

D3.3 Second Version of the Data Acquisition System Prototypes

Project acronym	TELMi
Grant Agreement nr	688269
Project title	Technology Enhanced Learning of Musical Instrument Performance
Call identifier	H2020-ICT-2015
Topic	ICT-20-2015
Type of action	Research and Innovation Action
Start date of project (dur.)	February 1st, 2016 (36 months)
Report availability	Pu-Public
Document due Date	September 30th, 2017
Actual date of delivery	September 30th, 2017
Leader	UPF (Internal review UNIGE)
Reply to	Alfonso Perez (alfonso.perez@upf.edu)
Additional main contributors (author's name / partner acr.)	Alfonso Pérez (UPF), Zacharias Vamvakousis (UPF), Sergio Giraldo (UPF), Oscar Mayor (UPF), Rafael Ramirez (UPF), Gualtiero Volpe (UNIGE)
Document reference	TELMi-D-WP3-UPF-20170930-D3.3-Second_Version_of the_Data_Acquisition_System_Prototypes
Deliverable type	DEM
Document status	Final



Executive Summary

This deliverable presents a technical description of the second version of the prototypes developed during the first half of the project. The deliverable is focused on the measuring systems (first version presented in D3.2) which are highly coupled with the feature extraction algorithms (D4.1 and D4.2) and feedback visualizations (D4.2).

To understand this document it is recommended to read the following deliverables.

Number	Title	Description
4.1	Early Stage Feature Definition and Computation Algorithms	Section 2, Audio Features and Section 3, Bowing Controls
4.2	Metrics for evaluation of skill performance and progress assessment	Section 2 and Section 3
3.1	Data Acquisition Plan and Setup	The whole document
3.2	First working prototypes of the measuring systems	The whole document



Table of contents

1 Introduction	5
2 Audio based prototypes	5
2.1 ViolinRT	5
2.1.1 Improvements with respect to the previous version	5
2.1.2 Improvements for future version	5
2.1.3 Screenshots and link to video showing the prototype	6
2.2 SoundsGood	7
2.2.1 Context and Use cases	7
2.2.2 Goals / Questions	7
2.2.3 Description	8
2.2.4 Technology used	8
2.2.5 Next steps	9
2.2.6 Demonstrations and related Publications	9
2.3 Indirect Acquisition - The Violin Palette Navigator	10
2.3.1 Context and Use cases	10
2.3.2 Goals / Questions	10
Figure 5. Violin timbre palette plane	10
2.3.3 Description	10
2.3.4 Technology used	11
2.3.5 Next steps	11
2.3.6 Links to Demonstrators	11
3. Prototypes based on High-end systems	11
3.1 Qualisys Cameras	11
3.2 Polhemus System	12
4. Prototypes based on Kinect	12
4.1 Context and Use cases	12
4.2 Goals / Questions	12
4.3 Description	13
4.4 Technology used	14
4.5 Next steps	16
5. Bio-signal based prototypes	16
5.1 Electromyogram (EMG)	16
5.1.1 Context and Use cases	16
5.1.2 Goals / Questions	16
5.1.3 Description	17
5.1.4 Technology used	18
5.1.5 Next steps	19
5.1.6 Demonstrator and related Publications	19



5.2 Electro-Encephalogram (EEG)	19
5.2.1 Context and Use cases	19
5.2.2 Goals / Questions	20
5.2.3 Description	20
5.2.4 Technology used	20
5.2.5 Next steps	20
6. References	21



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

1 Introduction

In the first 8 Months of the project we have been mainly developing the high-end (and alternative high-end) sensing systems (Qualisys and Polhemus based, D3.2) to set the ground truth for the measurements. In the period reported in this document (M8 to M18) we have been mainly working on the development of the low-cost measuring devices based on audio, kinect and bio-signals. In the document we report the main advances with these low-cost systems and indicate links to video using the developed demonstrators. Additionally, we report on the main improvements to the high-end sensing platforms.

These prototypes will be further developed and presented in future deliverables at the end of the project, *D3.4 Final platforms of the data acquisition systems (M33)*.

2 Audio based prototypes

Audio measurement and synchronization is carried out as explained in D3.2. However, many advances have been done in the prototypes (feature extraction and visualizations). Although this document focuses on the measurements, we think it is useful to include some details about the advances of the prototypes using audio.

2.1 ViolinRT

The ViolinRT prototype is used to record and analyse user performances, and present feedback in real-time. Audio feature descriptors are extracted and visualized in real-time. It also does performance analysis to evaluate the performance of the user in terms of tuning and timing.

Users perform a musical score presented on the screen and after playing it, they can review the performance and get feedback from the system to improve the performance. The prototype acquires audio from the violin using a microphone or pick-up and it can combine the audio features with other modalities acquired from a remote computer, e.g. gestures (from Kinect, Myo or Qualisys cameras) that are received through an OSC connection.

The prototype has been already presented in *D3.2 First working prototypes of the measuring systems*. Since then, during the following months of the project, the prototype functionality as well as the features and visualization feedback cues have been further extended as described in *D4.2 Metrics for evaluation of skill performance and progress assessment* to the point that now we have a mature prototype to be used by student groups.

2.1.1 Improvements with respect to the previous version

There are a few improvements and new features implemented in the current version compared with the previous prototype described in D3.2:

- Improved rating algorithms
- Possibility to load and save performances from/to persistent disk.
- Compare a user recording with the reference expert performance.
- Visualize real-time 3D mocap data and time series from other sensors (received via OSC).



2.1.2 Improvements for future version

Until the end of the project, the ViolinRT prototype will be continuously under development adding new features based on user feedback gathered in workshops and focus groups as well as feedback contributions from RCM students using the prototype. Here we enumerate some ongoing features that are expected to be included in next releases of the prototype:

- Upload a performance to the public database (repovizz)
- Retrieve a performance from the public database (repovizz)
- Practice loop mode
- Display intelligent visual summaries of a performance
- Improve score layout rendering
- Piano roll note zooming to view more analysis details

2.1.3 Screenshots and link to video showing the prototype

The following screenshots show the current version of the ViolinRT prototype. Figure 1 shows the score and piano roll widget where it is easy to appreciate the tuning deviations of the performer compared to the perfect execution of the score. Also the dynamic curve is shown on the lower part of the screen to be able to see the dynamics evolution of the of the performance in real-time. A global rating based on tuning and timing accuracy is computed during the performance and presented to the user as shown in the figure.

In figure 2 a score-based interface is shown providing timing and tuning feedback over the musical symbols with the aim of displaying the feedback in the same environment as musicians are used to practice. Once the user has finished a performance he/she can switch from one view to another and review the performance feedback after each practice session.

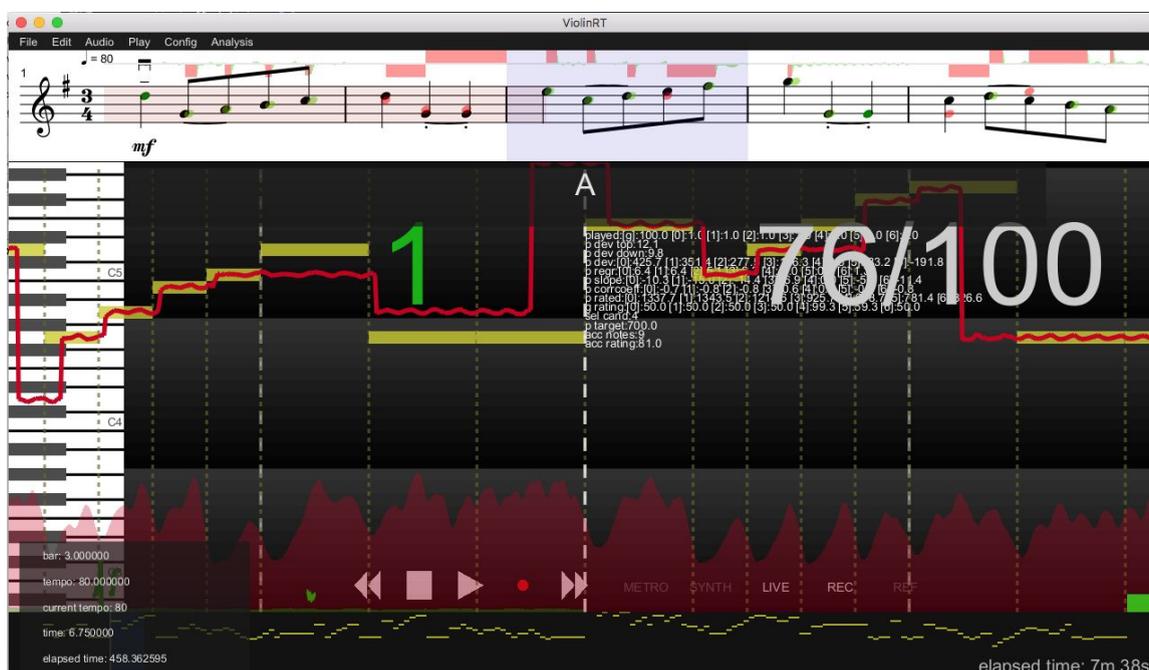


Figure 1: Screenshot of the ViolinRT intonation feedback on Piano Roll Mode



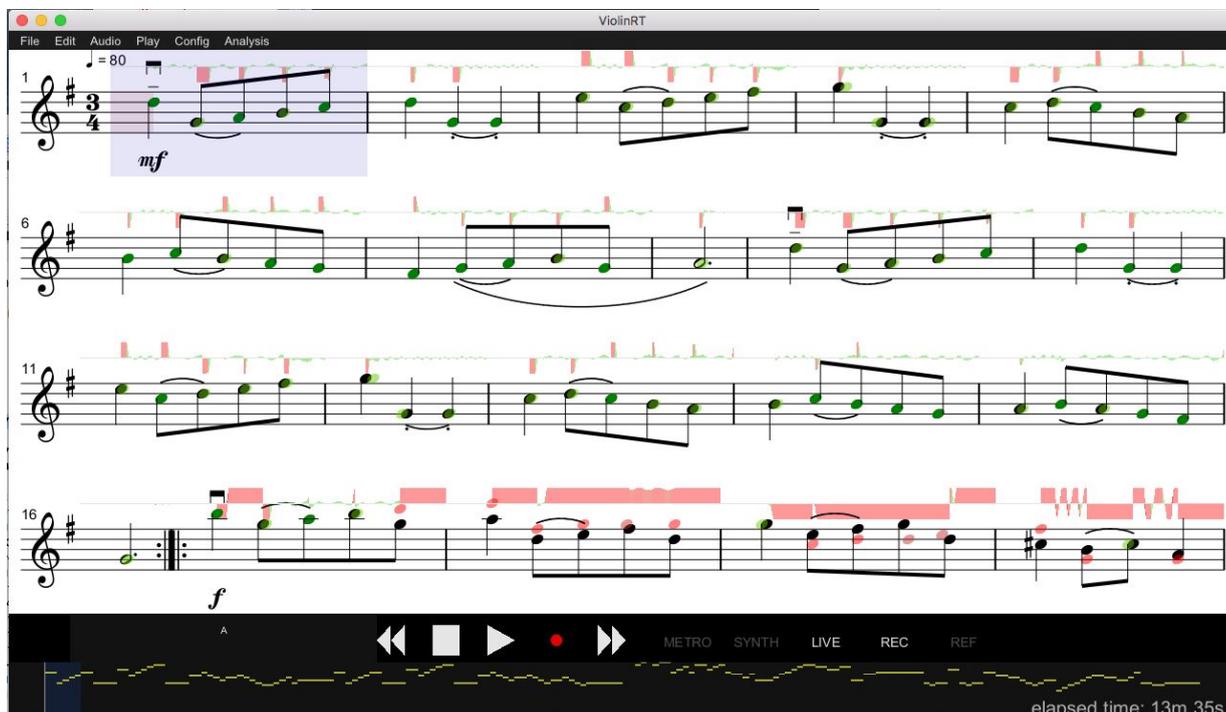


Figure 2: Screenshot of the ViolinRT intonation feedback in Score View Mode

The following videos show the prototype being used:

<https://vimeo.com/235749962>

<https://vimeo.com/235750072>

2.2 SoundsGood

2.2.1 Context and Use cases

The assessment of the sound properties, and concretely the tone quality, of a performed musical note is a relevant aspect in the performance of a musical instrument. However, there is not a formal definition of performance tone quality due to its subjectivity, although some consensus exist among musicians on what is a good or a bad musical tone. The SoundsGood widget is a computational approach for the automatic assessment of violin sound production which implements a real-time feedback system of several sound dimensions related to the multidimensional aspect comprising the timbre quality. For an overview of the sound dimensions considered and its calculation see D.4.2, Section 3.

2.2.2 Goals / Questions

The aim of the widget is to provide real-time feedback of several sound dimensions that comprise tone quality. Given the subjective nature of the tone, we have trained tone quality models using machine learning techniques to evaluate each of the defined tone dimensions of a note from the features obtained from the note audio. However, several aspects of tone production may vary across



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

performers, based on performer's technique, instrument, and/or criteria. Therefore, this raises the question whether a unique model for tone quality might suit a wide range of performers. Also how consistent are the performance aspects of tone quality among performers. Work is being carried out to investigate these issues.

2.2.3 Description

The “goodness” of the sound of a note or set of notes is represented by a spider chart where each vertex represents each dimension comprising tone quality. Dimensions consider specific aspects of sound such as pitch, spectral content, and dynamics (see section 3.7 of D 4.2), as well as several high-level definitions of tone quality, e.g dark/bright, cold/warmth, etc (see Table 2 on D 4.2.). The different dimensions are obtained from the real-time computation of audio features (see D 4.1 for an overview of the audio descriptors considered) and machine learning models previously trained based on human perceptual ratings (See D 3.7). In Figure 3 we present a screenshot of the tone quality prototype.

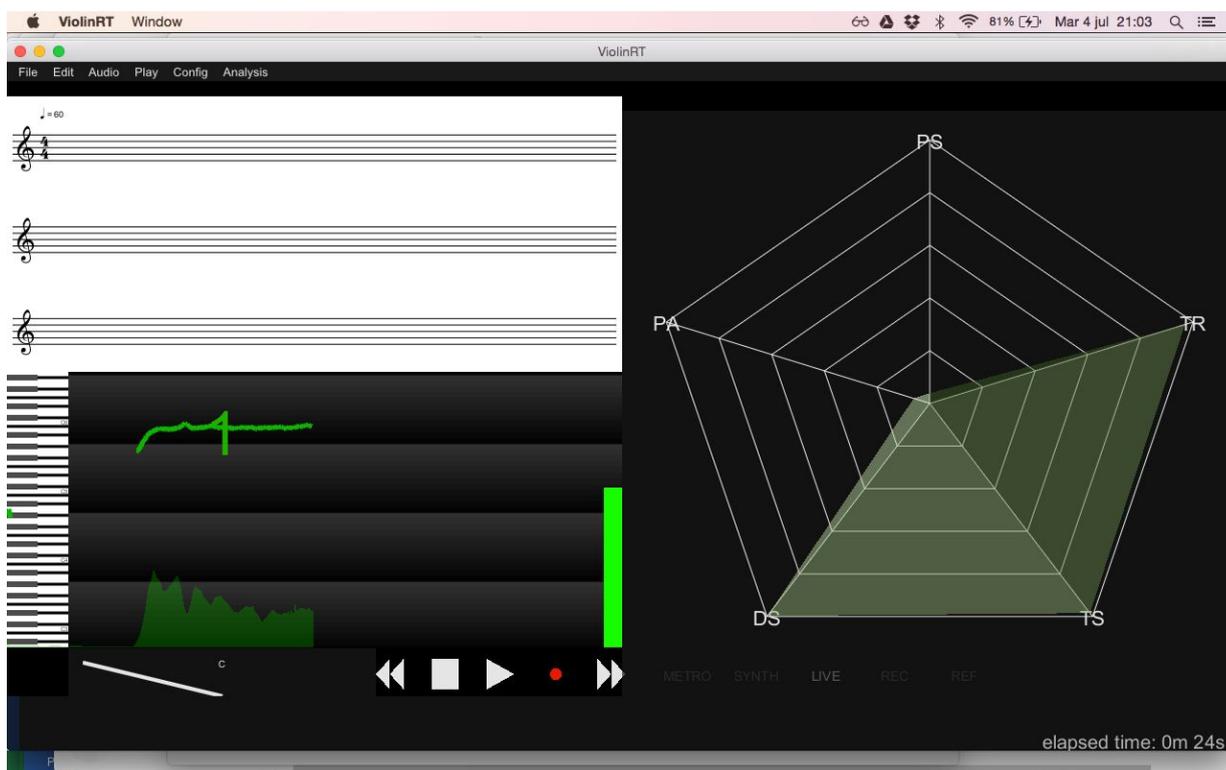


Figure 3. Screenshot of the SoundsGood widget. The system computes in real-time several low-level features from the audio signal. From these computations, a feature vector on a frame basis is obtained, and previously trained machine learning models are applied to obtain a value for each of the considered dimensions.

2.2.4 Technology used

The “SoundsGood” widget is implemented inside the Violin RT app, using the OpenGL C++ framework. The technology used within the app is of two types: one is used to extract features from audio and a second one is used to obtain the machine learning models. For audio feature extraction, as mentioned in D 4.1 we used the Essentia library developed at the Music Technology Group of the Pompeu Fabra



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

University (Bogdanov et al. 2015). For the Machine Learning modeling stage we used the WEKA Machine Learning library (Frank et al. 2016) to obtain off-line models. The obtained models are later implemented in C++ in the ViolinRT app. For the real-time application of the tone quality models the C++ “ml-pack” library by Curtin et al. (2013) is used.

2.2.5 Next steps

As mentioned in Section 2.2, the subjective nature of perception of tone quality entails several complications in terms of the validity of the output of the models, given the differences across users in terms of technique, instrument, and criteria. Therefore, the app will incorporate a default model which is the one obtained from the recordings of professional musicians, as well as the possibility for the user to add new recording examples for each tonal dimension, to be incorporated to the audio features data. A new model can be re-trained within the app with the user data or by merging new data with the default one. Also we plan to include the possibility for the user to define new tonal dimensions/recording data.

2.2.6 Demonstrations and related Publications

A public demonstration of the prototype took place at the 2nd public event of the TELMI project (Deliverable D7.4) within the International Symposium on Performance Science (ISPS 2017) on August of 2017, in Reykjavik, Iceland. Figure 4 shows the prototype being demonstrated at ISPS 2017.

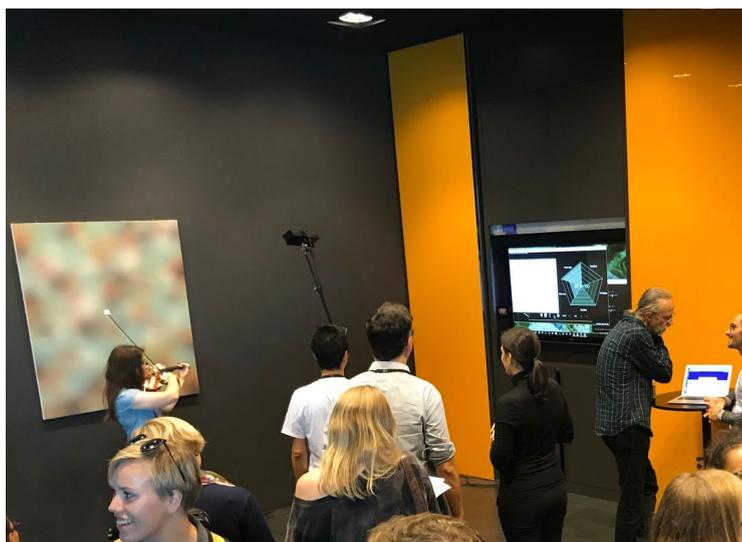


Figure 4. Demonstration of the SoundsGood prototype at ISPS 2017.

2.2.6.1. Links to Video, GitHub.

- [Link to ISPS video demos of SoundsGood prototype](#)
- Link to SoundsGood video demos:
 - [Video 1](#)
 - [Video 2](#)
 - [Video 3](#)

2.2.6.2. Publications.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

The framework proposed to assess tone quality used in the soundsGood widget has been published by Giraldo et al.(2017a) and Giraldo et al.(2017b). An extended publication for the Frontiers in Psychology Journal (special issue in Performance Science) is in preparation.

2.3 Indirect Acquisition - The Violin Palette Navigator

2.3.1 Context and Use cases

Timbre is a very complex concept, usually related to spectral content and variation of this content. In violin playing, timbre in sustained parts of the sound is mainly controlled with 3 bowing parameters, speed, pressure and point of contact (also called in the acoustic literature bowing velocity, bowing force and bow-bridge distance or *bbd*).

The idea of this demonstrator is to build a timbre navigator that we call *The Violin Palette Navigator*. The Navigator allows the user to self-monitor the coordinates of his/her bowing in a 3D space (force, speed and *bbd*).

2.3.2 Goals / Questions

The goal is to allow the performer to explore the timbre space and be able to discover and self-monitor his/her use. This is shown in Figure 5. Colors represent different timbre 'colors' while the lines represent the boundaries that delimit the space that produces sounds that are (traditionally speaking) considered to be good. Figure 5 shows a projection in 2D with axis *bbd* and force.

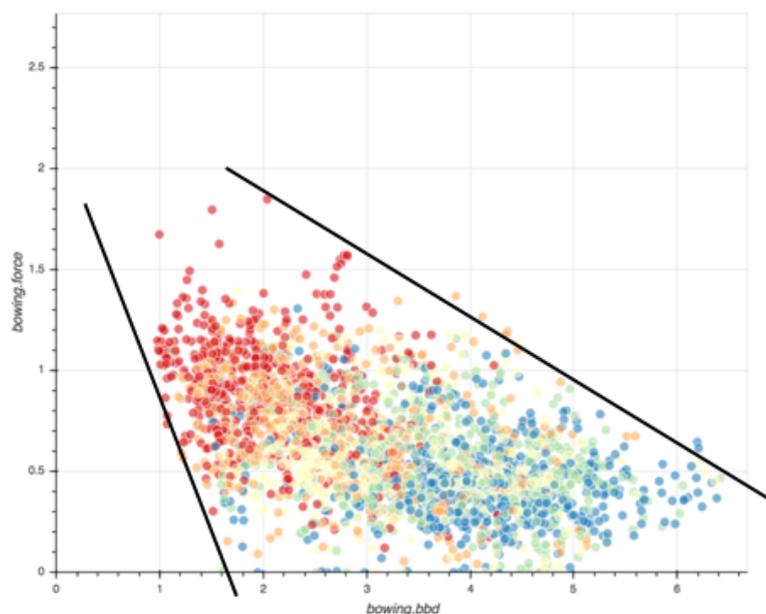


Figure 5. Violin timbre palette plane

2.3.3 Description

Such a timbre space, which has as axes the bowing parameters, could be built using Motion Capture sensing systems. However, in this demonstrator, bowing parameters are solely estimated from audio analysis. This is done by learning models with a database of multimodal data (bowing parameters +



audio features). Such an analysis allows for mapping audio features to bowing parameters and build generative models that can estimate bowing from audio analysis.

2.3.4 Technology used

The prototype has been implemented in Python (it will be ported to C++ and integrated into *violinRT* in the future). It uses *Essentia* library (UPF Technology) for audio feature computation as well as extra audio descriptors that are tailored for this prototype (see D4.3). Bowing estimation is based on machine learning models that are trained with a database of sync. Data with bowing plus audio. Several models have been tested using scikit.learn python package. Among them, the most successful are Support Vector Machines and Neural Networks.

2.3.5 Next steps

- For now, bowing velocity and bowing force are estimated with accuracy. We expect to have an accurate estimation of the point of contact or *bbd* in the future.
- Audio-based models will be combined with the models based on the kinect/webcam.
- The limits of maximum and minimum bowing force (as depicted in the Figure 5) have to be determined through a short calibration for every violin being tested.

2.3.6 Links to Demonstrators

<https://vimeo.com/235750264>

3. Prototypes based on High-end systems

3.1 Qualisys Cameras

The Qualisys measuring system was already fully functional in D3.2 (Section 4.1). However, several improvements have been carried out.

3.2.1 Improvements

- Physiological sensors, including the Myo sensors, have been fully integrated and their data streams synchronized in the recording platform.
- OSC communication modules for sending and receiving full motion capture data (e.g., the body skeleton) to/from the ViolinRT application have been developed and integrated in the platform.
- Interfaces have been developed/improved to enable quick and effective control of a recording session and to facilitate synchronized data playback.
- Algorithms to extract a broad collection of bow and body movement features were developed as EyesWeb modules and applications and integrated in the platform (see the features described in D4.1 and D4.2).
- Algorithms to assess the technical quality of a violin performance, based on the recorded data and features, are currently under development and are going to be progressively included in the platform..
- Research and design of musically and pedagogically effective visualizations of recorded data and extracted features is ongoing, e.g., how to visualize in a meaningful way the bow and body movement features extracted from the TELMI reference archive.

3.1.2 Links to Demos



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

<https://youtu.be/njlvdbPgsk>

3.2 Polhemus System

The Polhemus-based measuring solution was fully already functional in D3.2 (Section 4.2). Below we list the main improvements carried out on this system.

3.2.1 Improvements

- The developed software, beyond the local visualizations, is able to send bowing and violin motion data via OSC protocol messages to communicate with the main *violinRT* application that is responsible for the visualizations.
- Research and design of musically and pedagogically effective visualization of bowing parameters. Visualizations are done in *violinRT* independently on the measuring system, be Qualisys, Polhemus, Kinect or web-cam.
- A panel has been added to the graphical interface that allows an easier calibration of the bowing force.

3.2.2 Links to Demos.

<https://vimeo.com/184842708>

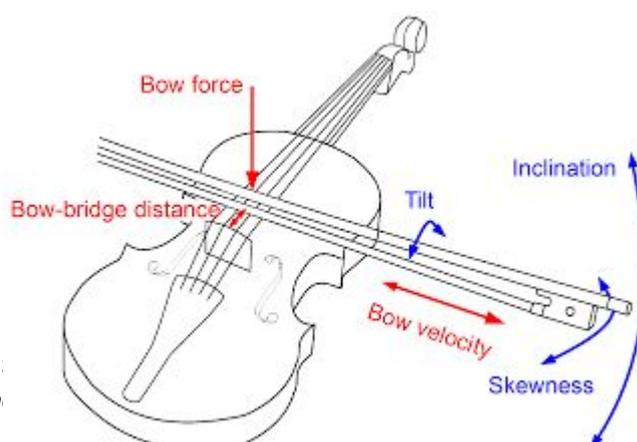
4. Prototypes based on Kinect

4.1 Context and Use cases

A low-cost motion capture system, based on Microsoft Kinect and reflective markers and capable of 6 degrees of freedom pose estimation of both the violin and the bow, has been developed. The low cost of Kinect, and the relatively high performance of the system, makes it an appropriate setup for both music schools and homes. The system requires placing 3 reflective markers on the violin, and 3 more on the bow. Initially, a calibration phase is required, lasting about 20 seconds. Our preliminary experience with the prototype indicates that the following features are robustly extracted with frame rate 30 fps: bow tilt, bow inclination, bow skewness, bow-bridge distance, bow-violin distance, and bow position. A speculation of bow force can also be made. In the current version of the prototype, all calculated features, along with the 3d position of all markers on the bow and violin, and the body skeleton acquired by the Kinect SDK, are sent to ViolinRT through the OSC protocol for recording or visualization purposes. We refer to program acquiring Kinect data and computing all features, as Kinect acquisition server. The acquisition server also provides the possibility of saving all recorded data of a session in a file.

4.2 Goals / Questions

The goal of this tracking system is to provide an easy to setup and low cost tracking system, that would make TELMI technology accessible to violin students for practice at home and music schools. This Kinect violin acquisition server sends to ViolinRT features related to the bow position in relation to the violin: bow tilt, bow inclination, bow skewness, bow-bridge distance, bow-violin distance, and bow position.



This project has received funding from research and innovation programme un

These features can be used to evaluate the performance of the students, or to give them real-time feedback.

4.3 Description

The first time the Kinect violin acquisition server is used, a calibration is required. First the user has to place the 6 reflective markers as shown in figure 6. As reflective markers we are currently using small pieces of reflective adhesive tape. A small stick is also placed at the top of the bow, allowing the placement of two markers in its edges. Alternatively, just one marker can be placed on the top of the bow. In that case though, the bow tilt is not computed.

When the server is executed, the user is asked if he/she wishes to calibrate. When the calibration is complete, calibration data are saved in a file. The next time the server is run, if the user chooses not to calibrate, these values are loaded. The calibration consists of three phases.

Calibration of bow markers

In that phase, the user is asked to place just the bow in front of the Kinect. The geometry of the markers on the bow is saved.

Calibration of violin markers

In that phase, the user is asked to place just the violin in front of the Kinect. The geometry of the markers on the violin is saved.

Calibration of the bridge and bow

In this phase, the user is asked to place the beginning of the bow hair (close to the frog) on the bridge of the violin. By projecting the bow line on the violin plane, we are able to compute the location of the bridge in relation to the violin markers. We also compute the beginning of the hair of the bow. This allows us to provide an accurate bow position value.

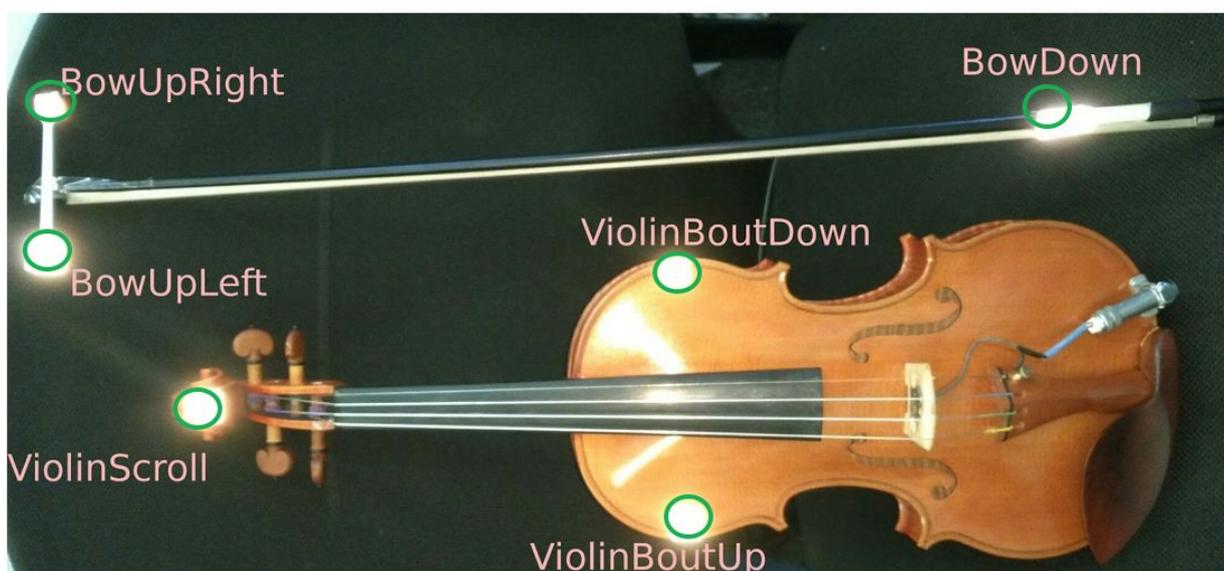


Figure 6. Reflective markers positions



After calibrating, or loading an existing calibration, the server is sending all the computed features, along with the 3d location of all markers and the skeleton of the violinist, to ViolinRT. In figures 7 and 8 two available visualizations are shown: top view and player's view, respectively. Such visualizations are expected to enhance the learning process of violin students. Video demonstrations can be found online at

<https://drive.google.com/open?id=0Bx3BxXLYoDUTRk5BN2U5Y2JZT3M>
<https://drive.google.com/open?id=0Bx3BxXLYoDUTbGptSEpURy15dDQ>
<https://drive.google.com/open?id=0Bx3BxXLYoDUTQ25CVUYtVVBza3c>
<https://drive.google.com/open?id=0Bx3BxXLYoDUTNEJodHBXaFd4SU0>

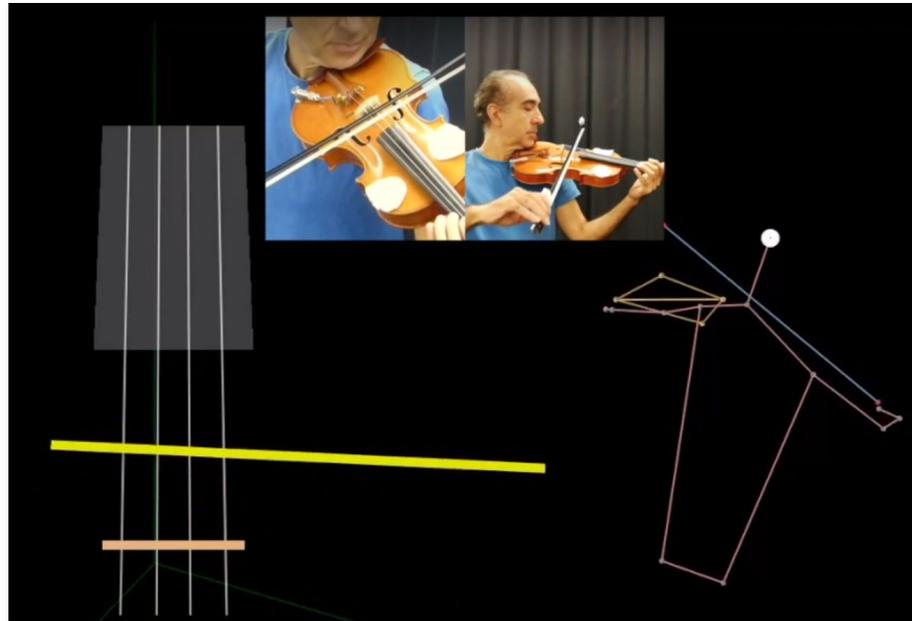


Figure 7. Visualization in ViolinRT from top view. Skewness and bow-bridge distance are visible.

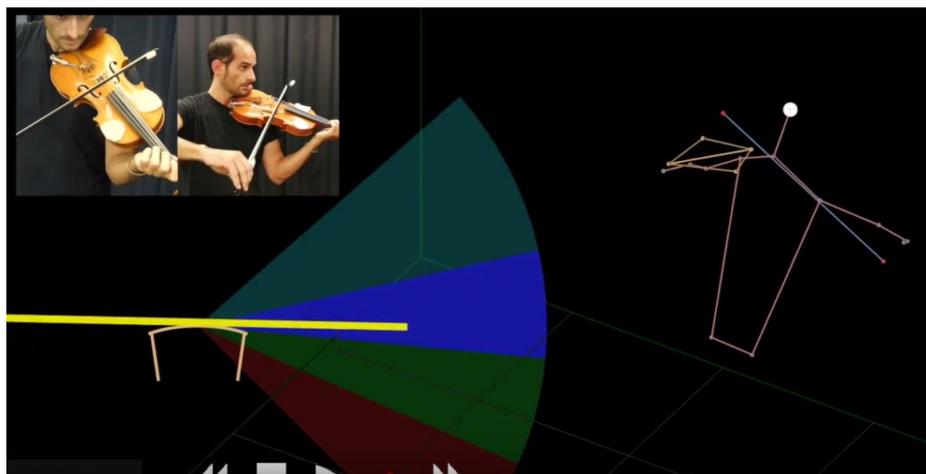


Figure 8. Visualization in ViolinRT from player's view.



4.4 Technology used

Three markers on the body of the violin, and 3 on the bow are being placed in the locations shown in figure 6. As markers we currently use reflective adhesive tape.

The OpenCV library is utilised for detecting the infrared markers in the infrared image. Initially, a high-pass filter is applied, and the infrared video frame (we will refer to it as image) is transformed to a binary image. A dilate function is then applied for increasing the size of the marker area to a minimum of 3x3 pixels. On this binary image, the reflective markers are visible as white spots in a black background. The findContours() function, based on the opencv library is then applied for detecting all markers in the infrared image. This function is based on an algorithm proposed by Suzuki and Satoshi (1985). If more than 6 markers are detected, something that happens due to additional reflective areas on the image, only the 6 markers with the minimum distance to a marker in the previous frame are considered. This way, noise introduced by reflection is disregarded.

3d position marker acquisition: Kinect does not provide depth data on the location of the reflective markers. Nevertheless, we observed in the depth stream, that around each reflective marker, there is a thin area of valid depth data. In order to acquire the depth of a marker, we had to look for the position of a virtual marker, in a 3x3 pixels area, 4 pixels close to the marker. What we are interested in is the distances between pairs of markers. The direction in which we move in order to measure these 3x3 area next to the marker is indicated by the line connecting each pair of markers. In order to label each marker to its valid location (violinScroll, violinBoutDown, violinBoutUp, BowDown, BowUpLeft, BowUpRight), the 15 distances in real world coordinates between all 6 markers are being measured. Out of these 15 distances, we are then looking for the 6 distances that fit the expected dimensions of two triangles: one formed by the 3 markers on the violin, and one more formed by the 3 markers on the bow. Once we know the 3d position in mm of the violin and the bow, we are able to compute all desired features (Figure 9), which include:

- bow tilt (in degrees ~ -40° to 40°)
- bow inclination (in degrees ~ -30° to 30°)
- bow skewness (in degrees ~ 75° to 105°)
- bow-bridge distance (in cm)
- bow-violin distance (in cm)
- bow position (in cm)
- bow velocity (in cm per second)
- an estimation of bow force

All computed features, along with the body skeleton, as acquired by the Kinect SDK, are streamed to ViolinRT through the OSC protocol.



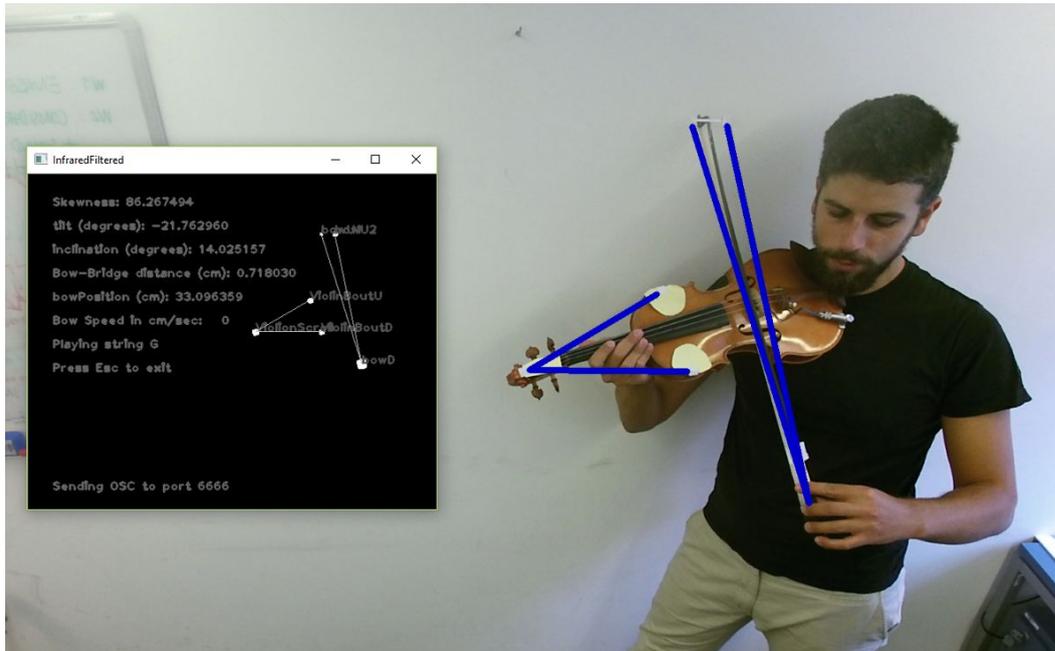


Figure 9. Six Degrees of Freedom Pose estimation of the Bow and the Violin using Kinect and reflective Markers. Markers positions and computed features are sent through OSC protocol to ViolinRT.

4.5 Next steps

The performance of the system remains to be compared with high-end systems, such as Polhemus. Although fully operative, the current prototype has some issues that require improvement. Firstly, sometimes marker tracking is lost because of possible reflections close to the reflective markers on the violin body. We are currently improving our tracking algorithms in order to provide more stable tracking. Secondly, kinect depth data are noisy. There is a jitter of around 1 cm on the depth data of each pixel. The current estimated latency in the tracking is around 45 ms. Although a moving average removes the noise, it increases the latency. We are working on the development of a model that could smooth the noisy depth data of kinect, without a compromise in the responsiveness of the system.

In parallel we have been trying to implement a marker-free version of the prototype. A six degrees of freedom pose estimation of the violin could be possible. Unfortunately, the fact that the bow is a thin object, makes it partially invisible in the depth stream of Kinect. Combining depth with RGB data might make possible tracking the bow in 3-dimensions. We are currently exploring such an alternative.

Studying how and whether different types of real-time visualization improve the learning process of violin techniques is also in our scope of interest.

5. Bio-signal based prototypes

5.1 Electromyogram (EMG)

The aim of this section is to apply low-cost sensor technology to the automatic detection and analysis of gestures related to music performance. With this aim, we measure both right hand (i.e. bow) and left hand (i.e. fingering) gestures. Our aim is to detect bow velocity and fingering by using off-the-shelf EMG



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

and motion capture devices such as the Myo and Riot sensors positioned in the forearms of the musician. Electromyogram activity helps to estimate finger-gesturing and force applied during musical pieces interpretation. We used the Myo device which in addition to electromyograms, provides magnetometer, accelerometer and gyroscope sensors.

5.1.1 Context and Use cases

The prototype provides a real-time accurate position and velocity estimation of the musician's forearm (Figure 10). It gives as well, information about muscle activity used to track finger-gestures and bow gestures with the help of machine learning algorithms.

5.1.2 Goals / Questions

- Calculate bow velocity.
- Estimate optimal finger-gesture performance among different users.
- Force estimation.
- Provide an easy to use recording data system that uploads files to *repoVizz* automatically.
- Design a gesture based violin-extension.

5.1.3 Description

The prototype transmits the 3D positioning data and the muscle activity data through Bluetooth into a custom C++ application which sends the filtered data to a Max application. EMG and orientation data is then processed using Hidden Markovian Models to give as an output forearm velocity, force, finger-gesture and bow-gesture detection.

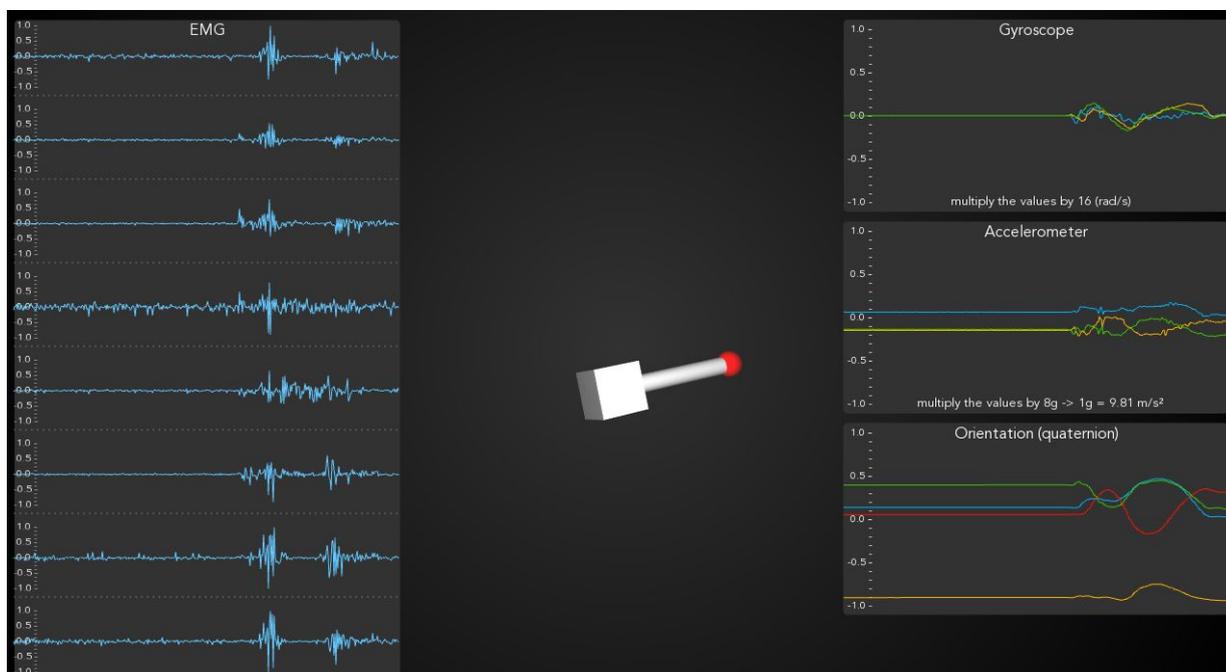


Figure 10. Image from the custom c++ application that reads Myo signals in real-time.



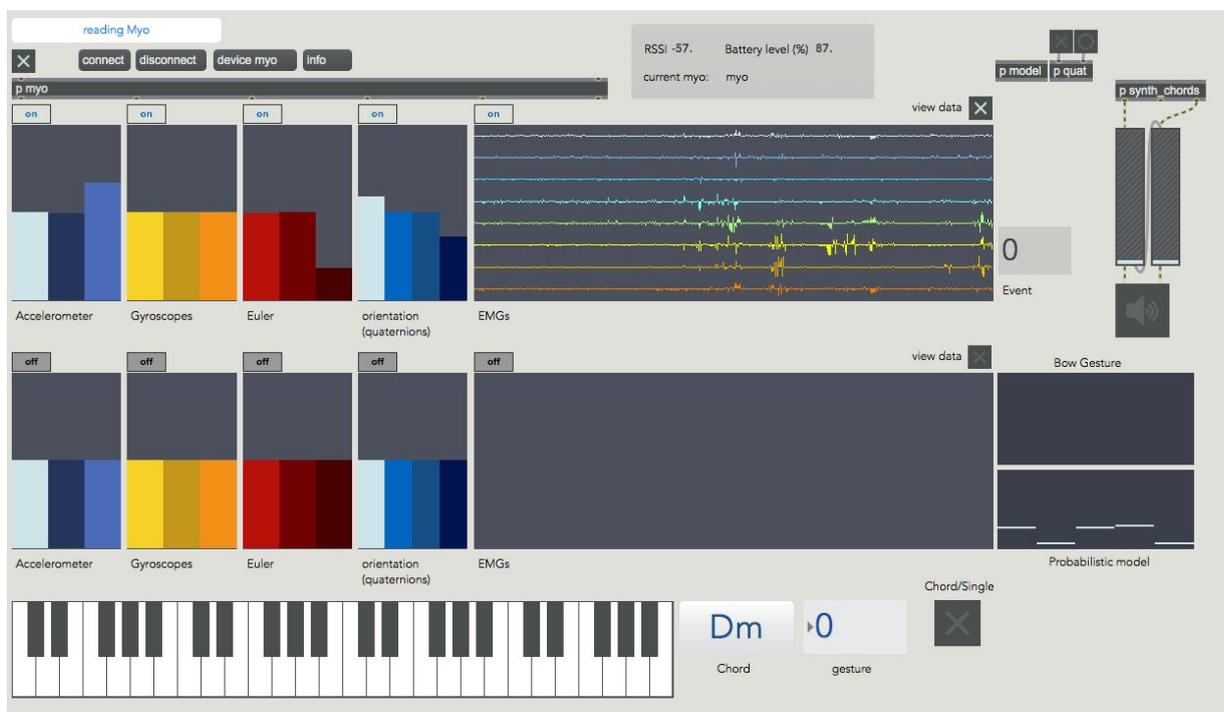


Figure 11. MYO-based gesture capture application

5.1.4 Technology used

The technology used is mainly based on two bio-signal sensors, Myo (Figure 12) and R-IoT (Figure 13), whose signals are then processed using C++, Max 7 and Machine Learning algorithms available on frameworks such as Max 7, Weka and Wekinator.



Figure 12. Image of Myo taken from the web <https://www.myo.com>

Myo Device is composed by:



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

Sensors	Medical Grade Stainless Steel EMG sensors, Highly sensitive nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer
LEDs	Dual Indicator LEDs
Processor	ARM Cortex M4 Processor
Haptic Feedback	Short, Medium, Long Vibrations

The R-IoT Sensor (Figure 13) is developed by Ircam and Bitalino. The R-IoT module allows for sensing movement, processing and wireless transmission through WiFi. It is developed as an essential brick towards creating novel motion-based musical instruments and body-based media interaction. The R-IoT sensor module embeds a [ST Microelectronics](#) 9 axis sensor with 3 accelerometers, 3 gyroscopes and 3 magnetometers, all 16 bit. The data are sent wirelessly (WiFi) using the OSC protocol.



Figure 13. Image of the R-IoT sensor taken from the web: <http://ismm.ircam.fr/riot/>

5.1.5 Next steps

Future work is focused on two areas: a) Gamification of an interactive musical tool for novice musical learners b) An online probabilistic measurement model to estimate finger disposition on a musical piece based on EMG sensor information, hence, giving feedback in real-time to self-practice learners. This system constitutes part of the main idea of developing a computer-based motion self-learning assistant. Tracking both left and right-hand muscular activity gives insights on how performers are executing fine gestural movements. Having precise models of "good practices" may provide useful real-time feedback and haptic feedback as well to learners on their deviation from a target performance.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

5.1.6 Demonstrator and related Publications

Paper “Air Violin : A Machine Learning approach to Fingering Gesture Recognition”. Is under publication, in Proceedings of 1st ACM SIG-CHI International Workshop on Multimodal Interaction for Education (MIE'17).

<https://vimeo.com/235760220>

<https://vimeo.com/235759761>

5.2 Electro-Encephalogram (EEG)

The aim of this section is to describe the capture of EEG data in order to study the cognitive and physiological processes involved in music learning and performance by finding correlations between student results and recorded EEG.

5.2.1 Context and Use cases

When we evaluate the efficiency and usefulness of a certain new implemented feedback technology, it is important not only to study the degrees of improvement between groups of students learning with different versions and interfaces (including a control group learning without the technology), but also to have information related to the mental state of the user in order to understand which factors may be affecting the results.

5.2.2 Goals / Questions

- To find EEG biomarkers of musical learning.
- To study the role of emotions in music learning
- To have a better understanding of the effects that different technologies can produce on the cognitive load levels of the users.
- To study through the use of Event Related Potentials (ERP) the effects of bad intonation in violin.

5.2.3 Description

The following description is related to the Emotiv EPOC EEG headset (Figure 14). Description for other EEG headsets is similar.

Headset Setup. A Headset Setup panel is used to calibrate the signal quality of the EEG headset. The main function of this panel is to display contact quality feedback for the neuroheadset's EEG sensors and provide guidance to the user in fitting the neuroheadset correctly. It is extremely important for the user to achieve the best possible contact quality before proceeding to the recording data. Poor contact quality will result in poor detection results.

Before putting on the neuroheadset, ensure that each of the 16 electrode recesses are fitted with a moist felt pad. If the pads are not already moist, wet them with saline solution before inserting into the headset, or, alternatively, use a medicine dropper to carefully moisten the pads while already in place.

After verifying that the Wireless Signal reception is correct, put on the neuroheadset by gently pulling apart the headband and lowering the sensor arms onto your head from the top down, near the rear of the skull. Next, slide the headset forward until the sensors closest to the headset pivot points are



located directly above your ears and as close to your hairline as possible. Adjust the fit so that the rectangular compartments at the front ends of the headband sit comfortably just above and behind your ears. Tilt the headset so that the electrodes are positioned in the correct place (Figure 14).



Figure 14. Emotiv EPOC EEG headset consisting of and correct position of the electrodes within the 20/20 system.

5.2.4 Technology used

The Emotiv EPOC EEG system consists of 16 wet saline electrodes, providing 14 EEG channels, and a wireless amplifier. Specifications can be found in Figure 15. The electrodes were located at the positions AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 according to the international 10–20 system. The data was digitized using the embedded 16-bit ADC with 128 Hz sampling frequency per channel and sent to the computer via Bluetooth. The impedance of the electrode contact to the scalp was visually monitored using the Emotiv Control Panel software. We collect the EEG data using the OpenVIBE platform. The data is processed in EEGLAB under the Matlab environment. Power spectral density (PSD) is computed from the right (F4) and left (F3) mid-frontal lobes using Welch's overlapped segment averaging estimator. The powers of four frequency bands were extracted corresponding to Theta (4-8Hz), Alpha (8-13Hz), Beta (13-24Hz) and Gamma (30-50Hz).

For future purposes we also dispose of the Smart BCI system. It consists of 24 Ag/AgCl electrodes located at the positions Fp1, Fp2, Fpz, F9, F10, F3, Fz, F4, C3, Cz, C4, P3, Pz, P4, O1, Oz, O2, T7, T9, T8, T6, A1, A2 and the reference. The data is digitized using the embedded 24-bit ADC with 250 Hz sampling frequency per channel and sent to the computer via Bluetooth. It is possible to collect the data using the OpenVIBE platform.



EEG HEADSET	
Number of channels	14 (plus CMS/DRL references, P3/P4 locations)
Channel names (International 10-20 locations)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
Sampling method	Sequential sampling. Single ADC
Sampling rate	128 SPS (2048 Hz internal)
Resolution	14 bits 1 LSB = 0.51 μ V (16 bit ADC, 2 bits instrumental noise floor discarded)
Bandwidth	0.2 - 45Hz, digital notch filters at 50Hz and 60Hz
Filtering	Built in digital 5th order Sinc filter
Dynamic range (input referred)	8400 μ V (pp)
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4GHz band
Power	LiPoly
Battery life (typical)	12 hours
Impedance Measurement	Real-time contact quality using patented system

Figure 15. Emotiv EPOC headset specifications

5.2.5 Next steps

Future work includes:

- To investigate the effect of feedback on intonation in violin from an electrophysiological perspective through the study of ERPs to assess the level of awareness of students.
- To investigate the role of feedback on motivation by comparing both behavioral and electrophysiological variables.
- To investigate the role of cognitive load in music learning by linking behavior with electrophysiological signals.

6. References

Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Curtin, Ryan R., Cline, James R. & Slagle, Neil P. and (2013). MLPACK: A Scalable C++ Machine Learning Library. Journal of Machine Learning Research, 14, 801-805.

Bogdanov, D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., Roma G., Salamon J., Zapata J. R., & Serra X. (2013). ESSENTIA: an Audio Analysis Library for Music Information Retrieval. 14th International Society for Music Information Retrieval Conference (ISMIR 2013).



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688269

Giraldo S., Ramirez R., Waddell G., Williamon A. (2017) A Computational Approach for Measuring Performance Quality in Violin Tones. In proc. of the International Symposium on Performance Science (ISPS 2017), August 29, Reykjavik, Island.

Giraldo S., Ramirez R., Waddell G., Williamon A. (2017) A Real-time Feedback Learning Tool to Vizualize Sound Quality in Violin Performances. In proc. of the 10th International Workshop on Machine Learning and Music (MML 2017), October 6, Barcelona, Spain.

Suzuki, Satoshi. "Topological structural analysis of digitized binary images by border following." Computer vision, graphics, and image processing 30.1 (1985): 32-46.

