G.L.I.D.E.: Correlating Real-Time Multi-Sensor Data to Figure Skating Movement Patterns as an Interface for Musical Expression

Alyssa Aska
University of Music and
Performing Arts Graz
alyssa.aska@gmail.com

Martin Ritter
University of Music and
Performing Arts Vienna
ritter-martin@mdw.ac.at

Jeffrey E. Boyd
University of Calgary
jboyd@ucalgary.ca

ABSTRACT

Motion tracking and movement capture are effective tools for creating new interfaces for musical expression. Tracking systems can encompass everything from computer vision to physical gestural control. The authors have developed a system for tracking the movement of a figure skater in real time using an M5StickC Plus2 laced into a skater's boot. The authors conducted experiments with this setup by recording skater movement data along with corresponding video. This data was then played back and applied to musical parameters and audio processes to create several works the authors describe as artistic documentation, or documentation modified for aesthetic purposes to become a standalone artistic work. A previously developed tool for tracking speed skaters was enhanced and combined with artistic research into gestural controllers for musical expression. Figure skating is already closely linked with music and with a framework that favors musicality and intimately connects gesture with music. Additionally, many of the movements within figure skating contain variations in direction, speed, and movements on various axes. For this reason, we decided to explore the potential of figure skating as not just a tool of embodying music through movement, but also as using movement as an extension of musical expression. Figure skating offers a compelling and novel way to explore the connection of physical movement with music and sound.

1. INTRODUCTION

G.L.I.D.E. (Gesture Logging for Ice-based Dynamic Expression) is used for tracking the movements of a figure skater with the intention of using the data for artistic purposes. G.L.I.D.E. focuses on exploring figure skating movements as a means of musical expression. The system captures these movements to create an interactive, adaptive, or reactive musical and movement performance environment. G.L.I.D.E. tracks a skater's movements in real-time and can then use these movements to control parameters such as sound synthesis or processing. The system is based on a previously developed tool for tracking speed skaters [1–3], which has been adapted to track figure skaters. The system uses an inertial measurement unit (IMU) to track the skater's movements and streams the data to a host computer, where it is processed and used to control sound synthesis or processing. The end goal of the project is to use the system in a live performance context, where a skater's movements can be used to control sound in real-time.
G.L.I.D.E. is to be flexible and adaptable, allowing for a wide range of musical and musical and movement possibilities and aims to bridge the gap between physical movement on the ice and musical expression. This system therefore provides a novel platform for creative exploration and performance.

2. GESTURAL CONTROL FOR MUSICAL EXPRESSION — PRECEDENTS

The authors of this paper have already developed several systems for gesture tracking that enable creative musical expression [4–7]. These involve the use of software tools such as computer vision, as well as hardware tools such as *Wiimotes*, *GameTrakPros*, *LeapMotion Controller*, and other devices that have accelerometers and/or gyroscopes.

2.1 Computer Vision

One of the initial precedent to the authors' exploration into motion tracking was IMuSE [1], a SSHRC [2] funded project, which took place at the University of British Columbia beginning in 2009 [7]. This project researched ways to use computer vision to track the movements of a pianist's hands. This, combined with existing software called antescofo [8–10], developed at IRCAM, was used to determine a players' temporal location in the score. This data could then be used to adjust the electronics in real time. While not originally intended to function as a gestural controller, the software developed as part of this research was integrated into several artistic works. The first such work, Martin Ritter's *insomniac's musings* (2011) was the first work to implement IMuSE's motion tracking in an artistic context. This was followed by Alyssa Aska's *Concurrent Shifting* (2012), and Keith Hamel's *Touch* (2013) [11]. Ritter developed a series of Max patches based on the cv.jit toolbox for musical expression in response to this interesting artistic application [12], which were also used in Aska's *The Woman and the Lyre* [13], and used by Aska to develop and integrate a diffusion system based on motion tracking for performative contexts. This diffusion system was used during a performance of Aska's

[1] Integrated Multimodal Score-Following Environment
[2] Social Sciences and Humanities Research Council

fixed media work *City of Marbles* (2016) at the Canadian University Music Society conference in 2016.

This concept was developed further with the implementation of computer vision to track the hammers inside of the piano and determine which key was struck, resulting in a similar system of spatialization, in which the pitch played would determine the speaker output [6].

2.2 Physical Gestural controllers

The authors have also undertaken research into the use of other motion and gesture-tracking controllers to translate movements into musical material and processes. Ritter developed an external for the LEAP MotionLEAP motion [3] to interface with Max/MSP [4]. This allowed composers of music that used Max to integrate the LEAP motion as an expressive musical device. It was explored first as an extension to the computer vision tracking of the piano, especially due to its higher precision and identification of specific gestures. Aska composed two works, *Prelude/Hyperkenisis* (2014) and *Circles* (2014) for piano and Leap motion that focused specifically on precisely tracked gestures.

[13] expanded the integration of the Leap Motion to allow a vocalist to adjust parameters of the music and sound output with their hand gestures in a more nuanced way than the computer vision motion tracking.

Video game controllers were also explored by the authors, including the Kinect [14,15] in Aska's *Kinesis* (2014) for violin and electronics, and Wiimotes and GameTrak for various laptop ensemble works between 2011–2013. All of these works and explorations provided valuable platforms for experimentation and development of software. However, none was regularly integrated into the musical works of any of the authors. Research and development of these systems did provide useful insight into interactive systems. This informed later interactive compositions by the authors. Particularly their later exploration into gamified works [16] provided a framework and platform for interactivity that the authors continue to develop for the project discussed in this paper.

3. TECHNICAL SPECIFICAITONS

3.1 Motion Measurement and Analysis

G.L.I.D.E. uses a system that measures a skater movement with an inertial measurement unit (IMU). The IMU is placed on the skater's body in a repeatable location. Measurements from the IMU then streamed by a local area wifi network to a Max/Pd patch running on a host computer. The patch is based on objects developed by Godbout et al. [1–3], and was designed to recognize and synchronize to periodic motions. The output of the recognition and synchronization process can be used to trigger sound events, thus allowing the skater to control sound through their skating movements.

An M5StickC Plus2 [4] (M5) provides a low-cost, general-purpose ESP32 microcontroller platform to serve as the IMU. The M5 features an MPU6886 with a 3-axis accelerometer and 3-axis gyro to measure acceleration and rotational velocities. The controller was programmed using MicroPython to establish a wifi connection to a host network, acquire accelerometer and gyro measurements from the MPU6886 at $50Hz$, and stream those measurements to a host computer using OSC [17]. The M5 also includes a battery with sufficient capacity to operate in this fashion for extended skating sessions (60 minutes or more).

Godbout and Boyd [1] show how to synchronize a system to the motion of a person measured from a stream of scalar measurements. Godbout et al. [3] extend the concept to work with streams of vectors, specifically from a 3-axis accelerometer. To synchronize we start with a model of the motion, $\mathbf{g}_n(i)$ of length n, indexed in time by i. Elements of $\mathbf{g}_n$ are an estimate of the measurements that we expect to see. The IMU provides us with a stream of measurements, $\mathbf{f}(i)$. To synchronize, we compute a match, $h(s,n)$ between $\mathbf{f}$ and $\mathbf{g}$ using

$$h(s,n) = \sum_{i=0}^{n-1} \hat{\mathbf{f}}(i) \cdot \hat{\mathbf{g}}_n((i+s) \bmod n), \tag{1}$$

where

$$\hat{\mathbf{f}}(i) = \frac{1}{\sqrt{\sum_{k=0}^{n-1} \mathbf{f}(k) \cdot \mathbf{f}(k)}}\mathbf{f}(i), \tag{2}$$

$$\hat{\mathbf{g}}(i) = \frac{1}{\sqrt{\sum_{k=0}^{n-1} \mathbf{g}(k) \cdot \mathbf{g}(k)}}\mathbf{g}(i), \tag{3}$$

$$\tag{4}$$

and s is a circular shift of the model that reflects the current phase of the periodic motion. For each sample we compute the period of the optimal match, $\hat{n}$ with

$$\hat{n} = \arg\max_n \max_s h(s,n), \tag{5}$$

and the current phase, ϕ, with

$$\phi = \frac{1}{\hat{n}} \arg\max_s h(s,\hat{n}). \tag{6}$$

The optimal value of $h(s,n)$ indicates the quality of match. The normalization done in Equations 2 and 3 results in $h = 1$ for a perfect match. The objects implemented in Max/Pd accept a stream of measurements, $\mathbf{f}$ h, $\hat{n}$ and ϕ.

While the original purpose of the synchronization system was to match repetitive motions such as walking, running, and skating, the method can be used for transient motion. For a transient motion, one observes low values of h, until the individual begins the modelled motion. As the happens, the value of h rises and eventually exceeds a threshold (typical threshold values are 0.8 to 0.85), to indicate that the modelled motion is occurring. In authors' experience, the h exceed the match threshold between one third and one half of the way through the motion. While this does not allow one to synchronize other process to the full motion, one can recognize when the model motion is occurring, and synchronize to the latter part of that motion.

Placement of the sensor on an individual's body is critical. The model consists of expected measurements, so the

[3] https://www.ultraleap.com/leap-motion-controller-overview/

[4] https://docs.m5stack.com/en/core/M5StickCPLUS2

position on the body must match the position used when acquiring the model. Beyond that, there is freedom to choose where the IMU should be placed (they are light and easy to mount). In speed skating, Godbout et al. [3] used a 3-axis accelerometer worn on the chest with excellent results. However, figure skating features a larger array of movements. A sensor worn on the chest, or lower back would be excellent for movements centered about the torso. Sensors could be worn on the individual skates, with the constraint that an IMU on the left foot tells one little about the motion of the right. In practice, depending on what motions are of interest, two or more IMUs may be necessary, provided that they do not exceed the limits imposed by network bandwidth and computation.

While Godbout et al. [3] developed the original synchronization system based on a 3-axis accelerometer, it is possible to use gyro measurements instead – the process remains the same. Care must be taken if accelerometer and gyro measurements are used together – they have different units, so one must scale the gyro units in $\mathbf{f}$ and $\mathbf{g}$. I.e., a single measurement vector will have six elements as follows:

$$\mathbf{f}(i) = \begin{bmatrix} \mathbf{f}_{accel}(i) \\ \alpha \mathbf{f}_{gyro}(i) \end{bmatrix}, \tag{7}$$

where $\mathbf{f}_{accel}(i)$ is a 3-element vector of accelerations, $\mathbf{f}_{gyro}(i)$ is a 3-element vector of angular velocities, and α is a scalar. In practice, α is tuneable, but selecting α such that the largest acceleration in the model is equal to the largest rotational velocity is a good starting point. It is necessary to scale both the model and the live IMU gyro measurements.

3.2 Software Implementation

In order to use the data from the M5StickC Plus 2, we wrote a Micropython [5] script that interfaces with the sensors, connects to a network, and creates OSC bundles to be sent over the network to a receiver. We implemented a simple low-pass filter to improve the sensor data. The low-pass filter can be expressed as:

$$\text{filtered}_i = \alpha \cdot \text{new}_i + (1 - \alpha) \cdot \text{old}_i \tag{8}$$

where α is the smoothing factor, new_i is the new data point, and old_i is the previous data point. The smoothing factor α is a value between 0 and 1 and has to be tuned to get appropriate results. A starting value of 0.2–0.25 has been shown to give good results with the M5 hardware. We ensure that the data used for synchronization is smooth and less affected by sudden spikes or noise, leading to more accurate and reliable performance of the system. We apply this filter to both the accelerometer and gyroscope data. Since we are tracking jumps and spins, we need to be careful with the filter settings to avoid losing important information about the motion. The filter settings can be adjusted based on the specific requirements of the application and the desired level of noise reduction.

A custom Max patch received the OSC messages sent by the M5. These messages can be saved for future analysis.

To visually verify that the data matches the skater's movements, a video of the skater was made while the data was recorded. At the start and end of the video, the computer screen is recorded with the current time-code visible. This allows us to synchronize the data and video by calculating offsets using the different frame rates for the data and video recording. The synchronization patch also allows the user to zoom into the data to get a better understanding of the local changes. Selections can be made, which can be played back, looped, and slowed down or sped up. Data that is played back is forwarded to the tracking patch.

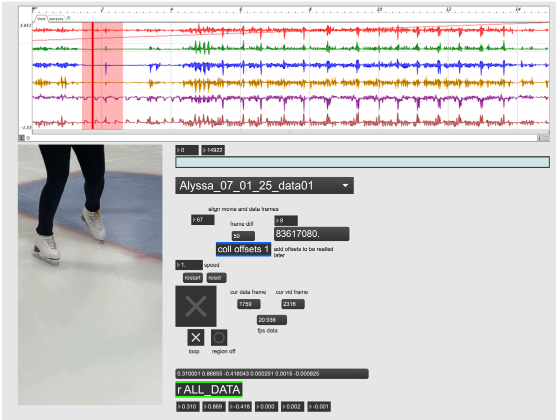

Figure 1. Max Patch to synchronize data to video and allow for data playback. Top 3 data streams represent accelerometer data, the bottom three data streams the gyroscope data

A new external is currently in development by the authors using the min-devkit [6] by Cycling74, in order to use C++ as the main development environment. The external — currently named *correlate* — is responsible for learning the models, and subsequently matching incoming real-time data to these models. *correlate* is agnostic to the size of data it receives. It is therefore possible to designate the vector size of the data that is expected, which means a single accelerometer can be used, or several different types of measurements by different sensors simultaneously. As mentioned in Section 3.1, we are currently using a combination of accelerometer data and gyroscope data, needing to process a 6-element vector. However, the vector size is kept agnostic, keeping future expansions of the system in mind. This will allow the addition of any number of new sensors into the system. In theory it would therefore allow us to add two M5s, one for each foot, into the model, to get a more complete representation of the skater's movements.

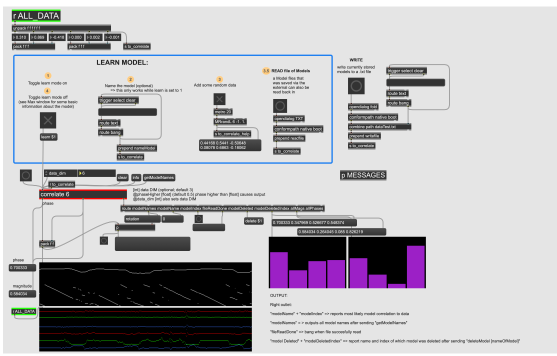

Figure 2. Max Patch showing *correlate* external and infrastructure

[5] `https://micropython.org/`

[6] `https://cycling74.github.io/min-devkit/`

The external is put in learn-mode in order to store a movement type/pattern to later match against the real-time data.

This tells the external that all data received until learn mode is disabled is part of an indicated movement pattern. Optionally, a name can be specified for easier model management. Models can be deleted, as well as written out to a text file, which in turn can be read back into the external. The data of the sensors have to be normalized. In this study, we implemented several normalization techniques to preprocess sensor data for correlation analysis. Each normalization method has its unique characteristics, advantages, and limitations, making them suitable for different types of sensor data and applications.

3.3 Normalization Techniques

L2 Normalization: L2 normalization scales each element of the vector by the L2 norm (Euclidean norm), ensuring that the vector's magnitude is 1.

$$x_i' = \frac{x_i}{\sqrt{\sum_{j=1}^{n} x_j^2}} \quad (9)$$

This method preserves the relationships between elements and ensures magnitude invariance, making it useful for integrating multiple sensor data with different units and scales. However, it is sensitive to outliers and may not be suitable for sparse data.

Z-score Normalization: Z-score normalization standardizes the data by subtracting the mean and dividing by the standard deviation. This method centers the data around zero and is useful for data with outliers, making it effective for integrating sensor data with varying distributions. It requires the calculation of mean and standard deviation and is sensitive to outliers.

$$x_i' = \frac{x_i - \mu}{\sigma} \quad (10)$$

where μ is the mean of the vector and σ is the standard deviation.

Min-Max Normalization: Min-Max normalization scales the data to a fixed range, typically [0, 1].

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (11)$$

where $\min(x)$ and $\max(x)$ are the minimum and maximum values of the vector, respectively.

This method preserves the relationships between data points and is easy to interpret, making it suitable for sensor data with known and consistent ranges. However, it is sensitive to outliers and may not handle data with different distributions well.

Max Normalization: Max normalization scales the data by dividing each element by the maximum value. This method is simple to implement and scales data relative to the maximum value, making it useful for sensor data where the maximum value is significant.

$$x_i' = \frac{x_i}{\max(x)} \quad (12)$$

where $\max(x)$ is the maximum value of the vector.

It is sensitive to outliers and does not center data around zero.

Decimal Scaling Normalization: Decimal scaling normalization scales the data by moving the decimal point of values.

$$x_i' = \frac{x_i}{10^j} \quad (13)$$

where j is the smallest integer such that $\max(|x_i'|) < 1$.

This method is simple to implement and preserves the distribution of data, making it suitable for sensor data with consistent ranges and no extreme outliers. However, it is less effective for data with large ranges and does not center data around zero.

Mean Normalization: Mean normalization scales the data by subtracting the mean and dividing by the range (max - min). This method centers the data around zero and reduces the impact of outliers, making it useful for integrating sensor data with different units and scales. However, it is sensitive to outliers and requires the calculation of mean and range.

$$x_i' = \frac{x_i - \mu}{\max(x) - \min(x)} \quad (14)$$

where μ is the mean of the vector, and $\min(x)$ and $\max(x)$ are the minimum and maximum values of the vector, respectively.

Robust Scaler (Median and IQR Normalization): Robust scaler normalization scales the data by subtracting the median and dividing by the interquartile range (IQR).

$$x_i' = \frac{x_i - \text{median}(x)}{\text{IQR}(x)} \quad (15)$$

where median(x) is the median of the vector and IQR(x) is the interquartile range (the difference between the 75th and 25th percentiles).

This method is robust to outliers and centers the data around zero, making it suitable for sensor data with outliers. It requires the calculation of median and IQR and may not be suitable for small datasets.

These normalization techniques provide a comprehensive set of tools for preprocessing sensor data, ensuring that the data is comparable and can be effectively used for correlation analysis and matching. Each method has its strengths and weaknesses, making it important to choose the appropriate normalization technique based on the specific characteristics of the sensor data and the requirements of the application. We are currently focused on testing the L2 normalization type, since we have a simple sensor setup that has, however, different unit scales.

3.4 Testing Setup

We are currently in the early stages of testing both the hardware and software implementations. Image 3 shows the placement of the M5 mounted on the boot of a skater. Since we are currently only testing with a single mounted unit, it is important to choose the foot to mount the unit on carefully based on the desired movements to track. After securing and turning the sensor on, it automatically connects to a predetermined network, and starts sending OSC messages at 50Hz. These messages are received, processed, and either recorded or used for tracking by a Max patch (see Section 3.2).

Figure 3. M5StickC Plus2 placement on Figure Skater boot

At these early stages we are only recording the data so it can be tested away from the rink in more controlled settings. We have yet to test real-time input from the M5 for tracking purposes at the rink.

These early tests have involved the recording of different basic skating skills and tricks, which were repeated several times. This ensures that we can both train the model on a specific type of movement but also test the tracking on variations of the movement, since no repetition of the same skill is exactly the same.

Since we have all the different versions of each skill recorded, we can test the model against itself - i.e. using the model also as tracking input to ensure the tracking is working by checking a 100% likelihood match. The synchronization patch explained in Section 3.2, was specifically designed to facilitate this kind of testing. It allows us to precisely select the regions of data to be used for a model, and then use the same selection for the tracking input. Subsequently, different regions with different versions of the same skill or regions featuring other skills can be used to check the stability and reliability of the tracking. Depending on the results of these tests the models might have to be adjusted to get a more reliable output.

4. EXPLORATION AND ARTISTIC DOCUMENTATION

The authors conducted a series of experiments involving motion tracking of figure skating to develop a proof of concept. This consisted of three separate on-ice recording sessions in which the movements of a figure skater were recorded as both data (accelerometer and gyroscope) and video. The first session involved testing the device for proof of concept. It was important to see if the device could be securely mounted on a figure skating boot, if the data could be transmitted over a wireless connection, and finally, if the transmitted data could be subsequently recorded and stored. The M5 was laced into a figure skating boot, which is currently secure enough for the skater to perform simple exercises such as turns, single jumps, and spins.

The second session involved more focused and planned movements. Prior to the recording, the authors made a list of important movements to track specifically to analyse the data. This involved all of the different jump types (Axel/waltz, salchow, toe loop, Rittberger/loop, flip, lutz), but also multiple spin positions (upright, camel, sit, layback), as well as skating skills and turns. The data transmitted from the M5 was recorded along with video footage of the skating performed during the recording.

The third and most recent test involved isolating and targeting specific movements to explore more specific attributes of the data. Following the data experiments and explorations resulting from the second session, the authors planned the third session to include deeper exploration into more limited movements. Three video and data streams were recorded. The first focused on Rittberger/loop jumps, a type of jump in figure skating that involves jumping off and landing on the same outside edge. The second focused on various types of spin positions. The third focused on double three-turns and twizzles, two different multi-directional skating skills.

In the first two videos, the skater first performed a series of pivots with slow rotation to establish a data example before recording. The data was then recorded and used to begin exploring recognition of specific movements as well as ways in which to focus the musical application.

4.1 Using motion tracking of a figure skater to create a musical composition: *Artistic Documentation*

The original intention of the test recordings was to determine if the software could recognize patterns and therefore identify specific movements in figure skating. However, we also sought to experiment with the raw data to determine what was potentially viable for musical purposes and live and interactive performance situations. To test this, we experimented with the data's capacity to influence MIDI messages to see if there was any inherent musicality in the raw data. We started with the material that was recorded during the second session that was played back in Max using the previously described patches (see Section 3.2). The data was then mapped to various MIDI and audio parameters.

Of all of the data transmitted by the M5, the gyroscope vector data was the most variant throughout the course of the recording. For this reason it was used for the first musical experiment, since the audio/visual link was very apparent. The gyroscope vector data was applied to affect pre-defined MIDI parameters such as pitch, velocity, and note duration. This was then sent to a Viscount Cantorum Duo [7], and the resulting playback was recorded as audio. Several variations of data manipulation were recorded. Clearly different musical results emerged from different figure skating movements even when only using the gyroscope vector data.

We then created several examples of what we call *artistic documentation* (see Section 4.2 and Section 4.3 below). We define *artistic documentation* as documentation that has been modified aesthetically to create a standalone artistic work. Therefore, it is a fusion between documenting the

[7] `https://www.viscountinstruments.com/instrument/cantorum-duo-plus/`

technical process with the corresponding musical results into one larger artistic expression.

4.2 Rittberger Etude (2025)

The first focused *artistic documentation* is in the form of a video piece titled *Rittberger Etude* [8] . It is named after one of the six types of figure skating jumps, the Rittberger or loop jump. This jump involves a figure skater taking off backwards on one foot and landing on the same foot after rotating in the air. For this research, we tracked single revolution loop jumps. This involved gathering data by recording several loop jumps in a row. To get contrasting data, we also recorded waltz jumps that involved a different rotation ($\frac{1}{2}$ rotation rather than full, and take off and landing on the opposite foot) as well as several slow non-jumping pivot rotations. This data was then used as training data to teach the software how to recognize three specific movements: a rotation, a waltz jump, and a loop jump/Rittberger. We also taught the software how to recognize a cascade of two loop jumps.

The resulting audiovisual work contains videos of the figure skating, the recall of the movement data, and of the data training process as an artistic aspect (see Section 4.1). Several musical layers were created using the accelerometer and gyroscope vector data similar to in the first work.

These layers were all continuous. The messages resulting from the trained data were used to trigger specific events in the piece, such as turning on various generated layers, adding audio processes, and adjusting the playback speed of the layers. A waltz, Rittberger, rotation, and Rittberger+Rittberger (two loop jumps in combination) all sent different messages to the music software, and each would trigger a different musical event. The form of this artistic documentation was therefore in part generated
through the form of the test video. Just as in the first video exploration, all of the video itself came from the initial skating videos as well as the data recordings. In this case, however, we took more liberty with artistically layering and transitioning between videos, since we were not trying to explore and demonstrate the same clear audiovisual link as in the first video. Just as in the first video, all sound was generated via MIDI messages on the Viscount Cantorum.

4.3 Spin Etude (2025)

Another important movement pattern in figure skating is spinning. Spinning generally takes place on one foot with both the body and the free leg in various positions. Because of the combination of rotational movement as well as the changes of position of the free leg (as well as which leg is the standing leg), this makes spinning an ideal movement to explore for getting a diverse data set within a singular movement.

The data, especially gyroscopic data, will vary depending on which leg the sensor is on. There was a large variation in data when the sensor was on the stationary spinning leg, compared to when the sensor was on the free leg. This means that a simple action like changing foot while spinning has a profound effect on the data output and in this case, the musical result. The data is also affected by the leg position when the sensor is attached to the free leg. A fully outstretched leg as in a camel spin produced significantly different data than that of a sit spin with the leg closer to the center of the body. This means that the data can be used to differentiate between different types of spins, and even different positions within a single spin.

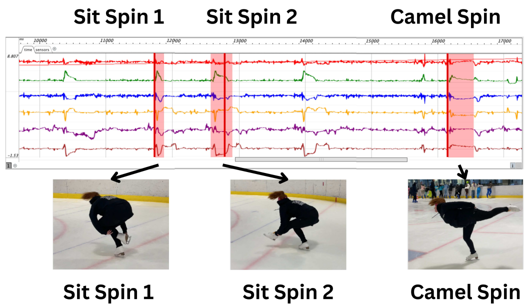

Figure 4. Shows how the data for 3 different types of spin positions changes. Top 3 data streams represents accelerometer data, bottom 3 data streams the gyroscope data

For this segment we also started the recording with slow pivot movement, but on the other leg, because this would be the leg most used for spinning (it is the forward spinning leg). Then we took data of spins with different body and leg positions. The sound was again all generated using the Viscount Cantorum, which received MIDI messages from Max based on the played back skating data. The gyroscope data is used again in combination with carefully constructed software to generate musical layers.

While previous tests used the Euclidean norm to calculate the magnitude of the gyroscope vector, here we used each axis of the vector individually. This allowed us to differentiate and sonify the movement data depending on leg angle and position as well as spinning foot [9] . The data was used to trigger different musical events, such as turning on various generated layers, adding audio processes, and adjusting the playback speed of the layers. The form of this artistic documentation was therefore in part generated through the form of the test video. Just as in the first video exploration, all of the video itself came from the initial skating videos as well as the data recordings. In this case, however, we took more liberty with artistically layering and transitioning between videos, since we were not trying to explore and demonstrate the same clear audiovisual link as in the first video. Just as in the first video, all sound was generated via MIDI messages on the Viscount Cantorum.

5. CONCLUSION AND FUTURE DIRECTIONS

The initial potential of G.L.I.D.E. for musicality and the use of the motion capture of figure skating movement using multiple sensors as a musically expressive tool was tested and evaluated in the early experimental stages of the project. We determined possible further explorations, both from a data tracking perspective as well as creative perspective. The system shows potential in early experiments,

[8] https://youtu.be/eOnfBccK7BE?si=qvArHUYK585vnlAB

[9] https://youtu.be/h2UBoPTfzwI?si=u7icxrTJuO9uganH

both from a technical perspective as well as for artistic application and musical expression.

Data acquisition needs to be refined with more focus on proper low-pass filter settings. Currently only by changing the Micropython code on the M5 can the filter be adjusted. Future implementations could use OSC messages sent to the M5 for making adjustments. The current version of the external does not allow switching between the different normalization types seamlessly; this needs to be addressed and tested. Specifically, we need to address the need to re-normalize any current model data stored in the external after a switch of normalization functions. Further research and development is needed to reliably pick between the most likely model candidate for the currently tracked gesture. The complexity of deciding which model is the likeliest candidate at any given time, taking previous model, current model, current phase of the model, current phase of likely model, etc., into consideration rises dramatically with the number of models to be tracked. Future iterations of the external also need to balance computational requirements as larger model counts, in combination with higher dimensional data structures, can get resource heavy. The current version of the patches as described in this paper run without lag or taxing the system on a 2023 M3 MacBookPro. However, the same setup on a 2020 2GHz Quad-Core Intel MacBookPro introduces major lag, making any real-time applications largely unusable.

Future developments will also include tracking both feet of a skater, experiments with tracking arms in isolation and in combination with feet tracking. Tracking multiple skaters will be explored for tracking pair skaters. Refining the tilt measurement of the tracking will be of the utmost importance to be able to easily distinguish between edges and edge changes. We will also have to track professional skaters that are able to jump double, triple, and quadruple jumps and that can spin at world-class levels to test the hardware. Extreme forces are created when performing at elite levels, forces we cannot simulate.

Several projects will be created as a means of realizing the potential of this motion tracking system. Exploration is needed to unlock different ways the movement could be mapped to different parameters and to what degree the movement is influencing the music. Future projects could experiment on a spectrum of musical reactivity. More reactive systems might involve a skater able to trigger single music changes through their movement. Adaptive systems would continuously respond to a figure skater's movements. At the highest level of interactivity, a figure skater would actively engage with the system, using their movement to intentionally affect musical parameters. Further developing the software and having the opportunity to track and collaborate with several skaters would be crucial in allowing for all of these possibilities.

Acknowledgments

- Zeitschleife — Verein für zeitgenössische Klangkunst [10]

[10] `https://www.zeitschleife-verlag.com/Verein`

6. REFERENCES

[1] A. Godbout and J. E. Boyd, "Corrective sonic feedback for speed skating: a case study," in *International Conference on Auditory Display*, Washington, DC, June 2010, pp. 23–30.

[2] J. E. Boyd and A. Godbout, "Multidimensional Synchronization for Rhythmic Sonification," in *International Conference on Auditory Display*, Atlanta, GA, June 2012, pp. 68–74.

[3] A. G. Godbout, C. Thornton, and J. E. Boyd, "Mobile synchronization for athletes: a case study in commercialization of sonification," in *International Conference on Auditory Display*, New York, NY, June 2014.

[4] M. Ritter and A. Aska, "Leap Motion As Expressive Gestural Interface," in *Proceedings of the International Computer Music Conference*.

[5] A. Aska and M. Ritter, "Approaches to Real Time Ambisonic Spatialization and Sound Diffusion using Motion Capture," in *International Computer Music Conference*.

[6] M. Ritter and A. Aska, "Extending the Piano through Spatial Transformation of Motion Capture Data," in *International Computer Music Conference*.

[7] M. Ritter, K. Hamel, and P. Bob, "Integrated Multimodal Score-Following Environment," in *International Computer Music Conference, ICMC 2013*, 2023.

[8] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," in *International Computer Music Conference*, pp. 33–40. [Online]. Available: https://hal.inria.fr/hal-00694803

[9] S. Lemouton and V. Nicolaou, "Polyphonic audio score following: The 'Otemo' case," publication Title: Composer in Research Project.

[10] J.-L. Giavitto and J. Echeveste, "Real-Time Matching of Antescofo Temporal Patterns," in *Proceedings of the 16th International Symposium on Principles and Practice of Declarative Programming*, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/2643135.2643158

[11] M. Ritter, "Playing the Field: A holistic approach to the analysis of electroacoustic music using higher order ambisonic technologies, computer vision, and the composer's collaborative environment," 2020. [Online]. Available: https://prism.ucalgary.ca/handle/1880/112917

[12] ——, "Ortus," 2014, series Title: Electronic Theses and Dissertations (ETDs) 2008+ Issue: December. [Online]. Available: https://open.library.ubc.ca/cIRcle/collections/24/items/1.0167636

[13] A. Aska, "The Woman and the Lyre," 2017. [Online]. Available: https://prism.ucalgary.ca/items/b28444c9-d644-4a48-b761-8e5835a08d0b

[14] M.-J. Yoo, J.-W. Beak, and I.-K. Lee, "Creating Musical Expression using Kinect," in *Proceedings of New Interfaces for Musical Expression*, pp. 324–325.

[15] A. Hadjakos, "Pianist motion capture with the Kinect depth camera," in *Proceedings of the International Conference on Sound and Music Computing*.

[16] M. Ciciliani, B. Lüneburg, and A. Pirchner, Eds., *Ludified: Artistic Research in Audiovisual Composition, Performance & Perception, Game Elements in Marko Ciciliani's Audiovisual Works*. The Green Box.

[17] R. B. Dannenberg and Z. Chi, "O2: Rethinking Open Sound Control," in *ICMC*, vol. 42, 2016, pp. 493–496.