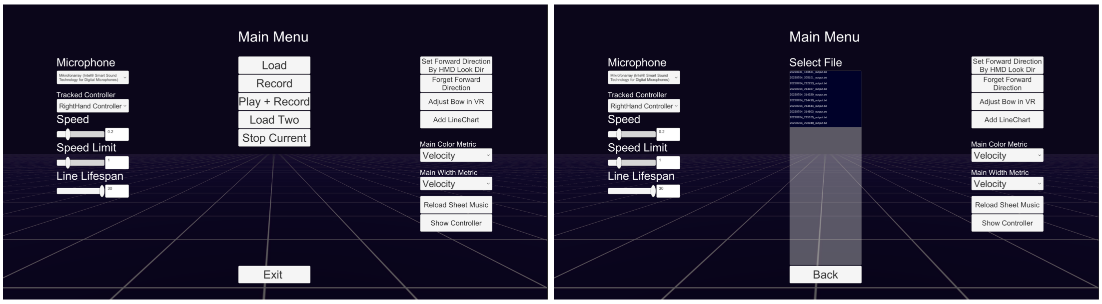


Figure 3.14: Color legend and sheet music shown in VR.

3.7.3 Main Menu

There are several ways in which we support user interaction with the software, including being able to choose between different modes, metrics, and visualizations, adjusting the environment according to personal preferences, and even just starting the recording and playback. We want to allow the user to take control while in virtual reality using the VR controllers, as well as leaving control to another person, such as a guide or a teacher, who can interact via a desktop user interface while the user focuses on playing the cello. We designed two relatively tightly coupled UI systems, such that most inputs that can be made in VR can also be made in the desktop view and vice versa. For our desktop UI, we created a simple overlay canvas with buttons, drop-down menus, and a scroll pane for the file selection, as can be seen in Figure 3.15.



(a) The desktop UI utilizes buttons, drop-down menus, and sliders for user input.

(b) When any of the playback modes is selected, a file selection scroll pane opens for the user to select an input file.

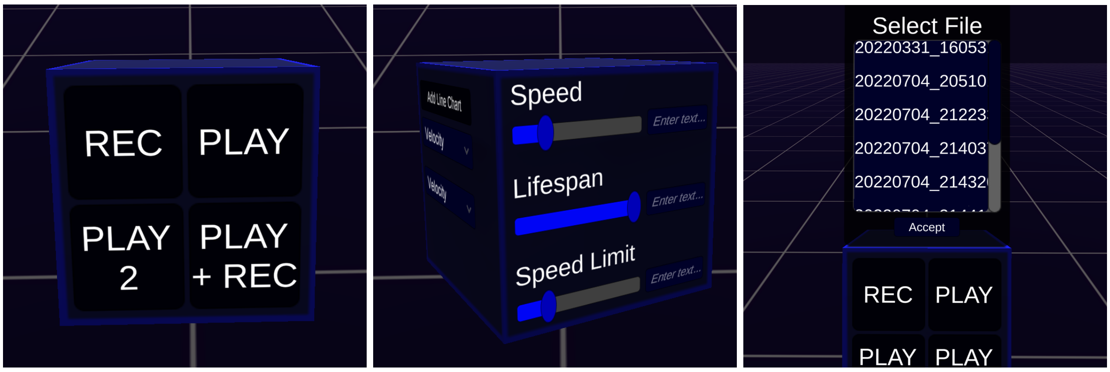
Figure 3.15: Images showing our desktop UI.

To allow user interaction while in VR, we created a UI cube that can be summoned to a position in front of the controller, as shown in Figure 3.16. Our initial approach for the VR UI was to simply place a world space canvas at a fixed position in the VR space. This, however, posed multiple problems and limitations: First, due to being in a fixed position, the canvas was often in an inconvenient angle and position to the user, making it difficult to interact with. As the amount of user-selected options increased, this issue became worse, as the canvas grew larger, creating more unfavorable angles. Second, the large canvas would obstruct the view when active during analysis, so we would have to hide it whenever any visualizations are viewed.

To address these problems, another approach could be to teleport the user to a separate UI room upon pressing a menu button. While this is a more viable option, it has the downside that the user cannot use the UI menu and view visualizations simultaneously.

Finally, we decided to create a VR user interface inspired by controller-attached interfaces such as the one used in the 3D painting app OpenBrush[2]. As one controller is usually taped to the bow for tracking, we can only utilize one controller for our UI, so instead of attaching the UI to the controller, we summon a UI cube to a position in front of the controller by pressing the menu button. It can then be interacted with using the same controller. The cube can be turned around by using the analog joystick to view the various UI elements attached to its different sides, which can then be interacted with via a ray cast from the controller.

[2] https://openbrush.app/



(a) The VR UI utilizes buttons, drop-down menus, and sliders for user input.

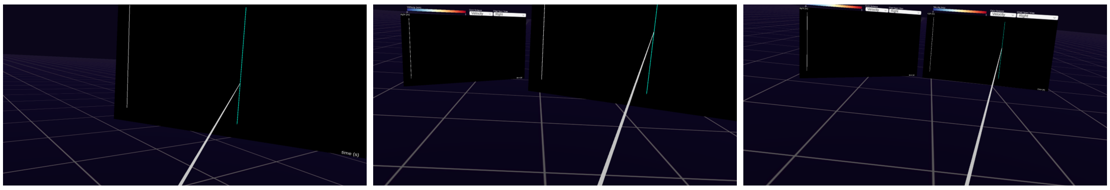
(b) By using the analog stick on the VR controller, the cube can be rotated to view its different sides.

(c) When any of the playback modes is selected, a file selection scroll pane opens for the user to select an input file.

Figure 3.16: Images showing our VR UI.

3.7.4 Moving and Scaling UI Components

We provide an option to move and resize each of the objects in the virtual space such as sheet music, floating color legends, and 2D line chart canvases, demonstrated in Figure 3.17. To perform these interactions, the user can simply press the grip button while hovering over an object to “grab” it, then move it around. Additionally, while grabbing, pulling the analog joystick of the controller to the left or right respectively decreases or increases the size of the object. Pulling the joystick forward or backward moves the object in the respective direction.



(a) A new 2D line chart was created in VR space.

(b) The line chart was grabbed and moved to be closer to a previously placed chart.

(c) The line chart was then scaled down, to match the size of the other chart.

Figure 3.17: Images series showing the user moving around UI elements and scaling them to fit the own requirements.

3.7.5 Time Manipulation

To view the playback animation in slow motion, fast forward, and rewind we provide a time control system to be used with the VR controller. By pressing and holding the grip button on the VR controller while not hovering over any grabbable object, the user activates the *time manipulation* mode. At the neutral position, the time scale is zero, thus the playback is paused. The user can then move the controller to the right to increase the speed at which time passes, or to the left to rewind time. This way, the time can be scaled arbitrarily depending on how far the hand is moved, where two centimeters correspond to ten percent of the normal playback speed. Only tapping the grip button for a short duration is a shortcut to pause or continue playback at normal speed.

4 Implementation

The implementation of our prototype uses the Unity[1] game engine for real-time rendering of 3D visualizations. During our design phase, we used Observable to view different possible 2D visualizations before implementing them in Unity, as it provides easy-to-use 2D visualizations.

Using OpenXR[2] and Unity's XR Interaction Toolkit[3], we implemented our prototype to be potentially compatible with most current VR devices for input and output. The XR Interaction Toolkit provides us with ray interactors, for the user to interact with UI elements, such as buttons, sliders, and drop-down menus. We also use its Locomotion System to give the user the ability to smoothly move around in the scene using the controllers. By extending parts of the XR Interaction Toolkit, we created the move-and-scale grab behavior described in Section 3.7.

For our 3D line, we use Unity's built-in line renderer. We chose this option because it is easy to use and produces good-looking results, but we are investigating other options, for reasons discussed in Chapter 6. We implemented our own dynamic 2D line charts based on Unity's meshes and canvases.

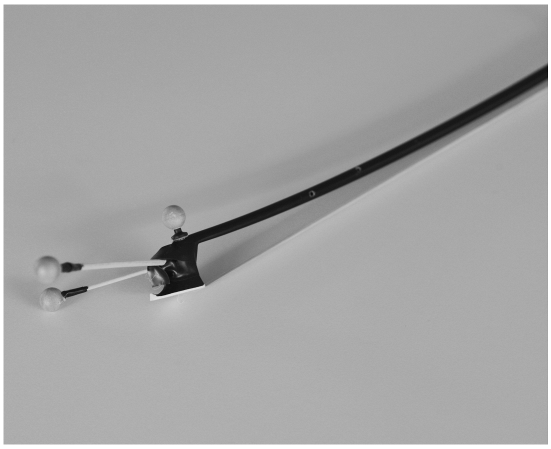
To record and play back audio, we use Unity's audio system, which also allows us to place the position of the audio source in 3D space to make the audio come from the position of the cello bow that plays it. We save audio files in the uncompressed wave (WAV)[4] format to preserve as much information as possible and because it is the required input format for our further processing as described below. To record audio from Unity's system to the WAV format, we use an altered version of a code snippet called SavWav[5]. We use the CREPE convolutional neural network [KSLB18] to retrieve the pitch of the audio recordings at discrete time stamps. From the pitch, we can compute the corresponding MIDI value as $midiValue = 12 * log2(frequency/440) + 69$. The zero-based index of the closest note ($C = 0, C\sharp = 1, D = 2, \ldots$) can then be obtained as $noteIndex = Round(midiValue)$ mod 12.

For input and output, we currently use the Oculus Rift S[6] head-mounted display (HMD) and its controllers. We taped one of the corresponding Oculus Touch 2 controllers to a cello bow as a cheap and easy option for bow tracking, as shown in Figure 4.1a. The orientation of the bow relative to the controller could then be adjusted by pressing a button in VR and overlaying the virtual model of the bow with the real bow. The HMD tracks the VR controller 80 times per second, with a "high level of accuracy and precision" [JNR21]. In our later development, we explored using the OptiTrack

[1] https://unity.com/
[2] https://www.khronos.org/openxr/
[3] https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/
[4] https://en.wikipedia.org/wiki/WAV
[5] https://gist.github.com/darktable/2317063
[6] https://www.oculus.com/rift-s/

motion capture system[7] to track the position and orientation of the bow with a camera frame rate of 120Hz. Using a cello bow with a pattern of six lightweight infrared markers as shown in Figure 4.1b, we were able to achieve sufficiently good tracking, without disrupting the musician's experience too much. Each of the solutions has its advantages: The controller is affordable and easy to set up and use, while the OptiTrack system adds less weight to the bow, making OptiTrack more convenient for the user.

(a) As a cheap and easy solution, we taped a VR controller to the bow. The controller is then tracked by the five built-in cameras of the HMD. Image from our previously published paper [HKT+22].



(b) Later, we used the OptiTrack system to capture the motion of the bow. The gray spheres are infrared-reflective markers that are tracked by eight cameras distributed around the room.

(c) Complete view of the bow with OptiTrack markers.

Figure 4.1: Photos of the cello bow prepared for the two different tracking approaches we use.

[7] https://optitrack.com/

When we refer to the bow's position as a point in 3D space, that is the position of the bow's adjuster screw. We chose to use the position of the screw over other alternatives, such as the bow's center (of mass), as the screw is an intuitively easy-to-recognize feature of the bow that is close to the cellist's right hand. The bow's center would be an intuitive measure of the bow's position but is rather far away from the right hand, thus representing a point in space that is not as clearly connected to the player's hand position.

5 Evaluation

This chapter explains the methods, setup, and results of the user study we conducted to evaluate the design.

5.1 Methods and Participants

To evaluate our idea, we conducted a user study, where we had experienced cellists use the prototype of our proposed design. As there are no clear performance measurements for musical exercise and self-analysis that could realistically be measured in the scope of this work, we chose a qualitative and explorative approach for our study design. We chose to perform pair analytics [AKGF11], where the designers guide the domain experts through the usage of the prototype while analyzing the data together. This allows us to capture reactions and feedback from domain experts, without them having to thoroughly understand our design and spend time on learning how to use it.

We reached out to other cellists who our main collaborator recommended to us. Unfortunately, many could not participate within the time limit of this work. Finally, we conducted the study with two participants (P1, P2), one of which is our main collaborator. This means we have one unbiased participant (P1), while the other already knew our design process (P2). The participants had 13 (P1) and 12 (P2) years of cello playing experience.

We conducted our study using the Oculus Rift S VR system for input and output. For motion tracking, we intended to use the OptiTrack system for all our participants. However, due to unexpected issues with the OptiTrack system, for participant P1 we had to fall back to our initial tracking setup, where we taped a VR controller to the cello bow. We had our prototype run on a VR-capable laptop connected to an OptiTrack system via WiFi. To collect the participants' feedback and reactions without the need of keeping notes during the study, we recorded the screen and audio via an external microphone with consent. This microphone was also used to record audio from the cello playing for the later analysis.

A few days before the study, each participant was provided with a slideshow containing simple explanations and images of the most important functions of the prototype, as well as a short summary of the procedure planned for the user study. The study itself consisted of two main parts, each of which we concluded with a semi-structured interview, taking a total of about 2 (P1) and 1.5 hours (P2). Participants were compensated for their effort with 10 Euro per hour.

As the studies were conducted in German, quotes from the participants were translated and potentially paraphrased for the following. In summary, the participants found our design interesting for analyzing their own play and expressed multiple ideas for future improvements and additional features.

5.2 Part 1: Live Visualization

The first part focused on the real-time visualizations displayed during playing the cello. For that, we asked each participant to play a part of Bach's *Cello Suite No. 1 Prelude*, a piece that they are currently working on, multiple times with the prepared bow while wearing the VR HMD. Before doing so, the participants could take a few minutes to get used to playing with the prepared bow as well as understand the general concept of the visualizations presented to them.

The first visualization we showed to each participant was the 3D line, with color and width both encoding velocity, *translated by time* with a speed of $0.2\frac{m}{s}$. We then proceeded by adding and explaining different variants of our visualizations, depending on what the participant was interested in.

P1 was fascinated when seeing the 3D line visualization coming out of the bow while playing for the first time: "it is totally fascinating to really see this", "I could basically see my interpretation".

The secondary line to show bow rotation turned out to be confusing for P1, who did "not entirely understand, what orientation" it was supposed to show. Being asked, P1 did not want the secondary line to be hidden, but found it "not immediately obvious what it should be good for".

When first seeing the 2D visualization, P2 found the *position vs. position* mode useful: "That is super helpful. I can really see whether the motions have edges or not.", "When I do jagged motions that are also not good for the sound, you can see that really well".

When viewing the stroke-angle metric, which shows straight strokes in blue and potentially inaccurate tilted strokes in red, P1 said it could be "very interesting for someone with a good bowing technique", but was surprised by the relatively low saturation of the positive color ("I would have expected the blue to be stronger"). P2 had similar experiences: They were surprised that something "looked very red, even though it was actually a very straight stroke", and mentioned the metric might not work correctly for the outer strings. P2 also proposed a potential improvement for the function, by ignoring motion in the upwards direction: "Most of it happens in this plane [parallel to the floor] anyways." P2 then asked to see movements in the mentioned 2D plane, as "it would be super interesting, as it would show how straight strokes are, too".

Surprisingly, after recording in *play along* mode, P1 did not like hearing their own recording while playing: "It rather confused me. I think it would be easier, if I would only see it, without hearing.", "Playing alongside a recording is always very difficult, as it is not as easy to hear and you cannot react to each other".

P1 found it interesting to play while wearing the HMD but had difficulties with not seeing the cello: "I do not know where exactly I was with the bow. There, I have to rely on my haptic feeling and I think I was rather not where I was supposed to be." In contrast to that, P2 had previously assumed that seeing the cello would not be that important. On the other hand, P1 also mentioned this might be a "very good exercise, to rely more on the feeling".

P1, who used the bow with the VR controller taped on, found it to be "not insanely comfortable, but working surprisingly well" and "exhausting, but less with the HMD than without".

P1 would consider using the system for self-analysis when available: "[...] to view my own technique. I also saw that there were small, unnecessary red strokes, movements that do not need to be there, unnecessary edges that I would have never noticed without seeing it."

5.3 Part 2: Analysis

During the second part, the participants analyzed the data recorded during the first part. Using the different visualizations and time controls, they viewed and reviewed the data in more detail.

When viewing the 2D visualizations in *zoomed view*, P1 found it “difficult to connect [the perceived] images and sound, partly due to it visually lagging”. While we previously recognized the limitation of reducing our virtual screens’ frame rate, this emphasizes the importance of further engineering to improve performance.

P1 found the color scheme that encodes notes inconvenient, which groups notes in adjacent pairs of two that are then colored similarly. The color scheme would group the notes E and F together, “instead of matching F and $F\sharp$”. P1 proposed an alternative in which “E would then just be an individual color”, similar to a piano keyboard. Furthermore, P1 found it difficult to use the note visualization for anything, did “not know what to do with it”, and thought it would at least take time to “get used to which sound is which color”. P2 found the note visualization “interesting, because it also brings in the left hand”, as sometimes there are multiple notes on one bow stroke. As an alternative for visualizing notes, P1 proposed to also show the deviation from the correct pitch, instead of the absolute note.

P1 was interested in different visualizations that were not yet implemented in the prototype: “Loudness [...] could also be interesting.” They would further find it useful to have “the cello [in VR], to see where exactly I stroke” and visualize “how much pressure I put on the string”.

P1 did not like the idea of comparing recordings in *playback two* mode for the sake of imitating the other’s playing: “I don’t know if that comparison between two recordings can really be applied. It is really not about playing exactly the same. [...] We want our own interpretation.” However, to analyze and find inspiration in others’ work, P1 found it to be potentially useful: “It might be useful for analyzing. [...] When the goal is just to compare, without judging. Not to see how to do it better but to see, how did this person do it and how did that person do it.” P2 said it would be interesting to record motion captures of the greatest professionals, “to see what they are actually doing”.

When asked whether they would use such a design when easily available in music school, P1 stated interest but mentioned the cost and setup of the tracking and VR devices as an obstacle.

P2 mentioned that our current design could hardly be used with more expensive bows, as it required permanent modifications to the bow: “There are bows for a hundred thousand Euros, you shouldn’t drill through those”, referring to holes in the bow that we previously used to attach the infrared markers. Our current design uses glue to attach the markers, but the statement about permanently modifying the bow still holds true.

When asked about the potential of using the design as a teacher, P2 said: “At the moment, I [mimic my student’s motion] to show: ‘Look, this is how you play. But it should look more like this.’ But it is something totally different to experience it yourself, through the HMD. And to really see the own play, not just the teacher mimicking it.”

P2 supported the idea of exploring different perspectives, which could be achieved through virtual mirrors, cameras, or already implemented player movement: “I could basically stand there and watch myself playing. That would be a great idea.”

6 Limitations and Discussion

Precise motion tracking of the cello bow comes with different compromises. Using affordable VR hardware, users can easily track their bow, but the tracked controller adds significant weight to the bow, which makes playing more exhausting and potentially distorts the recorded data. While OptiTrack delivers more precise motion data without adding as much weight to the bow, it is likely too expensive and requires a too complicated setup for most users, though it could be interesting for music schools and universities. Even the lightweight OptiTrack markers can be perceived as a compromise, as they are modifications to the bow that may harm the balance of the bow. The optimal solution with regard to usability and bow modification would be a purely camera-based computer vision tracking that uses feature detection to track the bow without markers. However, this would require a custom setup, which is inaccessible to most potential users, but probably cheaper than OptiTrack.

In our current implementation, we glued the infrared markers to the bow, permanently modifying the bow. This was criticized in our user study, as it essentially rules out the option to use the design with more expensive bows. This problem could be approached by designing lightweight clips that can be easily attached and removed.

A previously known limitation when using VR while playing an instrument is the absence of the instrument, as mentioned by Ppali et al. in similar work [PLB+22]. This was also noted by participants in our user study, along with the comment that perceiving the room around oneself can also be important while playing. Exploring solutions to those two problems is subject of future work, as mentioned in the following chapter.

Our current color encodings are not necessarily suitable for people with color vision deficiencies. While this did not pose a problem in our current user study, it is certainly an important limitation to work on in the future. Creating color maps that are distinguishable despite any color vision deficiency can be difficult for certain tasks. We discuss future approaches to this problem in Chapter 7. Furthermore, P1 criticized our note color scheme, which we can improve based on their feedback by assigning a separate color to the note E, such that F and $F\sharp$ are paired together instead of E and F.

Due to our initial approach of dedicating one of the two VR controllers to tracking only, we designed the user interface to be used with one controller. Using only one controller as an input device limits the options for user interaction. Additionally, the player's hands are usually occupied with the cello and bow, making it even more difficult to interact with UI.

Keeping up with the Oculus Rift S target frame rate of 80 frames per second posed quite a challenge, as we create 80 or more new data points per second that should be represented in real time by multiple line charts as well as three-dimensional lines. Since engineering for performance was not our focus and not in scope for the current prototype, we instead implemented some performance trade-offs to prevent the frame rate from dropping too low. These include updating the line charts every second instead of each frame, limiting the number of vertices displayed on line charts, and potentially skipping or merging samples of the three-dimensional line to reduce the number of line pieces. Long-term solutions for the performance problem could include finding a 3D line renderer that causes less computational load, as the Unity built-in one in our current implementation requires creating a new game object for each data point and line segment, potentially causing unnecessary overhead.

During the time frame for which we planned our user study, many of the cello students that were initially interested in participating in our study were busy and had no spare time to participate. This led us to have fewer participants than previously planned. Although our evaluation has already been partly done through our participatory design study approach [SMM12], a larger and longer user study, optimally a field study, would be required to evaluate real world usage.

7 Conclusion and Outlook

Learning and mastering the cello is a difficult and time-consuming task, for which visualizations can be a useful support. Since one of the key aspects of playing the cello is the bowing, we developed a design to assist cellists in analyzing and improving that motion. Our design aims to provide options to self-analyze their own motion and compare it to those of other cellists, to find interesting variations, flaws, and other aspects of their interpretation.

We use motion capture to record the position and orientation of a cello bow over time, along with other data features. To visualize the recorded data, we proposed displaying the trajectory of the bow over time as a line in 3D space, using different visual channels to encode the different data features. To support the 3D visualization with additional information and perspectives, we present several 2D charts that can be placed in 3D space and viewed in VR alongside the 3D line. In addition to the collected data features, we implemented a simple metric to highlight potentially interesting aspects in a recording, demonstrating how more complex metrics could be used to help the user find such aspects.

We evaluate our design in a qualitative pair analytics study with two experienced cellists. The study revealed some of the limitations of the design and provided us with ideas for future improvements, which we discuss below.

Outlook

During the design and evaluation of the prototype, we discussed new ideas regarding features that were not implemented as well as potential applications. In this section, we summarize some of those ideas, as well as other plans that were out of scope for this thesis.

The issues that occur due to not seeing the cello and the environment can be approached in different ways. By tracking the cello just like the bow, a model of it could be added to the virtual environment. This could also be used to display different visualizations on the cello itself, such as the interaction between cello and bow, similar to visualizations proposed by Reipschläger et al. [RBD+22] Adding ideas from *Keep the VRhythm going* [PLB+22], the user could be placed in different virtual environments with simulated acoustics, or even a virtual room that is modeled after the player's real environment. Alternatively, by using AR, the visualizations could simply be projected into the real environment, effectively addressing both problems.

To support unrestricted use of our design for people with a color vision deficiency, we intend to explore different color schemes and other visual channels to encode the data. As finding a solution that is ideal for everyone might be impossible, different modes for different color vision should be considered to utilize the respective perceived color spectrum.

As our stroke-angle metric shows, we can already use the different visual channels to highlight interesting aspects of a recording. Improving this metric and finding, exploring, and evaluating similar functions would make interesting future work, to make it easier for the musician to find potential flaws.

During our user study, one participant wished for a virtual mirror. Some cellists are already familiar with viewing their playing from different perspectives, as mirrors are commonly used among cellists for practice. We could build upon that by exploring how different perspectives can help when using our visualizations. For other instruments that are held more closely such as violins, a third position view from farther away could be better, as the visualization could otherwise be uncomfortably close or hard to read.

As mentioned in Chapter 6, the VR UI posed a difficult problem, which was not our focus for this prototype. Creating a better user experience will play an important role in the future of the project, as our visualizations are meant to be interacted with. To do so, exploring different options of interaction will be necessary. As the user's hands are usually occupied with the cello and bow, hands-free interaction or interaction using the cello bow might be interesting options.

Two recordings of the same piece of music can be different in their tempo, leading to asynchronous playback. In addition to the already implemented user-controlled time offset, future work could explore different techniques to automatize the synchronization of multiple recordings and between recordings and sheet music. One possible approach to do this would be to use dynamic time warping, which is used in audio recognition as well as motion capture to match two given sequences.

While participants said they would use our design to practice if it was available, they showed difficulties in immediately applying it to improve their play. This was to be expected for the short amount of time given for the user study but indicates the importance of exploring different methods to use the design in the future. In a larger user study with more participants and more time per participant, musicians could learn to use the design more thoroughly, exploring more methods of usage and documenting potential effects of that. Furthermore, through a longitudinal field study, real-world usage and acceptance of the design could be tested.

Bibliography

[AKGF11] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, B. Fisher. "Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics". In: *2011 44th Hawaii International Conf. System Sciences*. IEEE, 2011, pp. 1–10. DOI: 10.1109/HICSS.2011.339 (cit. on p. 37).

[BTR21] A. D. Blanco, S. Tassani, R. Ramirez. "Real-Time Sound and Motion Feedback for Violin Bow Technique Learning: A Controlled, Randomized Trial". In: *Frontiers in Psychology* 12 (2021). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.648479. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2021.648479 (cit. on pp. 11, 15).

[BWSW22] C. Bunks, T. Weyde, A. Slingsby, J. Wood. "Visualization of Tonal Harmony for Jazz Lead Sheets". In: *Eurovis 2022*. 2022. URL: https://openaccess.city.ac.uk/id/eprint/28140/ (cit. on p. 19).

[DVO+20] V. D'Amato, E. Volta, L. Oneto, G. Volpe, A. Camurri, D. Anguita. "Understanding Violin Players' Skill Level Based on Motion Capture: a Data-Driven Perspective". In: *Cognitive Computation* 12 (2020), pp. 1356–1369. DOI: 10.1007/s12559-020-09768-8 (cit. on pp. 11, 15).

[FMK+20] D. Fürst, M. Miller, D. A. Keim, A. Bonnici, H. Schäfer, M. El-Assady. "Augmenting Sheet Music with Rhythmic Fingerprints". In: *2020 IEEE 5th Workshop on Visualization for the Digital Humanities (VIS4DH)*. 2020, pp. 14–23. DOI: 10.1109/VIS4DH51463.2020.00007 (cit. on pp. 11, 13).

[Foo99] J. Foote. "Visualizing Music and Audio Using Self-Similarity". In: *Proc. Seventh ACM International Conf. Multimedia (Part 1)*. MULTIMEDIA '99. ACM, 1999, pp. 77–80. ISBN: 1581131518. DOI: 10.1145/319463.319472. URL: https://doi.org/10.1145/319463.319472 (cit. on pp. 11, 13).

[HKT+22] F. Heyen, Y. Kohler, S. Triebener, S. Rigling, M. Sedlmair. *Immersive Visual Analysis of Cello Bow Movements*. 2022. DOI: 10.48550/ARXIV.2203.13316. URL: https://arxiv.org/abs/2203.13316 (cit. on pp. 12, 17, 18, 34).

[JNR21] T. A. Jost, B. Nelson, J. Rylander. "Quantitative analysis of the Oculus Rift S in controlled movement". In: *Disability and Rehabilitation: Assistive Technology* 16.6 (2021). PMID: 31726896, pp. 632–636. DOI: 10.1080/17483107.2019.1688398. eprint: https://doi.org/10.1080/17483107.2019.1688398. URL: https://doi.org/10.1080/17483107.2019.1688398 (cit. on p. 33).

[KKM+20] R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, S. Jänicke. "A Survey on Visualizations for Musical Data". In: *Computer Graphics Forum* 39.6 (2020), pp. 82–110. DOI: https://doi.org/10.1111/cgf.13905. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13905. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13905 (cit. on p. 13).

[KSLB18] J. W. Kim, J. Salamon, P. Li, J. P. Bello. *CREPE: A Convolutional Representation for Pitch Estimation*. 2018. DOI: 10.48550/ARXIV.1802.06182. URL: https://arxiv.org/abs/1802.06182 (cit. on pp. 18, 33).

[LSBJ11] J. van der Linden, E. Schoonderwaldt, J. Bird, R. Johnson. "MusicJacket—Combining Motion Capture and Vibrotactile Feedback to Teach Violin Bowing". In: *Instrumentation and Measurement, IEEE Trans. on* 60 (2011), pp. 104–113. DOI: 10.1109/TIM.2010.2065770 (cit. on pp. 11, 15).

[MBE19] M. Miller, A. Bonnici, M. El-Assady. "Augmenting Music Sheets with Harmonic Fingerprints". In: *Proc. ACM Symposium on Document Engineering 2019*. DocEng '19. ACM, 2019. ISBN: 9781450368872. DOI: 10.1145/3342558.3345395. URL: https://doi.org/10.1145/3342558.3345395 (cit. on pp. 11, 13).

[Mun09] T. Munzner. "A Nested Model for Visualization Design and Validation". In: *IEEE Trans. Visualization and Computer Graphics (TVCG)* 15.6 (2009), pp. 921–928. ISSN: 1941-0506. DOI: 10.1109/TVCG.2009.111 (cit. on p. 17).

[PLB+22] S. Ppali, V. Lalioti, B. Branch, C. S. Ang, A. J. Thomas, B. S. Wohl, A. Covaci. "Keep the VRhythm Going: A Musician-Centred Study Investigating How Virtual Reality Can Support Creative Musical Practice". In: *Proc. 2022 CHI Conf. Human Factors in Computing Systems*. CHI '22. ACM, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3501922. URL: https://doi.org/10.1145/3491102.3501922 (cit. on pp. 14, 41, 43).

[PWG03] C. Peiper, D. Warden, G. Garnett. "An Interface for Real-time Classification of Articulations Produced by Violin Bowing". In: (2003), pp. 192–196 (cit. on pp. 11, 14).

[RBD+22] P. Reipschläger, F. Brudy, R. Dachselt, J. Matejka, G. Fitzmaurice, F. Anderson. "AvatAR: An Immersive Analysis Environment for Human Motion Data Combining Interactive 3D Avatars and Trajectories". In: *Proc. 2022 CHI Conf. Human Factors in Computing Systems*. CHI '22. ACM, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517676. URL: https://doi.org/10.1145/3491102.3517676 (cit. on pp. 14, 43).

[Shn03] B. Shneiderman. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations". In: *The Craft of Information Visualization*. Ed. by B. B. BEDERSON, B. SHNEIDERMAN. Interactive Technologies. Morgan Kaufmann, 2003, pp. 364–371. ISBN: 978-1-55860-915-0. DOI: https://doi.org/10.1016/B978-155860915-0/50046-9. URL: https://www.sciencedirect.com/science/article/pii/B9781558609150500469 (cit. on p. 23).

[SMM12] M. Sedlmair, M. Meyer, T. Munzner. "Design study methodology: Reflections from the trenches and the stacks". In: *IEEE Trans. Visualization and Computer Graphics (TVCG)* 18.12 (2012), pp. 2431–2440. ISSN: 2160-9306. DOI: 10.1109/TVCG.2012.213 (cit. on pp. 17, 42).

[SSM+22] L. R. Skreinig, A. Stanescu, S. Mori, F. Heyen, P. Mohr, M. Sedlmair, D. Schmalstieg, D. Kalkofen. "AR Hero: Generating Interactive Augmented Reality Guitar Tutorials". In: *2022 IEEE Conf. Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2022, pp. 395–401. DOI: 10.1109/VRW55335.2022.00086 (cit. on p. 14).

[TYB+15] R. Tang, X.-D. Yang, S. Bateman, J. Jorge, A. Tang. "Physio@Home: Exploring Visual Guidance and Feedback Techniques for Physiotherapy Exercises". In: *Proc. 33rd Annual ACM Conf. Human Factors in Computing Systems*. CHI '15. ACM, 2015, pp. 4123–4132. ISBN: 9781450331456. DOI: 10.1145/2702123.2702401. URL: https://doi.org/10.1145/2702123.2702401 (cit. on pp. 11, 13).

[WAPW06] C. Ware, R. Arsenault, M. Plumlee, D. Wiley. "Visualizing the underwater behavior of humpback whales". In: *IEEE Computer Graphics and Applications* 26.4 (2006), pp. 14–18. ISSN: 1558-1756. DOI: 10.1109/MCG.2006.93 (cit. on p. 14).

[YAM+20] X. Yu, K. Angerbauer, P. Mohr, D. Kalkofen, M. Sedlmair. "Perspective Matters: Design Implications for Motion Guidance in Mixed Reality". In: *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2020, pp. 577–587. DOI: 10.1109/ISMAR50242.2020.00085 (cit. on p. 13).

All links were last followed on July 16, 2022.

Kurzfassung

Um Cellistinnen und Cellisten dabei zu unterstützen, ihre eigene Interpretation eines Musikstücks zu analysieren und mit anderen zu vergleichen, präsentieren und erforschen wir in dieser Arbeit Möglichkeiten, musikbezogene Daten und mit Motion Capturing aufgezeichnete Bogenbewegungen in Virtual Reality zu visualisieren. Der vorgestellte Prototyp nutzt Verbraucher-VR-Technologie und optional das OptiTrack Motion Capture System um die Bewegungen beim Spielen aufzuzeichnen und zu visualisieren. Die entstandenen Visualisierungen können dabei entweder bereits während des Spielens in Echtzeit angesehen werden, oder danach erneut abgespielt werden. Wir präsentieren mitunter eine direkte Abbildung des Pfads des Bogens im dreidimensionalen Raum, welche durch verschiedene visuelle Kanäle zusätzliche Informationen darstellen kann. Desweiteren nutzt unser Design verschiedene 2D Liniendiagramme, die neben dem 3D Pfad im virtuellen Raum platziert werden können und mit diesem synchronisiert sind. Um das vorgeschlagene Design zu evaluieren wurde eine Nutzerstudie mit zwei erfahrenen Cellistinnen beziehungsweise Cellisten durchgeführt. Dabei wurden Anwendungsmöglichkeiten und Ideen für zukünftige Verbesserungen aufgezeigt.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature