

Visualization Research Center of the University of Stuttgart (VISUS)

Bachelorarbeit

# **Augmented Reality Drum Kit Feedback Visualization**

Jonas Lammert

**Course of Study:** Informatik  
**Examiner:** Prof. Dr. Michael Sedlmair  
**Supervisor:** Frank Heyen, M.Sc.

**Commenced:** February 6, 2021  
**Completed:** August 6, 2021



## **Abstract**

This thesis explores several visualizations that aim to assist aspiring drummers with learning the drums using projector-based augmented reality technology. It aims to provide feedback on their training sessions when they do not have a professional teacher available. This feedback is projected directly onto the instrument, so the user does not have to transfer the information from e.g. a screen to their drums and can save cognitive resources. We give feedback in real-time and retrospectively. For the real-time feedback, we have to consider a visualization that keeps the information lightweight, to not divert too much attention away from playing the instrument, so we use simple encodings like color or a speedometer. For retrospective feedback, we use more complex visualizations like a pie chart. Our approach can give feedback on timing and dynamics of the played pattern. We evaluated our visualizations with three case studies, one that focuses on comparing the different visualizations, one that evaluates their efficiency in identifying errors, and one that investigates the influence of the visualization regarding errors made over time. We conclude that the visualizations help the user identify their errors, however, we did not have enough data to truly verify that they reduce errors in one's performance over time, or if one visualization has a better performance than the others.





# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Related Work</b>	<b>11</b>
2.1	Mixed Reality . . . . .	11
2.2	Mixed Reality in Training and Education . . . . .	11
2.3	Mixed Reality in Music Education . . . . .	12
2.4	Glanceable Visualization . . . . .	15
<b>3</b>	<b>Design</b>	<b>17</b>
3.1	Hardware . . . . .	17
3.2	User Tasks . . . . .	18
3.3	Workflow . . . . .	18
3.4	Data . . . . .	18
3.5	Tracking the Drums . . . . .	19
3.6	Types of Feedback . . . . .	19
3.7	Types of Visualization . . . . .	22
<b>4</b>	<b>Implementation</b>	<b>27</b>
4.1	Programming . . . . .	27
4.2	Manual Configuration . . . . .	27
4.3	Drum Trainer . . . . .	28
<b>5</b>	<b>Evaluation</b>	<b>35</b>
5.1	Case Study 1: Evaluating the Learning Effect of Visualizations Over Time . . . . .	35
5.2	Case Study 2:Evaluating the Helpfulness in Identifying Errors . . . . .	37
5.3	Case Study 3: Comparing the Visualizations . . . . .	39
5.4	Limitations and Discussion . . . . .	39
<b>6</b>	<b>Conclusion and Outlook</b>	<b>45</b>
	<b>Bibliography</b>	<b>47</b>



# List of Figures

1.1	Visualization techniques. . . . .	9
3.1	Projector with webcam attached on top. . . . .	17
3.2	E-drum kit with white paper stuck onto the drum pads to improve the visibility of the projections. . . . .	18
3.3	Attempt to use Hough transformation. . . . .	19
3.4	Manual tracking. . . . .	20
3.5	Concept sketch for our timing feedback. . . . .	21
3.6	Concept sketch for our dynamics feedback. . . . .	22
3.7	Testing the maximum amount of information that is possible to display on a pad. . . . .	22
3.8	Concept sketches for speedometer visualization, signal visualization, and pie charts. . . . .	24
4.1	Configuration help function. . . . .	28
4.2	Setting up the drum trainer. . . . .	28
4.3	Comparison between dark and white background when projecting onto the drums. . . . .	29
4.4	Red-white-blue color scheme. . . . .	30
4.5	Dynamics feedback using color visualization. . . . .	30
4.6	Red-yellow-blue color scheme. . . . .	30
4.7	Timing feedback using color visualization. . . . .	31
4.8	Timing feedback using speedometer visualization. . . . .	32
4.9	Dynamics feedback using speedometer visualization. . . . .	32
4.10	Implementation of pie charts. . . . .	33
4.11	Implementation of our signal visualization. . . . .	33
5.1	The groove used for evaluating the signal visualization. . . . .	36
5.2	The groove used for evaluating the color visualization. . . . .	36
5.3	The groove used for evaluating the the speedometer visualization. . . . .	36
5.4	Errors made per repetition in session 1. . . . .	36
5.5	Errors made per repetition in session 2. . . . .	37
5.6	Errors made per repetition in session 3. . . . .	37
5.7	Dynamics errors made per repetition in the special dynamics session. . . . .	38
5.8	The groove used for comparing the visualizations. . . . .	39
5.9	Results of the case study shown as box plots per session. . . . .	40
5.10	Results of the case study shown as box plots for all sessions combined. . . . .	40



# 1 Introduction

The drums are a difficult instrument to master, since a set consists of several drums and cymbals most of the time, and playing it requires both hands and feet to move, often at different speeds to hit the drums asynchronously. This kind of multitasking requires a lot of cognitive resources for the inexperienced player so their goal should be to achieve automaticity, which means they can play a pattern without thinking about it too much. This saves a lot of cognitive resources which can be used to synchronize with the other instruments of a band or following a conductor.

Achieving automaticity requires disciplined and frequent training sessions over a long time. For making the most of these sessions, a professional teacher who supervises the learning and gives guidance and feedback is required. However, having a professional teacher is a luxury not everyone can afford, so many aspiring drummers turn to the internet to find resources for self-teaching the drums. Here, they can find many text and video tutorials and scores to play for free, which they can use to get some of the guidance a professional teacher would give. However, these resources can not give direct feedback on a player's performance, which is troublesome for self-teaching learners, because they often have problems identifying their errors, since it is hard for the inexperienced ear to identify them in timing or dynamics while playing. This can be a problem even for learners who have access to a professional trainer because most of the time they will meet them only for a



**Figure 1.1:** In this thesis, we use different visualization techniques to help the user learn playing the drum.

single training session once a week. However, if the learner has serious ambitions to become good at playing the drums, one training session a week is not enough. For every other session they are not able to rely on the feedback of their teacher, which reduces their potential learning effect.

In recent years, several educative programs have been released that aim to teach the user an instrument and give feedback while playing. Apps like Yousician<sup>1</sup> or Rocksmith<sup>2</sup> let their user play a song and can give live feedback on their timing. Their success proves that there is a demand for such music education programs, however, they can visualize their feedback only on the screen of the PC or mobile device they are running on. This is not optimal, because the user has to transfer the information on-screen onto their instruments. This transfer function requires cognitive resources of the user. As we established, especially on an instrument that requires as much multitasking as the drums, these cognitive resources are sparse, therefore it would be ideal if we could save the resources the user would have to invest, by visualizing the feedback directly onto the instrument.

In this thesis, we introduce visualizations that can be used in such a program. Our goal is to give helpful feedback to an aspiring self-teaching drummer that increases the learning effects of sessions where no teacher is available. We project this feedback directly onto the drums using augmented reality technology. We want to use a visualization that is simple and lightweight to not demand too much attention from the user, so they can concentrate on playing the instrument. This feedback should help the learner to self-evaluate their performance and identify their errors correctly. To give the user patterns to play, we provide MIDI files (Musical Instrument Digital Interface) [Moo86], a communication protocol to connect electronic musical instruments and computers. We also save their performances as MIDI so we can analyze them later for giving feedback.

---

<sup>1</sup><https://yousician.com/>

<sup>2</sup><https://www.ubisoft.com/de-de/game/rocksmith/plus>

## 2 Related Work

In this chapter, we present related work done in researching mixed reality usage in training and education and discuss what we can learn from them and where our work differs. We distinguish between work used in music education and work used in general education. Before that, we briefly discuss work that try to define what mixed reality is. Finally, we discuss work that researched glanceable visualization and why this field is related to our work.

### 2.1 Mixed Reality

Before we can discuss related work in the field of mixed reality, we first have to clarify what we mean when we use this term. This is not as simple as one might think, finding an adequate definition of MR is a highly discussed topic. An early attempt at finding this definition was made by Milgram et al. [MK94]. They introduced a continuum with two extremes, on one side a fully real environment and on the other side a fully virtual environment. This is called the reality-virtuality continuum, everything in between, excluding the extremes, is mixed reality. Speicher et al. [SHN] criticized this definition for not including VR and because it is mostly focused on visual displays. They argued, after interviewing several professors and surveying many papers on that topic, that MR can be many things depending on context, and suggest that MR can be seen for example as a combination of augmented- and virtual reality, as a more capable version of AR, or even as a synonym for AR. For our causes, understanding MR according to the reality-virtuality continuum but including VR is sufficient.

### 2.2 Mixed Reality in Training and Education

Using augmented or virtual reality in training and education is not a new concept. Early research into this field dates back to 1995 when Psootka and Joseph [Pso95] discussed the possibilities of using VR environments in training. They already saw a lot of potential in these technologies and remarked how VR environments were able to create a compelling sense of immersion and could be used in situations where setting up real-life simulations would be expensive, like training for NASA flight controllers.

Nowadays, MR applications are used in the education of several areas, like in Medicine. In 2019, De Luca et al. [DCPL19] introduced a multi-procedural virtual reality simulator for orthopaedic training, which is an interactive simulation based on virtual reality to target open orthopaedic surgery training. The evaluation showed that it achieved an acceptable level of realism. In 2020, Shao et al. [SYQ+20] introduced a virtual reality technology for teaching neurosurgery of skull base tumors, evaluation showed that the score of the group who used the VR teaching exceeded those of

the group who did not. Heinrich et al. [HJLH19] use projector-based AR technology like we do to improve instrument navigation in needle-based interventions. They tried different visualizations for navigation support for the needle insertion task that were directly projected on a training dummy.

Some projects use MR to teach the user specific movements for sport and exercise. Chan et al. [CLTK10] created a virtual reality dance training system using motion capture technology, where a virtual representation of the user that moved via motion capture and a virtual trainer who demonstrates the motions the user wants to learn is projected on a wall. The program can give feedback on which body parts the user had most problems with. They conducted a user study that suggests that the system does help to keep students motivated and improve their skills. In 2015, Tang et al. [TYB+15] build a prototype, that guides people through pre-recorded physiotherapy exercises using real-time visual guides and multi-camera views. They visualized the feedback directly on the human body, which is kind of comparable to us visualizing feedback directly on the instrument. Their user study suggested that this kind of feedback helped the participants to be more accurate in their motions.

MR training is also often used in fields where the situation that needs to be trained is too dangerous or impossible to create a real-life training environment of. For example, Iguchi et al. [IMS16] introduced a system in 2016 that used virtual reality for disaster education, it was a digital game-based evacuation drill system using Using Augmented Reality and a smartphone-based head-mounted display that should encourage adults to think about proper evacuation strategies. A comparable system was introduced in 2020 by Andersen et al. [AGSH20], it used a VR environment to train workers of underground mines how to react in cases of emergency.

Using MR in training and education also proves useful in more common fields, as several studies show that it can increase the user's motivation, engagement, and learning. For example, Abrar et al. [AIH+19] could observe these effects on preschool children in Bangladesh. In 2019, Khan et al. [KJO19] observed similar increases in motivation on undergraduate health science students when they researched the impact of AR applications on learning motivation.

A common result in the above work is that the Mixed Reality systems they use have a positive influence on the user's learning. Therefore, we are confident that we will also be able to increase the learning effect of drum training sessions using MR technologies. Most of the work also observed a positive influence on a user's motivation, with the last two especially investigating this phenomenon. As some learners who try to teach themselves the drums lose motivation and give up, we hope that our approach will also have a positive effect on their motivation and prevent that.

### 2.3 Mixed Reality in Music Education

Using MR in music education is a field where a lot of work has already been done for many different instruments. We first present some work that have the intention to teach instruments other than the drums, and afterward, we take a look at the work that has been done in teaching the drums.



### 2.3.1 Mixed Reality Systems To Teach Instruments Other Than the Drum

In 2006, Motowaka et al. [MS06] developed a support system for guitar playing using augmented reality. It shows the learner how to correctly hold the strings by overlaying a virtual hand model and lines onto a picture of the guitar taken by the webcam. This feedback is only displayed onto a computer display, so the user has to transfer this information onto their instrument, which requires cognitive resources. We save these by projecting the feedback directly onto the drum. Unlike the drums, the guitar is not a static instrument, it will be moved around while being played, so tracking it just with its natural features is hard. The authors use a physical marker to help with keeping track of the guitar even if it is moved around.

Markey et al. [MWM+21] also designed a system for guitar learning, this time they visualized the feedback directly on the instrument by using colored LEDs on a 3D printed fretboard. This is comparable to what we do with our color visualization, however, our drum kit does not have to be modified directly, we only need to use a projector.

Huang et al. [HZY+11] introduced an augmented reality based system to teach the piano, this is interesting for us because they also used projector-based AR. They projected virtual hands onto the piano depending on which notes should be played next. They were able to implement a system that could track the keyboard of the piano naturally and automatically without virtual markers. This was possible because the piano is a static instrument so the tracking does not have to be adjusted while playing in contrast to the for example the guitar. Their program identified the keyboard by the structure of the black and white keys. We also planned to track our drums automatically, but the drums do not have such a convenient structure so the means of tracking it would have been a bit more complicated, we discuss this in detail later.

Johson et al. [JDT19] researched the effectiveness of mixed reality music instrument learning with the theremin. To do this they conducted a study that compared traditional music learning with two virtual learning environments, an augmented reality environment like ours, and a fully immersive virtual reality environment. Their goal was to increase the understanding of how skills learned with training in the MR environment would transfer to real-world performance, which is also an interesting question for our project. To do this they tried measuring the learning transfer, which is in this context the degree of which learning in one MR environment affects performance in the real world.

In the study participants had significantly smaller performance errors when using the immersive VR environment than in the other two environments, however, training there lead to smaller learning transfer than training in the real-world training environment. Johson et al. assumed that this could be the case because the participants focused their attention more on the visuals in the immersive VR environment than on the audio feedback of the instrument. This went so far that participants' real-world performance got worse after using the immersive VR environment for training. The researchers suspect that this happened due to the lack of spacial awareness between the virtual and the real world. Despite all that, the participants preferred the immersive environment and found it more engaging, probably because it was more novel and exciting. Regarding learning transfer, the augmented reality environment performed better though, which is reassuring for us because we plan to use an AR environment too.

### 2.3.2 Mixed Reality Systems for Drum Training

There is only little work that aims to create a MR system specifically to teach the drums.

Holland et al. [HBDH10] introduced a drum learning system based on haptic technology. They state that simply hearing examples is not sufficient to deal with technical and written aspects of music connected with rhythm. Therefore, an additional system would be necessary to give feedback and optimize training sessions. They choose to create a system that uses computer-controlled bracelets on both ankles and wrists that can vibrate to give feedback in form of haptic pulses that would guide the playing.

They evaluated their program with a user study, and got generally positive comments. However, a few participants noted that the haptic volume could sometimes be too low, for example on fast sequences pulses could blur into one another. The most obvious difference between this project and ours is that it uses a different sense than ours, they used haptics, we use vision.

Another work for teaching the drums, that uses visual feedback with projector-based technology as we do, was introduced by Yamabe et al. [YAKN11] in 2011. In contrast to our project, their focus was to design a program to keep intrinsic motivation by offering playful features such as a game-like interface and a variety of tasks with different difficulty levels, in contrast our approach is not as game-like.

They stated that for playing the drums, the player has to move both feet and hands at different times to interact with several interactive points (every part of the drum counts as one), which makes it more complicated to play than instruments with only a single interaction point like a guitar. Therefore, one should consider keeping the visual information lightweight to save cognitive resources. The authors spread their feedback onto several displays, with a projector they projected ambient feedback onto a wall that the user could perceive through their peripheral vision, another projector could project direct feedback directly on the pads. On a nearby computer monitor, they could either display the music score or a gaming UI.

On the gaming UI on the monitor, notes of a pattern get visualized as vertically down scrolling lines, just like in a rhythm game. The user has to hit the drum when this line reaches a line at the bottom of the screen that corresponds to a drum, they are identified with pictures of that drum. Feedback messages such as good or miss are given for every played note, at the end a score is calculated based on the player's performance. The ambient feedback is a huge cycle projected on the wall and is used to give timing feedback, for example when a drum is played too late it will turn red. Directly on the drum, they visualize what note should be played next, they constructed several ways to visualize with which timing the next note should be played, for example one needle for every note that rotated clockwise on the drum and indicates that the drum should be hit when it hits twelve.

To evaluate their program, they conducted a user study. Most of the participants preferred the gaming UI on the PC monitor, they could not recognize information well when they were distributed onto several display devices. Because of this, we decided to only project our feedback directly onto the drum and abstain from using any additional displays. They also identified that it is problematic to visualize rhythm onto the small surface of the drum pads, especially if the song is fast it gets too difficult to distinguish individual notes when for example using the clock visualization described above. For this reason, we do not try to visualize which notes to play next, we assume that the user is already aware of which drum they have to hit, so we can concentrate on giving feedback on the notes that are played.

## 2.4 Glanceable Visualization

We expect users of our approach to only glance at the visualization on the drums for a short time while continuing to play, therefore it is important to keep that in mind when designing the types of visualization we use. For this, we need to consider glanceability, which is the ability to understand information on a screen or a surface quickly at a single glance. Glanceable visualization is a field with a lot of existing research. Most of the research we found was done on smartwatches, their small and usually round surfaces are comparable to the surfaces of our drums, the user is also expected to perceive the information on them in only short glances.

Blaschek et al. [BBB+18a; BBB+18b] compared several types of visualization for their glanceability on smartwatches. They compared three types of charts: bar, donut, and radial charts. A user study was conducted, where the donut charts performed best, the bar charts were close second, and the radial charts performed worse. The authors assumed that pie charts should perform just as well as donut charts for they are relatively similar. Their difference is that donut charts have a “hole“ in their center while pie charts are filled.

Later, the study above was replicated by the same team [BI21], this time they used a laptop computer screen to conduct the study, but kept the same resolution for the visualization as they did on the smartwatch. The goal was to determine if the study would yield the same results when conducted on the laptop setup as it would on a smartwatch setup. They concluded that the results of the study are not dependent on the setup it was executed on, therefore we can deduce that the results of the study are also relevant for the visualization projected onto our drum’s surface.



## 3 Design

In this chapter, we explain how our project was designed, we discuss which kinds of visualization we used and why, and also describe setup and use of the program.

### 3.1 Hardware

For projecting the visualization, we use the elephas Portable LED Projector of the model GC333 (Figure 3.1). It has a maximum display resolution of 1080p (1920×1080 pixels) and a physical resolution of 800×480 pixels. We mounted it at a height of 1.85 meters and pointed it down on the drums to which it has a distance of 1.6 meters. The stand we mounted it on is a Walimax pr FT-8051 lamp tripod with a maximum height of 2.60 meters. On top of the projector, we placed a webcam, the Microsoft LifeCam HD-3000, that has a resolution of 720p (1280×720 pixels), which will later help us track the location of the drums. The skidproof stand of the webcam is already effective to keep it in place, but to be safe we additionally used tape to attach the webcam onto the projector. The digital drum kit we use is the Millenium MD-90 mobile drum (Figure 3.2). It is a small MIDI drum kit that can easily be connected to a PC via USB and communicate with it via MIDI. The drum pads of the kit are completely black, so in order to give our projections better visibility, we attached white paper onto the drums. While testing, we found that the paper does not influence the detection of drum stick hits in any significant way.



**Figure 3.1:** Projector with webcam attached on top.



**Figure 3.2:** E-drum kit with white paper stuck onto the drum pads to improve the visibility of the projections.

## 3.2 User Tasks

Our approach is designed for users who want to get specific feedback on their training sessions. It gives feedback on dynamics, the volume the instrument has to be played (e.g. piano or forte), so the user can learn to hit the drums with the right amount of force to produce the required loudness. The program also gives feedback on timing, so the user can learn how to play a pattern with the correct rhythm.

## 3.3 Workflow

The user chooses a music piece they want to learn and imports it. When learning a new pattern, it is a good idea to play it slower at the start, which we support. They can request the e-drum to play the pattern by itself for reference, while it is playing, guidance on how the pattern is played is projected on the drums. Afterward, they can play the pattern and the program gives live feedback on the dynamics or timing, depending on which mode is chosen. After playing, the user can request retrospective feedback on their timing.

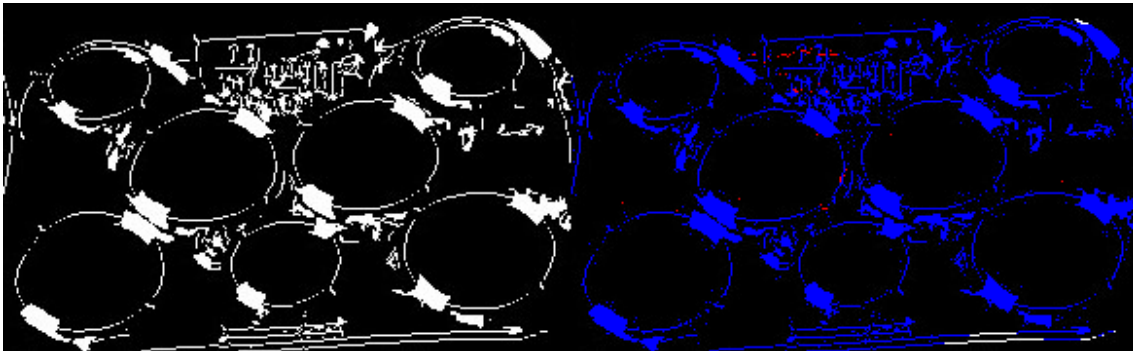
## 3.4 Data

For recording and handling the music data, we use MIDI (Musical Instrument Digital Interface), a communications protocol to connect electronic musical instruments and computers. MIDI files save several pieces of information about every played note of a music piece. The ones we use are name and octave, whose combination corresponds to a specific part of the drum kit in drum MIDI files, timestamp, to identify when it was played, and velocity, which translates to how much force a note is played with. The user can import the pattern they want to learn as a MIDI file that contains at least these pieces of information. Our program also records all this data for every note played, to allow comparisons between them and the imported notes.

### 3.5 Tracking the Drums

To project the visualization on the correct position in the real world the program needs coordinates that correspond to the position of the drums, for this we need some way to track the drum kit.

We originally planned to use Hough transformation [Hou62] to realize this. This would theoretically allow us to identify all ellipses and their centers in a given picture automatically, so by using it on a picture of the drum set taken by our webcam it should have been possible to find the position of all the drum pads. We used an implementation of the Hough transformation [pgu14] as a base for our implementation, when getting a completely black and white picture as input it should have given a picture as output where all the ellipses are colored blue and their centers are marked with a red dot. Unfortunately, the implementation did not yield the desired results as seen in Figure 3.3, almost all white areas were colored blue, and the red dots were placed at seemingly random locations. After trying and failing multiple times to fix our implementation, we ultimately decided that we would leave Hough transformation for future work and implement an easier way to track the drums.



**Figure 3.3:** Input (left) and output (right) of our failed attempt to implement Hough transformation. The red dots were supposed to correspond to the center of the found ellipses, but they ended up in seemingly random locations.

As an alternative, we decided to implement a system for the user to set the position of the drums manually. They are provided with the picture from the webcam and can click on the center of the drum they want to set, and then adjust the height and width of the projected ellipse with a slider. A picture of the full system can be seen in Figure 3.4.

### 3.6 Types of Feedback

We consider two types of feedback, timing feedback and dynamics feedback. These are two important aspects of playing the drum where self-evaluation can prove difficult, especially for the untrained ear, since it is hard to perceive subtle errors in these areas. We are confident that our visual feedback can help with this.



**Figure 3.4:** User interface to manually track drum pads. Users can go back and forth to a list of all drums and set their locations by clicking on the center of the corresponding drum on the webcam picture. They can then set the desired size of the ellipses via the sliders on the bottom.

### 3.6.1 Timing Feedback

When playing in a band, the drums are the instrument that sets and maintains the rhythm and tempo of a song, mainly by playing a repeated phrase called a groove. Therefore, it is important to be able to play a song with consistent timing and rhythm to not confuse their fellow band members. To do so, the player has to master all the grooves the song consists of, which requires frequent training over a long time. Usually, an aspiring drummer would use a metronome to practice their timing. However, the metronome is unable to give direct feedback on how precise the beat was hit. The learner has to rely exclusively on their hearing to evaluate their timing which can be difficult, especially for the untrained ear. Therefore, feedback is required to identify the timing errors the player might make. We aim to provide that feedback visually using projections directly on the instrument.

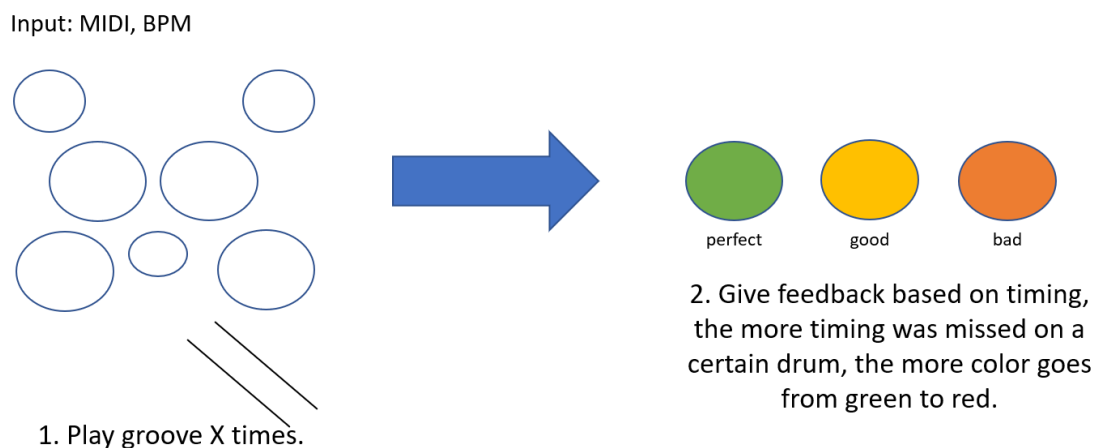
Here two types of timing feedback are possible, active timing feedback, where the feedback is given while the pattern is played, and retrospective timing feedback, where it is given after the pattern was played. Both have benefits and drawbacks:

When giving feedback in real-time while playing the user can immediately see where they made the mistake and can adapt accordingly, on the other hand, the feedback will, especially in faster patterns, only be visible for a very short period of time, so the user may miss it. Also one could question the



habitational effect of such a system. If the user only adapts to the live feedback they may grow too dependent on the program and feel lost without it. Therefore they would not be able to play the pattern in e.g. a concert scenario where the program is not available.

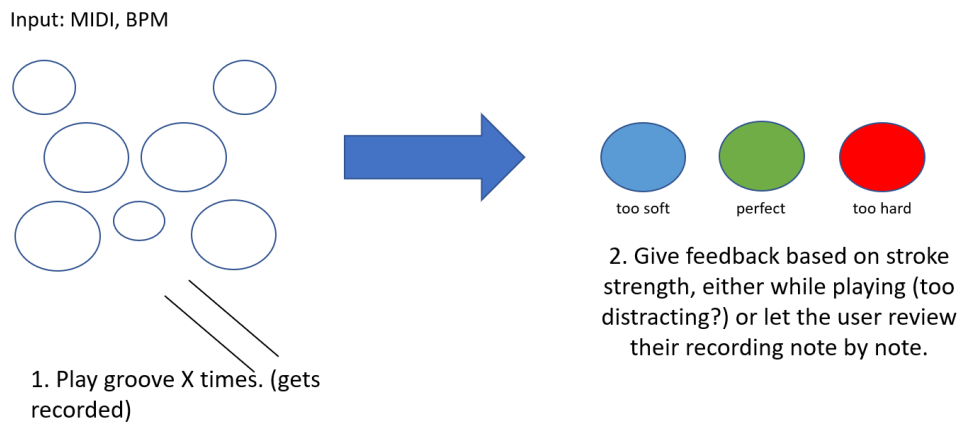
Retrospective feedback will stay on the drum as long as the user likes, therefore more detailed and less lightweight types of visualization are possible, for example pie charts (Section 3.7). We can also project feedback onto all drums at once, for example to show how many mistakes were made on each drum, which would be too distracting for live feedback. If feedback is only given after playing, the user also has no opportunity to grow too dependent on the program. A weak point of retrospective feedback is that it is hard to visualize where exactly in the pattern mistakes were made. One could implement a system where the user can go through the pattern note by note after they played it and find their errors this way. We decided against this, because we feel that most users would find this a bit too laborious and tedious. Also showing a second view, for example a piano roll on the wall, would be too distracting as the evaluation Yamabe et al. [YAKN11] did on their project suggested. Our first sketch of how this could look like using colors can be seen in Figure 3.5. Of course, both active and retrospective feedback can be used in combination. While playing one can provide active timing feedback and afterward one can analyze the recorded data and provide additional retrospective feedback.



**Figure 3.5:** Concept sketch for our timing feedback.

### 3.6.2 Dynamics Feedback

Many drum beginners make the mistake of not paying attention to dynamics. As a result, their playing can sound stiff and not very dynamic. This is especially hard to self-evaluate, because it is difficult to reproduce the exact volume of a note you just hear. The visual feedback of our approach can provide is easier to adapt to. Figure 3.6 shows a first sketch of how this could look like using different colors.



**Figure 3.6:** Concept sketch for our dynamics feedback.

### 3.7 Types of Visualization

When picking the type of visualization we want to use, there are several things we need to keep in mind. For once, the drums are small and round, so only a limited amount of information can fit on their surface (Figure 3.7) so a lot of classic types of data visualization, such as bar charts or line charts, are not suitable.



**Figure 3.7:** Test how much text could fit on the biggest drum. The text on the bottom left drum is the smallest possible font size that is still readable.

When planning to provide active feedback, it is also important to keep the visual information lightweight, so perceiving them does not take too much cognitive effort and therefore does not distract the user from playing. Considering all this, we explore several methods of visualization below.

### 3.7.1 Color Visualization

The first visualization method we consider is visualizing our feedback by projecting simple colors onto the drums. The color fills the whole surface of the drum, so the visibility is good as long as we do not choose colors that are too pale. Colors are also lightweight information, they do not need much cognitive capacity to be perceived and are easily identified at a single glance.

One important thing to consider when deciding to do color visualization is picking an appropriate color scheme. We first considered a traffic light color scheme, red to yellow to green, because it is intuitive, in general green is associated with correct and red with wrong. However, if we want to build an accessible program, we run into problems here. People with red-green color blindness will have trouble differentiating the colors. This condition is not uncommon (it concerns 8% of the male population [Kal05]), so it is something we should consider. An easy fix for this problem is to change green to blue, this reduces the intuitiveness of the color scheme, so it is important to provide a legend to the user. This color scheme is sufficient if we just want to convey sequential information to the user, for example, if the drum was hit right or wrong and to which extent.

If we want to convey a bit more detailed information, for example, if the drum was hit too strong or too hard, this color scheme is not appropriate, we need two extremes with the ideal in the middle. We first considered using the “turbo” color scheme by Anton Mikhailov [Mik19] (blue to green to red), however, we run into the same accessibility problems as the traffic light color scheme here, so we decided to use a blue-white-red diverging scheme instead. We use white to indicate that the drum was hit with the correct intensity. In practice, this looks as if nothing is projected onto the drum, so the user can assume that nothing is wrong. Red indicates that the drum was hit too hard because red is seen as an intense color and because it is often used to indicate danger, red seems like an appropriate choice to use for indicating that something was hit too hard. Blue indicates that the drum was hit too soft, we use this color because it is associated with calmness so the association with playing too quiet is somewhat intuitive. Nonetheless, this scheme is not as intuitive as the traffic light scheme, so a color legend is necessary.

Our color visualization has some weaknesses we need to consider. For once, the human eye can only differentiate so many colors, so the amount of data that can be conveyed is limited. It is also not easy to see subtle differences, for example, if two shades of yellow are projected on two different drums it is not always easy to tell on which of them the user had the better result. We also need to consider that a color scheme we find intuitive can be perceived differently in another culture due to cultural differences in the meaning of colors, for example, some cultures do not differentiate between green and blue [KM13]. We therefore continued to explore other encodings, as described below.

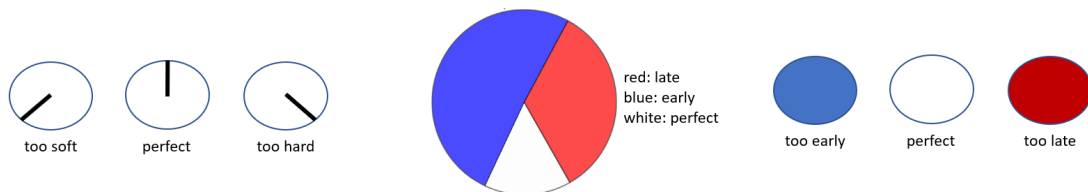
### 3.7.2 Speedometer Visualization

The second visualization method is projecting a needle on the drum. The information is conveyed by how much the needle deflects. We call this “speedometer visualization” because it is reminiscent of a speedometer in a car. This is especially intuitive for dynamics feedback, because the harder the drum is hit, the more the needle deflects, just like a speedometer would deflect more the faster one drives. We are aware though, that it is not necessarily intuitive that the optimum is in the middle. When using this method for timing feedback, it is also not necessarily intuitive that the more the

needle on a drum deflects to the right, the more the notes were played correctly on the respective drum, so color would arguably be the better encoding for that use case. As before, we should always provide a legend to the user, to compensate for the lack of intuitiveness. This visualization method has an advantage over the color visualization as it is more precisely readable. Remembering the example we had earlier, where the values on two drums were very close, we can now more easily tell which of them performed better than we could with color. However, because the needle has to be narrow (its readability would decrease if it was too wide), its glanceability seems to be a bit lower than the glanceability of the colors which cover the whole drum.

### 3.7.3 Pie Chart Visualization

The next visualization method uses pie charts, these are usually not recommended for data visualization because humans are bad at reading angles [Dat18] and it is hard to figure out which part of the pie is the biggest if they are close in size. However, they lend themselves quite well for our means, because they can make ideal use of the round shape of our drums. We can convey more information with pie charts than with the former two visualization methods, however, they are not lightweight information anymore and therefore not suitable for real-time feedback. That is why we only use them for our retrospective timing feedback. This way, the user can get a good overview of all the notes they played. We can “cut” all the played notes into different “slices”, for example, a slice for all notes that were played too early, one for all that were played too late, and one for all that were played perfectly.



**Figure 3.8:** Concept sketches for speedometer visualization for dynamics feedback (left), pie charts (middle), and signal visualization for timing feedback (right). In the speedometer visualization, the harder the drum is hit, the more the needle deflects. A pie chart is projected on every drum after a pattern was played, shows how many notes were played too late/early or perfect. In the signal visualization, feedback is given when the user plays a note wrong. The opacity of the colors depends on how big the error was.

### 3.7.4 Signal Visualization

The last visualization method only projects feedback when a mistake is made, we call it “signal visualization”. Our Signal visualization is similar to colors, in that it also uses colors for visualization. However, we only need two colors, and we only project them in certain situations. When notes are played correctly, nothing is projected on the drum and the user can therefore play without distractions. Once the user makes a mistake, the program gives a signal to them in form of a color. Depending on the degree of the mistake, the opacity of this color increases. It is possible to project different colors for different kinds of mistakes, for example for timing feedback we project red when

a note was played too late and blue when it was played too early. This method of visualization is very lightweight, the user realizes immediately when a mistake is made and they do not have to comprehend the color first.

For an early concept of this and the prior two visualizations, see Figure 3.8



## 4 Implementation

In this chapter, we explain in detail how we implemented the visualizations described in the design chapter.

### 4.1 Programming

Our program is implemented in Javascript, it runs in the browser therefore we need a backend server to save and manipulate configuration data, so the user does not have to set up the program every time they restart it. For implementing the frontend, we use [React](#), because it is well suited for creating interactive user interfaces. We use [Express](#) to implement the backend. In addition, we are using several packages. We use [axios](#) for communication between the front- and backend, we also use [webMIDI](#) to communicate with the e-drum, we utilize [d3-scale-chromatic](#) to get color schemes for several of the visualizations, the [musicvis-lib](#) package makes it easier to work with MIDI files and the data recorded from the e-drum, [material-ui](#) provides us with components like sliders to create a more user-friendly UI.

### 4.2 Manual Configuration

As described in the Chapter 3, we decided to leave the tracking of the drums to the user. For this purpose, we provide a system where the picture of the webcam is given to the user on a clickable canvas. The picture is zoomed in to match the surface on which the projector can project as much as possible.

Because the webcam can only give us a picture with a resolution of 640×480 pixels, we have to map the coordinates to a position on a canvas with the size of the resolution of the projector we are using. Also, because the drums are not on a perfectly flat surface (some drums are higher or lower than the others) we have to modify the y coordinates further. We figured out how we need to modify them through try and error. These modified coordinates are saved as x and y coordinates server-side via an axios write call.

Because the projector does not project onto a flat surface, perfect circles in the program will not appear round in the projection, so we decided that using elliptical forms would fit better. Therefore, the user can set the x and the y radii of the ellipses separately.

Once the user clicks the button to start the configuration, a separate tab opens where the user can request to display the ellipses they last set. We recommend to set the projector as a separate screen and set the tab this configuration program is running in to full screen there, so the user can set the position of an ellipse and immediately see where it will be projected. The test program draws the

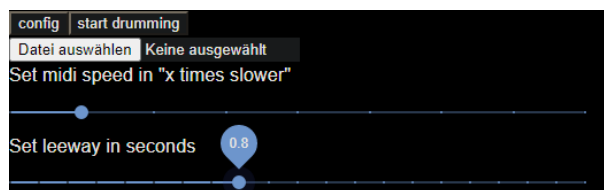
ellipses on an SVG with a small black dot in the middle, so the user can identify where exactly they clicked, see Figure 4.1. To only show the ellipse that is currently in the progress of being set, the configuration program sets the opacity of all the other ellipses to zero, after the user presses the button “end config”, the opacity of all ellipses is set to one, so the user can check the result after all drums are set.



**Figure 4.1:** Ellipses are projected onto the drums right as their positions are set, to help users more easily find the center. Black dots show the coordinates the user clicked.

### 4.3 Drum Trainer

When the user finished setting up the configuration and is ready to start their practice, they can click the “start drumming” button and are brought to the main component of our program, the drum trainer. First, as seen in Figure 4.2, the user can import a MIDI file containing the piece they want to learn. They can decrease the speed of the MIDI if they want to practice the pattern more slowly. Furthermore, we allow the user to set a leeway, so the program is not as harsh with evaluating the correct timing. We decided to let the user set this themselves because beginners probably do want a higher leeway than advanced players.



**Figure 4.2:** The user can import the piece they want to learn as a MIDI file and configure its speed and the timing leeway.

To parse the imported MIDI we use musicvis-lib, which saves all the information into a NoteArray. A music piece comes with many useful functions, we use, for example, the scaleTime function to scale the start time of every note according to the MIDI speed the user gave as input to slow down the piece.



After the user imported their MIDI file, the program requests the configuration from the server and draws the ellipses onto an SVG. We first used a white background and projected only the outlines of the ellipses as long as the drums were not played. Later we decided to invert the color of the background to black and project the ellipses in white as long as no drums are played. This leads to a better visibility of the colors and helped the player to focus more on the projections on the drum because not everything was illuminated anymore. As a bonus, it also looks more aesthetically pleasing. For a comparison, see Figure 4.3. Next, we discuss what happens when the user plays the drums, which depends on the visualization the user chooses.



**Figure 4.3:** Comparison between dark and white background when projecting onto the drums.

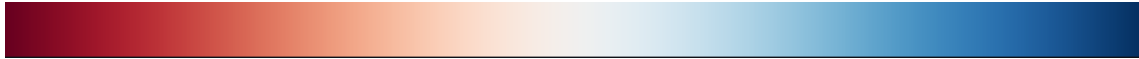
### 4.3.1 Color Visualization

In this mode, we give active feedback on dynamics and retrospective feedback on timing. Every played note is recorded in a NoteArray, so we can compare them to the notes that were imported from a MIDI file.

#### Dynamics Feedback

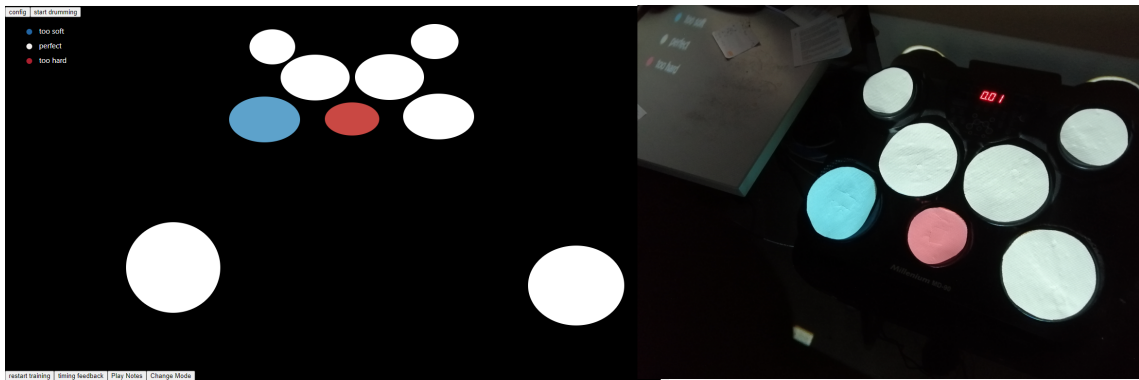
For dynamics feedback, we measure the velocity with which a drum is hit (this translates to the note's loudness) and compare it with the velocity of the corresponding note in the NoteArray of imported notes. We can find the position of this note when searching for the note in the position of the length of the played notes subtracted by one, because counting on NoteArrays starts from 0. We also take this number modulo the length of the imported notes array, so if the user wants to play a certain music piece on repeat, they do not have to restart the session on every repeat. Of course restarting the session is also supported

As discussed in Chapter 3, we use a red-white-blue color scheme to visualize the feedback, d3-scale-chromatic provides such a scheme (Figure 4.4). We color the drum blue if the drum was played too soft, red if it was played too forceful, and white if it was played with the correct force, to get the right color we use the interpolateRdBu function from d3. It requires a number in the range of 0 to 1 and returns the corresponding color of the scheme (where 0 is red and 1 is blue).



**Figure 4.4:** The red-white-blue color scheme we use for dynamics feedback provided by d3-scale-chromatic<sup>1</sup>.

This number is calculated by dividing the velocity of the played note by the velocity of the corresponding imported note. We want to divide this result by two, so when we hit the drum with the ideal force, we get 0.5, which will give us the color we use to indicate the ideal. Because the sequence of colors on our color scheme is reversed compared to what we would like to have (blue for weak and red for strong), we invert the final result. Figure 4.5 shows what our finished implementation looks like on screen and projected on the drums. Note how we can make the color legend visible to the user even on projection when we position the drum next to, for example, a white desk.



**Figure 4.5:** Implementation of dynamics feedback using color visualization. Drums are colored blue when hit too light, red when hit too hard, and white when hit with the ideal intensity. Left: what happens on screen, right: what is projected.

### Timing Feedback

After playing, the user can request retrospective timing feedback. As discussed in Chapter 3, we use a red-yellow-blue scheme this time, once again we can get it from d3-scale-chromatic, see Figure 4.6.

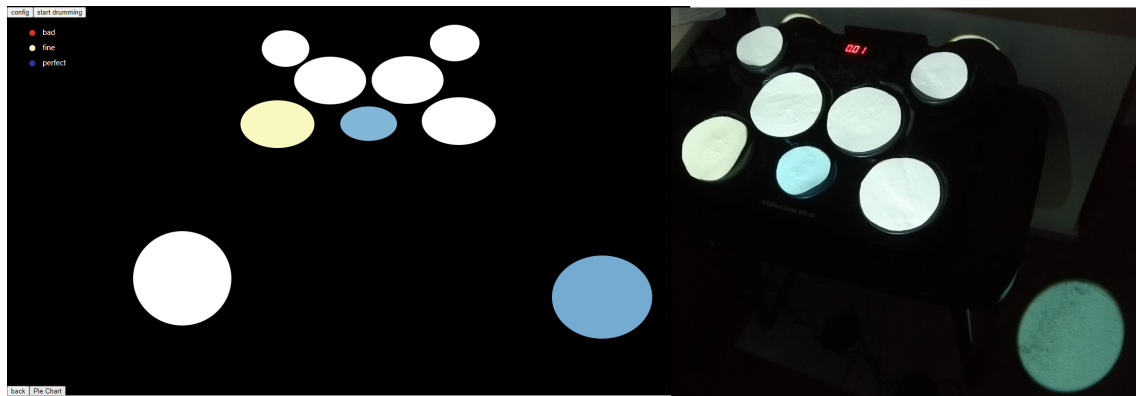


**Figure 4.6:** The red-yellow-blue color scheme we use for timing feedback provided by d3-scale-chromatic<sup>2</sup>.

To find the correct value of the color map we use a kind of score system. We compare the list of notes we recorded from the player with the list of notes the player imported note for note. If the user played more notes than there are in the imported notes NoteArray (when for example the user repeated a phrase several times), we can easily extend the imported notes NoteArray by repeating it

with the `NoteArray` function `repeat`, this function concatenates a time-shifted copy. Before we can compare the two arrays, we also have to change the start time of the first note of the played notes list to zero.

Now we can compare each note of the two `NoteArrays` according to start time. Depending on how much the start time of the played note and the start time of the imported note differs, the user can achieve a certain amount of points on each drum. If the timing difference is within the previously set leeway, the user gets a full point. When the difference is relatively close to the leeway, the user gets half a point, we say it is close if the difference is smaller than 1.2 times the leeway. An error that is smaller than 1.5 times the leeway, still gets awarded with a quarter of a point. By dividing the number of achieved points by the maximum number of points that were achievable on that specific drum, we get a number between zero and one that we can use to get the corresponding color from our scheme. Figure 4.7 shows what the implementation looks like in practice.

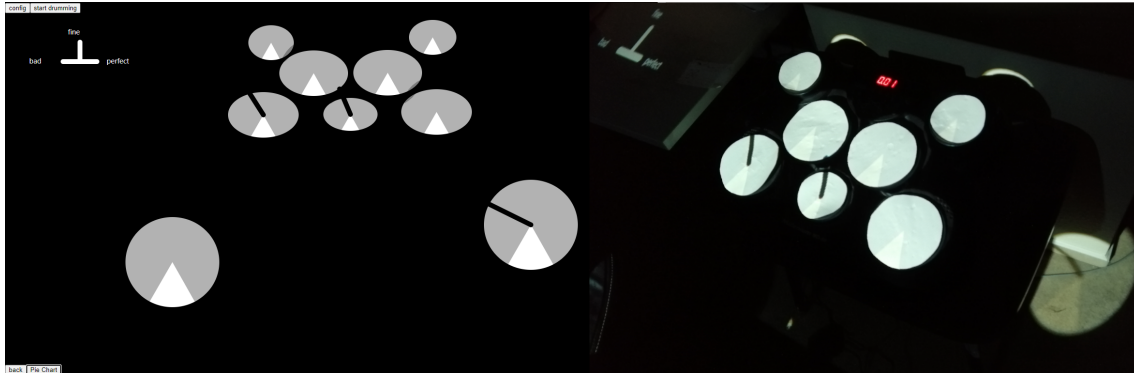


**Figure 4.7:** Implementation of timing feedback using color visualization. Drums are colored with a color more towards blue in the color scheme the more notes were timed correctly on the respective drums. Left: what happens on screen, right: what is projected.

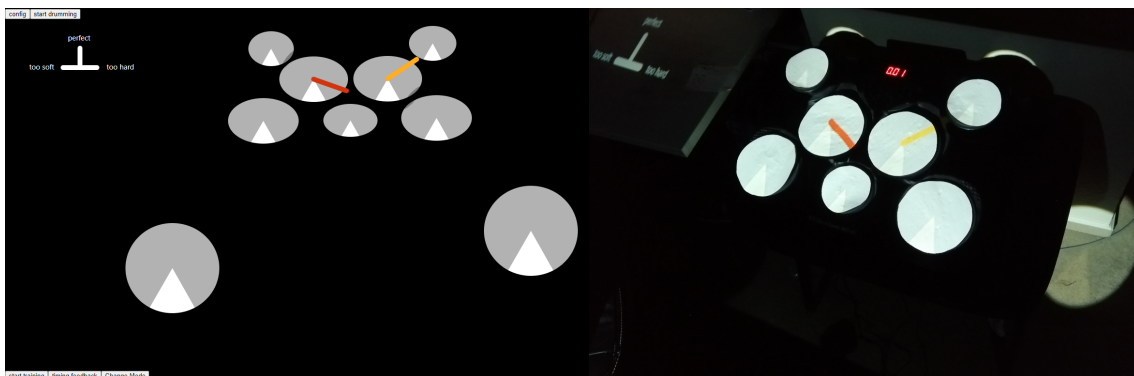
### 4.3.2 Speedometer Visualization

By clicking the “change mode” button, the user can change to the speedometer visualization. We color the part of the ellipses where the needle can move gray, so the user can comprehend the needle movement better. The needle has a room for maneuver of 300 degrees which is approximately 83% of the ellipse.

For visualizing our feedback, we need to find the right angles with which we let our needles deflect. To calculate these, we can use the results we already calculated for the color visualization. These are numbers between one and zero, so we can interpret them as percentages. If we just multiply them by 300, the degrees in which the needle can move, we get a certain percentage of this number which corresponds to the angle we want the needle to deflect. We then draw the needle as a line from the center of a circle to the edge into the SVG that we use for the projections, and move it into the right start position. To let the needle deflect we then use a transform function. Figure 4.8 shows how what implementation looks like for timing feedback, Figure 4.9 for dynamics feedback.



**Figure 4.8:** Implementation of timing feedback using our speedometer visualization. Projected needles deflect more to the right the more notes were timed correctly on the respective drums. Left: what happens on screen, right: what is projected.



**Figure 4.9:** Implementation of dynamics feedback using our speedometer visualization. Projected needle deflects more to the right the harder the drum was hit. Left: what happens on screen, right: what is projected.

### 4.3.3 Pie Charts

After requesting timing feedback, the user can request more detailed information in form of a pie chart by clicking on “Pie Chart” button. In our first implementation, we just counted, while calculating the timing feedback, how many times the user got one, a half, a quarter, or no points on each drum. We divided the pie by the share of how often these points were given concerning all played notes, so the user could see how often they played a note perfect, great, good, or bad.

Later we thought that it would probably be more interesting for the user to see how many notes they played too early or too late, so we changed the division of the pie slices. Now we count the number of notes that were played too early and too late, and those that were played much too early and much too late for every drum. With too early/late we once again mean that the difference between the played note and the imported note is bigger than the set leeway, much too early/late is when the difference is bigger than 1.5 times the leeway. We can see the implementation of both in Figure 4.10. We did not make these pie charts elliptical in our implementation, but as seen in the figures one can barely notice this when it is projected on the drums. Furthermore elliptical pie charts could be confusing, since area sizes are more dominantly perceived than angles.



**Figure 4.10:** Implementations of pie chart visualization. We tried visualizing how accurate notes were played from perfect to miss (left), and visualizing how many notes were played too late, too early, or perfect (right). We decided to use the latter implementation because we believe this information is more useful.

#### 4.3.4 Signal Visualization

Our signal visualization is the only type of visualization where we tried active timing feedback because we think that it lends itself perfectly for that. For this visualization, we only need two colors a red one and a blue one.

We compare the start time of the last played note with its corresponding notes start time in the imported NoteArray every time a drum is played. We find the corresponding note the same way we did for the dynamic feedback, however, we can not use modulo when the list of played notes gets bigger than the list of imported notes. We have to expand this list with the repeat function of NoteArrays or else the start time would not be correct. We also have to subtract the start time of every note in the played note array with the start time of the first played note in that array to make it like that array had its first note start on zero and shift all other notes accordingly.



**Figure 4.11:** Signal visualization implementation. When the user is too slow, the drums are colored red (left), when they are too fast, the drums are colored blue (right).



When a note is played, we check if the start time of the imported note minus the start time of the played note is smaller than zero. If that is the case, we know that the note was played too late, so we set the color to red. Otherwise, the note was played too early and we set the color to blue. We then project the color onto the played drum with the opacity of the timing difference minus the set leeway (if the result is smaller than zero we set the opacity to 0), so if the timing difference was within the leeway, nothing will be projected. If the drum was hit too early, it will be colored blue, if it was hit too late it will be colored red (Figure 4.11).

## 5 Evaluation

Due to the ongoing global pandemic, we were regrettably unable to conduct a user study. Instead, we evaluated our visualization by conducting three case studies, which we describe and discuss in this chapter. We first describe how the study was planned, then how it was conducted, and then we show off the results. Finally we discuss them.

The sole participant in all these studies was the author of this thesis, someone who had no prior experience in playing the drum apart from learning some easy grooves so he could test and debug the prototype while developing it.

We conducted 3 studies, in the first we measured how the number of errors made while playing would change over time when using our visualizations to evaluate if our approach has a positive effect of reducing the errors. In our second study, we intentionally make some typical errors that often happen to drum beginners to observe how the visualizations help to identify them. Finally, in our last study, we let the participant learn one pattern using all three visualization methods with live feedback to compare their performance.

### 5.1 Case Study 1: Evaluating the Learning Effect of Visualizations Over Time

In this case study, the player learns a new groove for every visualization technique. We want to investigate how our visualizations influence the users learning effect over time by observing how the number of errors made evolves over the duration of the sessions.

The participant learned three grooves from scratch, one for the speedometer, one for the signal, and one for the color visualization, as we wanted to avoid learning effects in this study. We conducted three training sessions per visualization, where one session consists of 10 repetitions. In the first session, the groove was played 1.5 times slower, in the next session it was played 1.25 times slower and in the final session, it was played at its original speed. We used a timing leeway of 0.3 for all visualization and all sessions. All sessions were recorded on video, so we could later extract the number of errors made from these recordings. For evaluating the training's effect over time, we observed the trend of errors made over time for each visualization separately.

At first, our inexperienced participant had trouble memorizing the grooves. We have a system that allows him to let the groove play on its own which could help with memorizing, but for our participant, this was not enough, so he had a score of the groove next to him. Especially in the first sessions, the participant had to focus way more on the score than the visualization on the drums to be able to play the groove at all. The player did not pay much attention to the live feedback, he saw it only in his peripheral vision most of the time. The retrospective feedback did not have this

problem. Because of that, we decided to make an additional session just for training dynamics after the player could play the groove without help from the score, because we did not implement a retrospective dynamics feedback system.

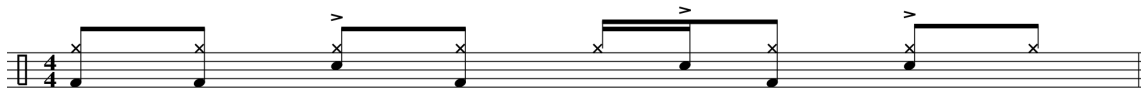
The music pieces we picked were short grooves, two of them with 15 notes, one with 16. We tried to pick the pieces such that they are relatively close in difficulty, preferable easy for our inexperienced player. The three grooves we picked can be seen in Figure 5.1, Figure 5.2 and Figure 5.3. Our participant played the groove one time, got feedback, and then restarted the training, this is what we call one repetition.



**Figure 5.1:** The groove used for evaluating the signal visualization.

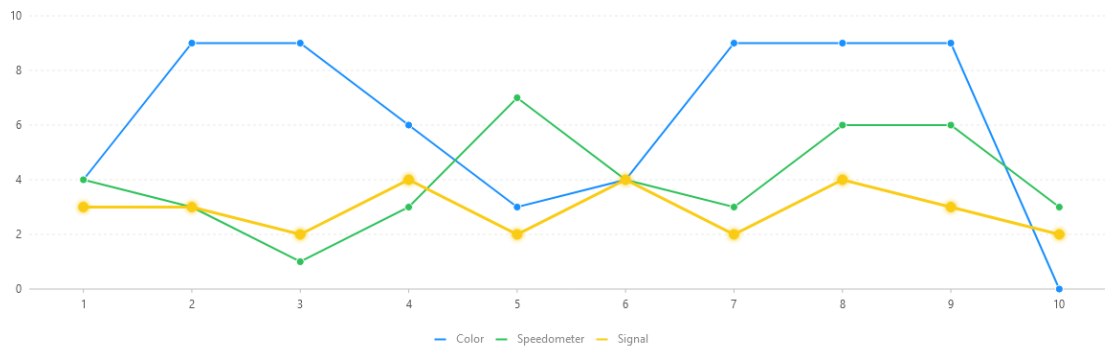


**Figure 5.2:** The groove used for evaluating the color visualization.



**Figure 5.3:** The groove used for evaluating the the speedometer visualization.

We measured in every session how many mistakes were made regarding timing. For color and dynamics feedback we extracted the number of errors from the pie charts, for signal feedback we watched the video recording and counted the times the drums were colored blue or red. The resulting data is visualized in Figure 5.4, Figure 5.5, and Figure 5.6.



**Figure 5.4:** Errors made per repetition in session 1.



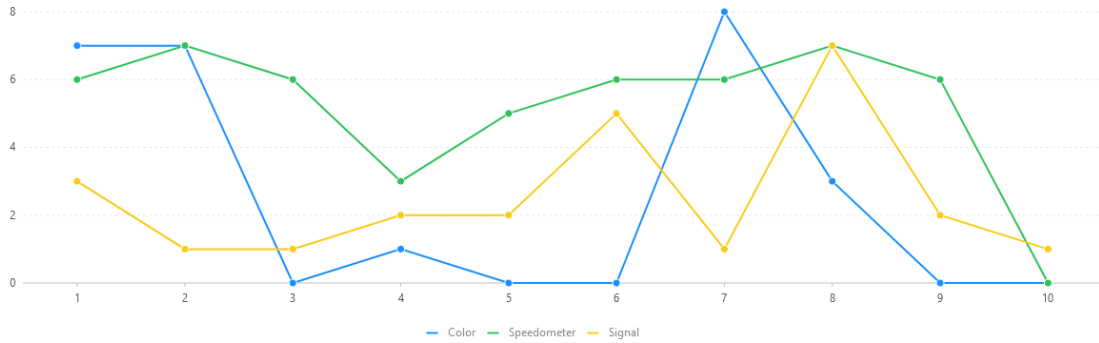


Figure 5.5: Errors made per repetition in session 2.

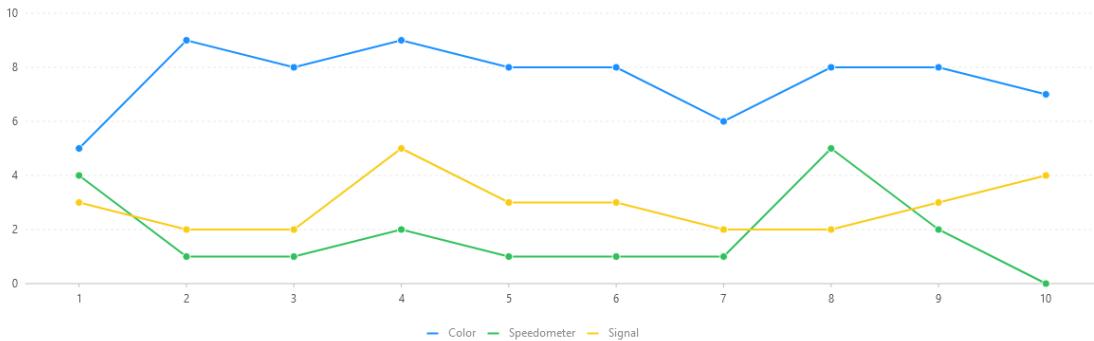
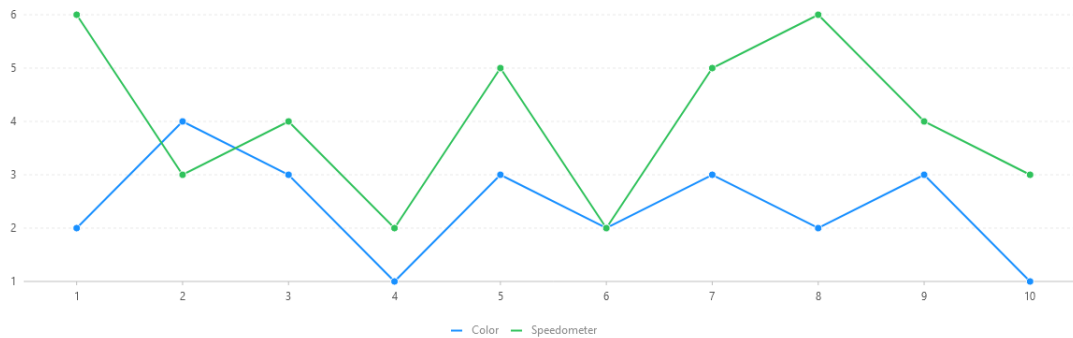


Figure 5.6: Errors made per repetition in session 3.

We evaluated the special dynamics session by how many errors were made regarding dynamics. Since we did not implement a dynamics leeway, we decided to be a bit less strict with counting something as an error, for our speedometer visualization we do not count notes as an error where the needle is within 45° of the center. For the color visualization, we do not count notes where the color is close to white, which is a bit harder to measure. We do not count the pedals, because they do not track how hard they were kicked, see Section 5.4.4. This time the player already knows how to play the groove, so he has no access to the score and has to fully concentrate on the visualization. The resulting data can be seen in Figure 5.7 as line charts.

## 5.2 Case Study 2:Evaluating the Helpfulness in Identifying Errors

In this case study, we let the player play a groove and intentionally make errors to observe how the program reacts. He mimicked several sorts of beginner-level players who make specific mistakes. First, he is going to impersonate a player who is getting increasingly faster while playing. Then he will imitate a player who has trouble playing with both, his hands and his feet, synchronously. Last, he will copy a player who is getting increasingly louder while playing. He will do all this for all types of visualization. Thereby, we can evaluate how a user can interpret the feedback and identify their playing errors with it.



**Figure 5.7:** Dynamics errors made per repetition in the special dynamics session.

### 5.2.1 Increasing Speedup

The color and speedometer visualization give their feedback retrospective, so while playing, a user who increasingly speeds up can not immediately identify this mistake. However, when requesting timing feedback, the player can see that the performance on all drums was bad: in our color visualization, the drums are colored yellow or even red, and in the speedometer visualization, the needle deflects more to the left. The player can still not identify when the mistakes were made or if the drums were played too early or too late. When requesting pie charts, however, he can at least see that his performance was bad because too many notes were played too early. When using the active feedback of the signal visualization, the player can immediately see while playing that the drums turn increasingly blue because he plays too fast, therefore he can adapt by slowing down.

### 5.2.2 Desynchronized Playing

This kind of mistake can be impossible to identify if the user chooses the leeway too high. When using our color visualization and requesting timing feedback, from the two drums that were supposed to be played in sync, one had a worse performance, indicated with a color that goes more towards red on the spectrum. This way the player could identify on which drum the error occurred and adapt accordingly. If the desynchronization is only small, the colors of the two drums can be very similar, which makes it hard to say which drum had the worse performance. This can be seen more clearly when using the speedometer visualization or pie charts.

When using the live feedback of the signal visualization, the participant saw immediately when he played out of sync. For example, when the first drum of the two that are supposed to be played synchronously was played too early it was colored in blue for a short time, a fraction of a second. Because the information is so lightweight it was still enough to identify the error.

### 5.2.3 Player Gets Increasingly Louder

Note that we did implement dynamics feedback for the signal visualization, but it turned out that this was pretty much similar to the dynamics feedback of the color visualization, so we did not test them separately. Also, on our e-drum the foot pedals will always be played too loud, so we do not count errors made on them in the evaluation (see Section 5.4.4).

With the color visualization, when the participant got louder and louder, the drum will go more and more towards red on the color scheme, the feedback is given in real-time, so we can identify our errors immediately and adapt by reducing the amount of force we play the drum with. The speedometer visualization works similarly, the participant got the feedback immediately in that the needle deflected more and more to the right.

## 5.3 Case Study 3: Comparing the Visualizations

In this case study, the player learns a groove using all visualizations in turns, so we can compare the performance of the visualizations. Our participant learned a new groove from scratch (Figure 5.8) he used the color visualization, speedometer visualization, and signal visualization in turns to do this. He conducted three training sessions, where each training session consists of 15 repetitions, for every repetition, the method of visualization was changed, so that it did not affect the results when the player became increasingly better at playing the groove over time. In the first session, the participant played the groove 1.5 times slower, in the second the song was 1.25 times slower, and finally, in the last session, the groove was played at the original speed. The leeway was 0.3 seconds for all sessions and all visualizations. The training sessions were recorded on video, for the evaluation we watch back those recordings and count the errors made, to help with this we analyzed pie charts after every repetition and extract the number of errors from them.

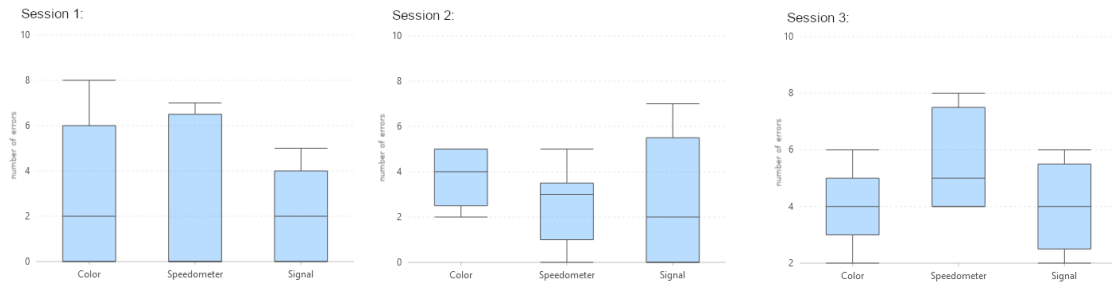


**Figure 5.8:** The groove used for comparing the visualizations.

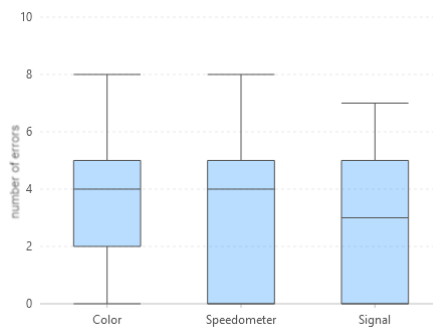
To not repeat what happened in the first case study, the player got himself familiar enough with the groove that he could play it confidently without looking at the score, and fully concentrate on the visualizations (he was able to do so after 3-4 test repetitions). However, he did not get any feedback, so he was unfamiliar with the timing of the groove. The results of the different sessions can be seen in the box plots in Figure 5.9 and Figure 5.10.

## 5.4 Limitations and Discussion

In this section, we shortly discuss the results of our evaluation and then try to interpret them. Afterward, we explain several limitations that hindered our approach, how we dealt with them, and what one might do if they were free from them.



**Figure 5.9:** Results of the case study shown as box plots per session.



**Figure 5.10:** Results of the case study shown as box plots for all sessions combined.

### 5.4.1 Evaluating the Learning Effect

Before we analyze the data, we must remember that it was only obtained by testing the program on a single individual who had no prior experience playing the drums. Therefore, a lot of circumstances could have influenced the data. For example, the participant played three sessions a day, one for every visualization method back-to-back. Because all the pieces have roughly the same BPM he could have gotten used to the beat, and therefore could have performed better on the visualization methods which were played last. To prevent this from influencing the results, we mixed up the order in which the sessions were conducted every day.

Due to the difference in how we counted the errors in signal and the other two visualizations, it could be that we tracked the number of errors in the color and speedometer visualization more precisely than in signal, because reading the pie chart is less prone to error than counting errors from a video. The reason we did this in the first place was that we did not want retrospective feedback to influence the result of the signal visualization in any way, so we could better compare the approaches. One major problem is that we do not have enough data to truly evaluate the visualizations. To do this properly, a user study has to be conducted. Because this was not possible for us, we leave it for future work. For now, let us draw some conclusions from the data we do have.

Apart from some unusual spikes in session two, the participant mostly stayed at a consistent level when using the active feedback from the signal visualization. The performance of the other two visualizations fluctuated more, and there is no real trend to be seen. This could be due to the participant trying to adapt to the retrospective feedback, sometimes wrongly. The results of the

color visualization seem to fluctuate more than the ones of the speedometer visualization. This could suggest that the user can read the speedometer visualization more precisely and is therefore, more efficient in identifying the errors.

In the special dynamics session, color seems to perform a bit better than the speedometer visualization. The player noted that the speedometer needle was harder to read than the color if it only appeared for a fraction of a second. Another reason for the differences in performance could be that it is more precise to read, so when we evaluated the videos, it was easier to say that the needle was not in the 45 vicinity of the middle, than it was to say that the color was not close enough to white. Binning the color into, for example, five or seven different hues could help here, but it would also reduce the overall expressiveness. To truly identify the reason, a user study with more participants is necessary.

In the color visualization, errors mainly occurred on the last four snare drum hits of the groove. This was easily identifiable, because timing feedback gave this drum the worst color, yellow or sometimes even red. In the signal visualization sessions, errors occurred mostly at the last four notes. This could be identified immediately due to the live feedback. Since the player was often focusing more on the score than on the live visualizations, he rated the retrospective feedback as more helpful to identify his errors, because he could not miss it, even if the data suggested that he was more consistent with the signal visualization that lacks the retrospective feedback. Our participant noted that trying to get the best possible result in the retrospective feedback had a positive effect on his motivation.

In the end, there is no real trend to be seen in the data we acquired. However, we believe that this is mostly due to us not having the time to conduct a study like this over a longer period of time. A longitudinal study with more participants has to be conducted to truly see the effect our visualization have on reducing the errors in performances over time. These results should then also be compared to how the error of a learner would reduce naturally, without help from any visualization, as a control group. We leave this for future work.

#### **5.4.2 Evaluating Helpfulness in Identifying Errors**

Our observations suggest that the active feedback is ideal for identifying where the mistakes that are made while playing happen. However, this information is only present for a short time, a fraction of a second if the pattern is fast, so it is questionable if the user can perceive and understand all this feedback that fast. Our participant did not have too much trouble with understanding the color and signal visualization even in fast patterns, but he could have trouble with reading the information of the speedometer visualization if it was gone too fast. This could suggest that the glanceability of this visualization is a bit worse than we assumed, however, to truly certify this, more feedback and data is necessary.

The retrospective feedback did not run into this problem, as the user has as much time as they like to read the feedback. It is not possible however, to read where exactly the mistakes happened, but one can see on which drum the performance was worse, so if one knows the music score one gets a general idea on which part the mistake could have occurred. To build a retrospective feedback system with which it is easier to identify when exactly a mistake happens, one could implement a system that could play back the recorded notes at whatever speed the user wants and visualizes the errors at the same time. We leave this for future work.

We conclude that our approach can be useful to identify one's own errors, whereat both active and retrospective feedback have their up and downsides, it is probably best to use both in combination, to maximize the usefulness in finding the errors. How these systems can increase motivation and help to maximize the training effect of sessions is something that can be researched through longitudinal user studies in future work.

### 5.4.3 Comparing Visualization Methods

We analyzed pie charts after every repetition, so we got retrospective feedback for every type of visualization, even though our signal visualization usually lacks it. Therefore, this case study comes down more to which visualization is the least distracting and therefore most helpful in reducing errors.

All three visualizations were relatively close in performance, although signal performed a bit better than the other two visualizations. Generally, the speedometer visualization performed a bit better than color visualization, except in session 3, where the pattern was the fastest, and the color visualization could even match the signal visualization.

This data suggests that the signal visualization is the least distracting. This was to be expected, because it only appears when the player makes a mistake. A bit more surprising is the fact that the speedometer visualization seems to be a bit less distracting than the color visualization. This could be because it only covers a small portion of the drum, while in color the whole drum is covered. This means that there is less change in the speedometer visualization than in the color visualization which is probably why it appears to be less distracting. However, when the pattern gets faster, speedometer's performance goes down, the participant noted that he had trouble reading the speedometer when it vanishes too fast, a failed attempt at reading data could be more distracting than the colors usually are.

Note that even when we say a visualization is distracting, we still think that it helps the user to reduce their errors more than when playing without visualizations. However, the less distracting a visualization is, the more it is going to help them. There is a trade-off between the amount of available information and the amount of distraction. To truly evaluate this future work could conduct a user study where all these kinds of visualization are compared to playing without any visualization feedback. Overall, every method of visualization seems to have it's up and downside, it is up to future work to conduct a user study and survey the participants to find out which method people find the least distracting, the most informative, and how all of these influence a user's motivation.

### 5.4.4 Hardware Limitations

The projector we worked with only had a physical resolution of 800×480 pixels. Paired with the small surface of the drums, many visualization techniques were not possible. For example, we had the idea to implement a sunburst diagram<sup>1</sup>, but due to the low resolution it would have been too hard to read. With a higher resolution, this might be a fitting visualization technique for our ends, that future work could attempt.

---

<sup>1</sup>[https://datavizcatalogue.com/methods/sunburst\\_diagram.html](https://datavizcatalogue.com/methods/sunburst_diagram.html)

We also had to work around the limitations of our e-drum. The pedals for the bass drum and the hi-hats did not measure the force with which they were kicked, they were always played with maximum velocity, which is why our dynamics feedback was not usable for them. Also, the pads of the cymbals and the hi-hat were more sensitive when it comes to velocity than the other pads. This is realistic, but unintuitive for an inexperienced player because apart from the size they look the same than the other pads. In contrast, the e-drum sometimes did not recognize a hit on the snare drum or one of the toms and to the end, we could not figure out why this was happening.

#### **5.4.5 Software Limitations**

The prototype we implemented only compared time and velocity of the played notes and the imported notes, we did not check if the note that was played was the correct one, so the prototype could not give feedback on playing wrong notes (aside from the mistiming an error like this often brings along). The prototype did not consider the last error when calculating the timing error, so when a note was played with the wrong timing all the other notes would be off too if the player does not adapt to their error. This was negligible for our prototype, because we only designed it for short music pieces like grooves that are not much longer than 15 notes. However, if future work wants to test the efficiency of their visualizations in a whole song, they should consider a system that is more flexible in calculating the timing error, or else even only small errors per note could add up fast and evaluation of the player's performance would be too harsh.

Something we did not find an ideal resolution for, is the placement of the visualization for the pedals. There is no associated pad for them, so we need to consider another placement for their visualization. In our prototype, we projected big ellipses onto the floor next to the left and right side of the drum kit. These could be hard to see if the user did not gaze on the kit at the correct angle, also the ellipse for the hi-hat pedal was always occluded by the player when we did the case study. This did not matter for us because we did not test a music piece where this pedal had to be used, but future work should look into alternative placements. For example, one could represent the pedals as circles on the wall, however, one has to watch out that the information is not spread out too much to still allow glanceability.

#### **5.4.6 Conceptual Limitations**

Our aim is to give helpful feedback on a training session like a professional teacher would do, however, said teacher would of course give more feedback than just on timing and dynamics. Things like how to hold the drum sticks, or how to execute certain drumming techniques, are things that are hard to teach with only using AR techniques. For now, aspiring drummers without access to a music teacher have to make do with online tutorials.

With our hardware it was only possible to use two dimensional visualizations. By using a head-mounted display like, for example, a HoloLens<sup>2</sup> one could also explore the possibilities that three dimensional visualizations could lead to.

---

<sup>2</sup><https://www.microsoft.com/en-us/hololens/>

#### **5.4.7 Evaluation Limitations**

The only player that could test our prototype, the author of this thesis, had no prior experience of playing the drum other than learning a few easy grooves to test the implementation in development. An experienced player would probably have been able to provide more useful data in the case studies. Because we could not conduct a user study, we only acquired little data, which limits the conclusions we can draw from it. With more data from a wide variety of users, more insightful evaluation results could have been possible.



## 6 Conclusion and Outlook

In this thesis, we introduced methods, that aim at giving aspiring drummers feedback on their training sessions. To realize this, we used projector-based augmented reality technology and several visualization methods. To project these visualizations directly on the drum we needed a method to track them. We tried Hough transformation, but we found it to perform badly in regard to both quality and computation time with the implementation we used. Therefore, we decided to let the user track the drums manually.

Feedback is given on both timing and dynamics, two important aspects of playing that are hard to self-evaluate. This feedback can be given as active feedback while playing, and retrospective feedback after a session is finished. For both cases, we tried different methods of visualization. Color visualization, where the feedback is given by projecting a certain color on the drums. Speedometer visualization where feedback is given through a needle projected on the drums and how much it deflects, much like on a speedometer in a car. Signal visualization, which we only use for active timing feedback, where signals only are given, if the player makes a mistake in timing, red if too late, blue if too early. And finally pie charts, which we only use for retrospective feedback, where the slices correspond to how many notes were played too late, much too late, too early, or much too early.

We conducted case studies to evaluate our visualizations. There, we tried to evaluate the effect of our visualizations on reducing errors over time. However, since we did not have enough data to identify a clear trend, we conclude that a real user study over a longer period of time would be necessary to get useful results. Signal visualization was helpful for identifying timing errors right when they happened. However, in fast patterns, the signal could only be seen for a fraction of a second. The retrospective timing feedback of the other visualizations did not have that problem, but the user could not see when the mistakes happened. Still, they got a general idea on which drums and how often they occurred. Pie chart and speedometer visualization were better when it comes to reading the information more precise. In contrast, color and signal were better for quickly reading the information. We conclude that our approach could help to identify their errors. When directly comparing our visualizations they were all close in performance. Although signal had a few less errors than the other two, which suggests that it is the least distracting visualization method.

### Outlook

Unfortunately, we were not able to conduct a longitudinal user study with a wide variety of participants. Seeing how users would review the prototype, which visualization they would prefer, and how these would influence their motivation, would have been interesting to investigate. Additionally, it would be interesting to repeat our case studies as user studies, to get more data and more robust results. We would also would like to see how beginners and advanced players would rate the visualizations differently.

One thing our approach cannot do, but a professional teacher can, is to give guidance to the player. It would be interesting to combine a program like ours with a program that would automatically create training regimes for the player based on their performance [Kon20], or to include our approach in a music lesson with a teacher. Another thing that could be done, is comparing our AR approach to an approach that tries to teach the drum within a fully immersive VR system, not using a real drum at all. One could compare which of them has a better learning transfer, similar to what Johnson et al. [JDT19] did with their theremin learning systems. This could provide new insights into whether the results of their study are transferable to other instruments.

The prototype we developed could be further extended in future work. There is room for improvement here, for example, we did not find a satisfying position for visualizing the pedals. We decided to place them as big ellipses on the floor, but this led to some problems (5.4.5). Future work could test out different locations to find one that can provide good visibility while not reducing the glanceability too much. One could also improve the detection of errors. In our prototype, we do not consider the last error when calculating timing error, which can lead to a higher error in the next played note. Future work should consider this, especially when they want to use their program for longer patterns. One could also explore the types of feedback more, for example develop a system for retrospective dynamics feedback.

We limited ourselves to the usage of projector-based augmented reality technologies, which only allow using 2D visualizations. However, future work could investigate the possibilities of other AR technology in drum feedback. For example, one could use a head-mounted display like a HoloLens<sup>1</sup> to make using 3D visualization possible. It would be interesting to see which kinds of feedback visualization would be possible in 3D and how they would compare to our 2D visualizations.

We would also like to see which other types of visualization would be possible for a team that does not have the same hardware limitation that we have. If one uses a projector with higher resolution, how could visualizations like a sunburst or a radial bar chart perform? Would something like small bar charts be viable? Questions like that would be interesting to evaluate.

---

<sup>1</sup><https://www.microsoft.com/en-us/hololens/>

## Bibliography

- [AGSH20] K. Andersen, S. J. Gaab, J. Sattarvand, F. C. Harris Jr. “METS VR: Mining evacuation training simulator in virtual reality for underground mines”. In: *Proc. 17th International Conf. Information Technology: New Generations (ITNG)*. 2020 (cit. on p. 12).
- [AIH+19] M. F. Abrar, M. R. Islam, M. S. Hossain, M. M. Islam, M. A. Kabir. “Augmented Reality in Education: A Study on Preschool Children, Parents, and Teachers in Bangladesh”. In: *International Conf. Human-Computer Interaction (HCI)*. Springer. 2019, pp. 217–229 (cit. on p. 12).
- [BBB+18a] T. Blascheck, L. Besançon, A. Bezerianos, B. Lee, P. Isenberg. “Glanceable visualization: Studies of data comparison performance on smartwatches”. In: *IEEE trans. visualization and computer graphics (TVCG)* 25.1 (2018), pp. 630–640 (cit. on p. 15).
- [BBB+18b] T. Blascheck, L. Besançon, A. Bezerianos, B. Lee, P. Isenberg. “Perception des visualisations sur smartwatch”. In: *Journée Visu.* 2018 (cit. on p. 15).
- [BI21] T. Blascheck, P. Isenberg. “A Replication Study on Glanceable Visualizations: Comparing Different Stimulus Sizes on a Laptop Computer”. In: *IVAPP Proc. Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VIS-APP)*. Vol. 3. SCITEPRESS-Science and Technology Publications. 2021, pp. 133–143 (cit. on p. 15).
- [CLTK10] J. C. Chan, H. Leung, J. K. Tang, T. Komura. “A virtual reality dance training system using motion capture technology”. In: *IEEE trans. learning technologies (TVCG)* 4.2 (2010), pp. 187–195 (cit. on p. 12).
- [Dat18] Data-to-Viz.com. *The issue with pie chart*. 2018. URL: <https://www.data-to-viz.com/caveat/pie.html> (visited on 06/10/2021) (cit. on p. 24).
- [DCPL19] G. De Luca, N. Choudhury, C. Pagiatakis, D. Laroche. “A Multi-procedural Virtual Reality Simulator for Orthopaedic Training”. In: *International Conf. Human-Computer Interaction (HCI)*. Springer. 2019, pp. 256–271 (cit. on p. 11).
- [HBDH10] S. Holland, A. J. Bouwer, M. Dalgelish, T. M. Hurtig. “Feeling the beat where it counts: fostering multi-limb rhythm skills with the haptic drum kit”. In: *Proc. fourth international Conf. Tangible, embedded, and embodied interaction (TEI)*. 2010, pp. 21–28 (cit. on p. 14).
- [HJLH19] F. Heinrich, F. Joeres, K. Lawonn, C. Hansen. “Comparison of Projective Augmented Reality Concepts to Support Medical Needle Insertion”. In: *IEEE trans. visualization and computer graphics (TVCG)* 25.6 (2019), pp. 2157–2167 (cit. on p. 12).
- [Hou62] P. V. Hough. *Method and means for recognizing complex patterns*. US Patent 3,069,654. Dec. 1962 (cit. on p. 19).

- [HZY+11] F. Huang, Y. Zhou, Y. Yu, Z. Wang, S. Du. “Piano AR: A markerless augmented reality based piano teaching system”. In: *IEEEET international Conf. on intelligent human-machine systems and cybernetics (IHMSC)*. Vol. 2. IEEE. 2011, pp. 47–52 (cit. on p. 13).
- [IMS16] K. Iguchi, H. Mitsuhara, M. Shishibori. “Evacuation instruction training system using augmented reality and a smartphone-based head mounted display”. In: *IEEE international conf. information and communication technologies for disaster management (ICT-DM)*. 2016, pp. 1–6 (cit. on p. 12).
- [JDT19] D. Johnson, D. Damian, G. Tzanetakis. “Evaluating the effectiveness of mixed reality music instrument learning with the theremin”. In: *Virtual Reality* (2019), pp. 1–15 (cit. on pp. 13, 46).
- [Kal05] L. C. Kalloniatis M. *The Perception of Color*. 2005. URL: <https://www.ncbi.nlm.nih.gov/books/NBK11538/> (visited on 07/18/2021) (cit. on p. 23).
- [KJO19] T. Khan, K. Johnston, J. Ophoff. “The impact of an augmented reality application on learning motivation of students”. In: *Advances in Human-Computer Interaction* (2019) (cit. on p. 12).
- [KM13] P. Kay, L. Maffi. “Number of Basic Colour Categories”. In: *The World Atlas of Language Structures Online*. Ed. by M. S. Dryer, M. Haspelmath. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. URL: <https://wals.info/chapter/133> (cit. on p. 23).
- [Kon20] M. Konecki. “Adaptive Drum Kit Learning System: Impact on Students’ Motivation”. In: *IEEE 43rd International Conv. Information, Communication and Electronic Technology (MIPRO)*. 2020, pp. 808–812 (cit. on p. 46).
- [Mik19] A. Mikhailov. *Turbo, An Improved Rainbow Colormap for Visualization*. 2019. URL: <https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html> (visited on 06/04/2021) (cit. on p. 23).
- [MK94] P. Milgram, F. Kishino. “A taxonomy of mixed reality visual displays”. In: *Trans. Information and Systems (IEICE)* 77.12 (1994), pp. 1321–1329 (cit. on p. 11).
- [Moo86] R. A. Moog. “MIDI: Musical Instrument Digital Interface”. In: *Journal of the Audio Engineering Society* 34.5 (1986), pp. 394–404 (cit. on p. 10).
- [MS06] Y. Motokawa, H. Saito. “Support system for guitar playing using augmented reality display”. In: *IEEE/ACM International Symp. Mixed and Augmented Reality (ISMAR)*. 2006, pp. 243–244 (cit. on p. 13).
- [MWM+21] K. Marky, A. Weiß, F. Müller, M. Schmitz, M. Mühlhäuser, T. Kosch. “Let’s Frets! Mastering Guitar Playing with Capacitive Sensing and Visual Guidance”. In: *ACM Extended Abstracts Conf. Human Factors in Computing Systems (CHI)*. 2021, pp. 1–4 (cit. on p. 13).
- [pgu14] pguyot. *Hough transform implementation*. 2014. URL: <https://gist.github.com/pguyot/26149fec29ffa47f0cfb> (visited on 06/07/2021) (cit. on p. 19).
- [Pso95] J. Psotka. “Immersive training systems: Virtual reality and education and training”. In: *Instructional science* 23.5-6 (1995), pp. 405–431 (cit. on p. 11).
- [SHN] M. Speicher, B. D. Hall, M. Nebeling. “What is mixed reality?” In: *ACM Proc. Conference on Human Factors in Computing Systems (CHI)*, pp. 1–15 (cit. on p. 11).

- [SYQ+20] X. Shao, Q. Yuan, D. Qian, Z. Ye, G. Chen, K. le Zhuang, X. Jiang, Y. Jin, D. Qiang. “Virtual reality technology for teaching neurosurgery of skull base tumor”. In: *BMC medical education* 20.1 (2020), pp. 1–7 (cit. on p. 11).
- [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: *ACM Conf. Human Factors in Computing Systems (CHI)*. 2015, pp. 4123–4132 (cit. on p. 12).
- [YAKN11] T. Yamabe, H. Asuma, S. Kiyono, T. Nakajima. “Feedback Design in Augmented Musical Instruments: A Case Study with An AR Drum Kit”. In: *IEEE 17th International Conf. Embedded and Real-Time Computing Systems and Applications (RTCSA)*. Vol. 2. 2011, pp. 126–129 (cit. on pp. 14, 21).

All links were last follows on August 5, 2021.



### **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