PHENICX

D4.4. Methods for recognising performer's gestures from visual live recording data

Grant Agreement nr	601166
Project title	Performances as Highly Enriched aNd Interactive Concert eXperiences
Project acronym	PHENICX
Start date of project (dur.)	Feb 1st, 2013 (3 years)
Document reference	PHENICX-WD-WP4-UPF-140715-Methods_for_recognizing_performer's_gestures_from_visual_live_recording_data_v1.1
Report availability	Re - Restricted
Document due Date	July 31st 2014
Actual date of delivery	July 30th 2014
Leader	UPF
Reply to	Carles F. Julià (carles.fernandez@upf.edu)
Additional main contributors (authors name / partner acr.)	Álvaro Sarasúa (ESMUC) Sergi Jordà (UPF) Enric Guaus (ESMUC)
Document status	Final

Project funded by ICT-7th Framework Program from the European Commission

PHENICX

Table of Contents

EXECUTIVE SUMMARY

This Deliverable (D4.4) “Methods for recognizing performer’s gestures from visual live recording data” presents the results of task T4.2 “Methods for recognising performer’s and conductor’s gesture”.

It deals with the recording and analysis of motion capture data in classical music performances. Provided that the tasks described in this document are related to task T6.4 “Interactive systems for the performer impersonation”, it also explains the strategy that has been followed in order to maximize the relation and synergies between both tasks.

In the Background section, we put this work in the context of the current state of the PHENICX project and define the concrete goals of this document. In Section 1, we do an overview of the steps we have followed and explain how they relate to the rest of the structure of this document. Then, in Section 2, we present a brief review of existing works that relate to ours. We continue, following a chronological order, by detailing the different tasks conducted so far. In Sections 3 and 4 we explain why we chose **Microsoft Kinect as the device for recording motion capture data** in our case and the tests we did with different musicians, after which we decided to put our focus on the **conductor**. We recorded different performances as described in Section 5, as well as a group of subjects as if they were conducting on top of some musical excerpts (Section 6), automatically finding **different patterns and styles that could be exploited** in interactive systems. In addition, in order to address automatic symbolic gesture identification, we made some recordings on real conducting teaching lessons and performed automatic gesture identification as described in Section 7. The current implementation is **able to identify different conducting gestures** in real time.

As a conclusion, we have better defined the scope of our work to focus on **conducting movements**. We carried recordings on live performances, rehearsals and lessons and made them available for the consortium. We also studied the movement of subjects with different musical expertise while conducting to explore ways to identify how they embody the expressive aspects of the performance. Finally, we have developed a framework to recognize symbolic gestures in real time.

BACKGROUND

Deliverable D4.4 “Methods for recognizing performer’s gestures from visual live recording data” presents the results of task T4.2 “Methods for recognising performer’s and conductor’s gestures”.

The main goal of this task and deliverable is to provide the necessary technology, methods and knowledge needed to perform task T6.4 “Interactive systems for the performer impersonation”; so its content is fully defined by the goal and needs of T6.4.

T6.4 goal (also placed in D6.7) was described as “[...] to allow offline spectators to recreate the experience of performing the concert, as if they were the musicians. [...] a computer program will allow the spectator to control the audio generated by a particular instrument of the ensemble”. The activities to be addressed by D4.4 would be:

- Test and define the consumer level gesture capturing technology to be used in T6.4.
- Test and define the particular more suitable performers to be recorded and later impersonated by users
- Record data from live concert and controlled environment performances.
- Develop strategies to make gesture recordings compatible with Repovizz format, in accordance with the technical requirements described in deliverable D2.1 “Technical requirements document”.
- Develop methods to extract useful information from the body movement.

Many of these activities have an impact on T6.4 problem space.

1 INTRODUCTION

In order to create the proposed methods for gesture recognition and analysis, we had to analyze the possible sensors to be used, their body parts tracking performance with different types of performers (e.g. string player, conductor, woodwind/brass player...). In Section 2 we make a brief review of works in the field to introduce the aim of our approach. Section 3 covers the analysis of the sensor devices to be used to capture the body movement of both performers and users, concluding that the best option is Microsoft Kinect. Section 4 extends this analysis to the performer roles, by testing the device with several instrument players and conductor, and finds out that the occlusion of the instruments makes the recognition of performers difficult and inaccurate. Because of that, we argue that the best option is to focus on the conductor.

The special role of the conductor in classical music has an impact on how his gestures relate to the music and how users perceive this relationship, and these questions are very relevant to our goal of creating a game-like experience which really allows users to control the performance in an expressive and natural way by impersonating the conductor. Section 5 explains the recordings we have taken from professional conductors while performing, that will allow us to understand the role of the gestures in the context of a concert.

Section 6 explores how users express themselves moving their body while pretending to be a conductor and present the results of a study where different subjects were asked to conduct on top of musical excerpts. With this information we found some hints on ways to identify different conducting styles.

Section 7 deals with the formal part of the conductor gestures and addresses symbolic gesture identification. It argues that most symbolic gestures are primarily used in teaching and rehearsal situation, for which we performed recordings of conducting lessons, and then developed the algorithms to identify a set of main symbolic gestures.

Finally, Section 8 presents the conclusions of the aforementioned work and discusses the next steps to be done.

1.1 Convention

We use the following writing conventions:

- underlined for cross-references and references to other documents
- `verbatim` for software entities
- **bold** for emphasis

2 SCIENTIFIC BACKGROUND

Capturing the gestures of performers and conductors while playing and analyzing them are not new problems. Performers movements have been analyzed to identify the style (Godøy and Jensenius, 2009) , to extract expressivity (Dahl and Friberg, 2007; Wanderley et al., 2005), as well as conductors movements have been used to extract the tempo and rhythm (Bergen, 2012).

Our final goal, however, of creating a consumer-level game-like experience poses some restrictions: conductor gestures recognition is not intended for recognition of real gestures from a conductor, but rather for evaluation of the validity of the gesture by a nonprofessional player. Many previous approaches to the conductor gesture recognition used methods that assume that all the gestures are valid examples, such as Machine learning oriented methods, for instance (Kolesnik and Wanderley, 2004). Those methods can't be used to evaluate the formal validity of a symbol or gesture, something that would be desirable in our game-oriented experience.

Another restriction of the task is the need of identifying gestures and behaviours that are linked to specific musical expressivity components for non-musicians. Similar kind of studies were previously conducted for air instrument performance, free dance to music, or sound-tracing (Jensenius, 2007), but there are no studies on how to identify and exploit subject-specific behaviours.

3 DEVICES AND HARDWARE

In PHENICX, we want to record gestural data from live performances in classical music. For that, we have to use motion capture methods that are completely unobtrusive for performers, as well as easy and fast to set up. This imposes certain constraints we have to deal with, such as avoiding wearable sensors or modified instruments. For this reason, we decided to use contactless, unobtrusive and commercially available vision-based solutions. Furthermore, the devices used to capture performers data should ideally be the same of the final user's in the game-like experience. With this in mind, depth-sensing cameras such as Kinect (Microsoft Kinect) are a good trade-off in the sense that they provide motion capture tracking with a very easy setup that is completely transparent to performers, as it is equivalent to placing a regular camera or microphone on the stage, while they are widespread and available to users.

At the beginning of this project, the OpenNI (Open Natural Interaction) framework, which provides open source APIs to access Kinect and similar devices, was freely available. So was the NITE middleware by PrimeSense, to perform skeleton tracking we needed from such devices. During the time of this project, PrimeSense was acquired by Apple and the OpenNI website was shut down in April 23, 2014. However, documentation and binaries are still available in websites such as structure.io and developing tools with existing versions up to that date is still possible.

4 MOTION CAPTURE TESTING

4.1 Tests with instrumentalists

In a preliminary set of experiments, we performed recordings of different musicians in ESMUC to test the robustness of motion tracking using Kinect for each case. Also, these experiments allowed us to check the conversion from the recorded motion capture data to a Repovizz-compatible format.

The instruments with which we made this preliminary experiments were trombone (brass), violin (bowed string), bass drum and vibraphone (percussion). From the initial observations we could do during the recordings, we observed the following issues:

- Kinect is not robust at all for gesture tracking of **trombone** (and, in general, brass instruments). Kinect's skeleton tracking constantly confuses the instrument with the arms regardless of the position of the performer.
- Gesture tracking works reasonably well for **violin** when **standing up**. There are moments, however, in which Kinect's skeleton gets confused with the bow. When the violinist is sitting (which is the case for orchestral classical music concerts), the motion tracking becomes more problematic.
- For **bass drum** and **vibraphone**, percussion mallets are captured as part of the arms, with the hand being captured as the joint corresponding to the elbow (thus losing the actual position of the elbow).

In summary, the conclusion was that depth-sense cameras are not suitable to perform motion tracking from instrumentalists in live performance scenarios.

Figure 1: Snapshot from test with percussionist.

4.2 Tests with conductors

Although Kinect's motion tracking is meant to work with body motion for subjects that are not holding any big object (as in the case of instrumentalists), we conducted some preliminary recordings in ESMUC rehearsals with a real conductor to make sure it was reliable to track his movements. During these recordings, we focused on simply observing if the skeleton tracking

got lost. From our observations, we concluded that this device was **reliable enough**, considering that we are focusing on general characteristics of the movement, rather than on very precise changes.

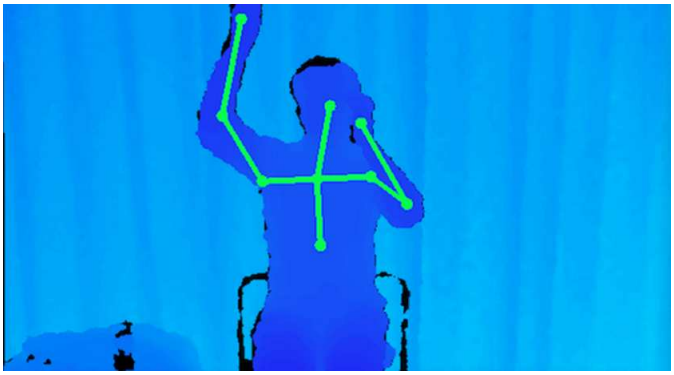

Figure 2: Snapshot from test with conductor.

4.3 Conclusions: focus on the conductor

As we have seen, performing unobtrusive motion tracking from musicians is difficult as most instruments occlude parts of the body that make impossible for the sensor to do a correct skeleton tracking. Existing state of the art works deal with the recognition of objects -and concretely, musical instruments- according to how people interact with them (Yao et al. (2013)), but are not suitable for tracking expressive body motion in a way that is easily applied to real-time interaction (as required in task T6.4). For these reasons, we decided to focus on the acquisition of **conducting** gestures.

This supposes a slight redefinition of T6.4 goal ("[...] to allow offline spectators [...] to control the audio generated by a particular instrument of the ensemble"), in the sense that, instead, we will allow users to control **general characteristics** of the performance related to high level properties of music such as dynamics, tempo, note density, instrumentation or articulation, imitating the role of the conductor.

In order to perform a comprehensive study on conducting gestures, we define a strategy that follows two parallel paths: on one hand, we are interested in recording conductors in **live performances** in order to study how their movement relates to expressive aspects of the resulting performance. In Section 5 we explain these recordings with more detail.

Also, having in mind the goal defined by T6.4, we want to study **how subjects with different musical backgrounds perform conducting movements**. This will allow us to come up with systems where rules for expressive control are not completely predefined and, instead, are adapted to the specificities of users' "**conducting styles**". The intention of this is to let users control music in a more natural and expressive way. Details on the recordings in this scope are given in Section 6.

5 PERFORMANCE RECORDINGS

Motion capture data was recorded in different performances. The recording of motion capture data from the 2013 spring ESMUC festival was already presented in deliverable 3.2 ("Standardised corpus of audiovisual performance recordings from multiple viewpoints").

In that document, we already mentioned that some alignment issues appear when using MS Kinect to record motion capture data in the `.ONI` OpenNI format. In order to overcome these issues, we implemented a new version of the motion capture recorder that, instead of recording data to a binary `.ONI` file, records different streams for RGB and RGBD (depth) videos, audio and skeleton data. By recording all these streams aligned, we can later align them to other data (video, audio) recorded at the performance by manually aligning audio files.

5.1 Recordings with Orquestra Simfònica del Vallès

As a result of a collaboration with Orquestra Simfònica del Vallès (OSV), we conducted multimodal recordings of Beethoven's 9th Symphony in a rehearsal and a concert during may 2014. The details about the available data will appear in D3.3 "Final corpus of audiovisual performance recordings from multiple viewpoints" and the corresponding corpus fact sheets.

Regarding motion capture acquisition, we used the new implementation of the motion capture recorder to record the conductor in a rehearsal and a concert. In order to maximize performance and minimize intrusiveness for the performers, a laptop was placed under the conductor's podium and controlled remotely via WI-FI (using RealVNC) from another laptop. Figure 3 shows a picture taken from the place from which the recording was being controlled in the rehearsal. Figure 4 shows a snapshot of the video stream recorded during the concert.

Once aligned, the data will be uploaded to Repovizz.

Figure 3: Rehearsal recording with OSV

Figure 4: Snapshot of video recorded with Kinect from OSV's concert

6 STUDY WITH DIFFERENT SUBJECTS

As already pointed out, our goal is to recreate the experience of conducting a classical music ensemble by allowing users to affect certain musical cues while playing an audio recording of a performance. In this sense, we considered that, instead of predefining certain rules for control, it would be interesting to first see how actually different subjects would move with no instructions given. Since we pursue a strategy with which users do not need to be trained, but instead the system learns their subjective ways of conducting, we performed the following user study involving 25 subjects.

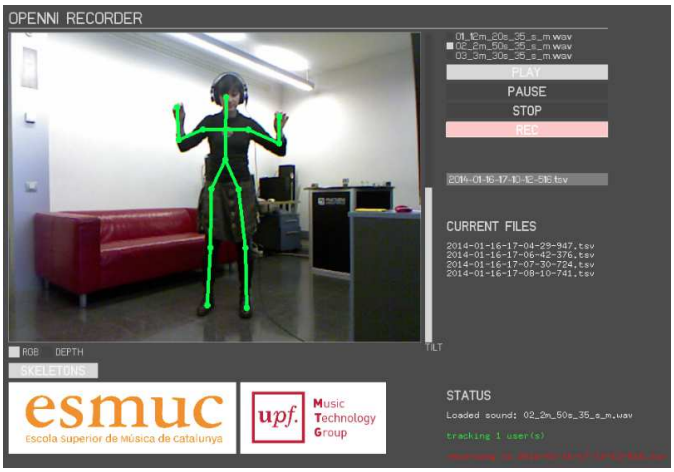

Figure 5: Snapshot of software used during different subjects recordings.

The analysis of these recordings is focused on the two main aspects of expressivity: **tempo and dynamics** (which are also the two parameters users control in most conducting interfaces -see Sarasúa and Guaus (2014b) for a review of these systems-). The goal is to establish how different conducting styles can be identified and exploited to build interfaces that are able to adapt to the specificities of each subject. All the recordings are uploaded in Repovizz and can be accessed online (subjectStudies14). The musical excerpts that were used for the recordings come from RCO's Eroica performance (detailed in D3.2 "Standardised corpus of audiovisual performance recordings from multiple viewpoints")).

6.1 Tempo, beat

Regarding tempo, we performed beat tracking from motion capture data for each subject. We estimated beat positions from acceleration data of the hands. Then, we estimated differences in time anticipation across subjects by building the error distribution of beat predictions, having manual annotations of beat positions as ground truth. We expected narrow error distributions centered at 0 for subjects moving in synchrony with the beat, narrow error distributions centered at a value below 0 for subjects anticipating the beat and wide distributions for subjects not following the beat. This is illustrated by the red, black and blue lines in Figure 6, respectively.

Indeed, this is the kind of effect that was observed after the analysis of the recordings. Figure 7 shows error distributions for 3 subjects from the study. Further details can be found in (Sarasúa and Guaus, 2014a) and (subjectStudies14).

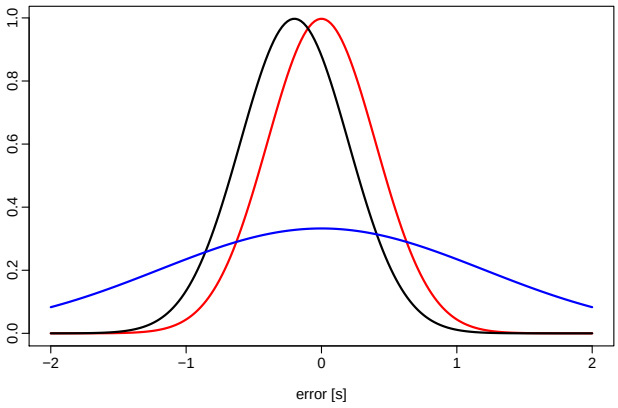

Figure 6: Theoretical error distributions from beat detection.

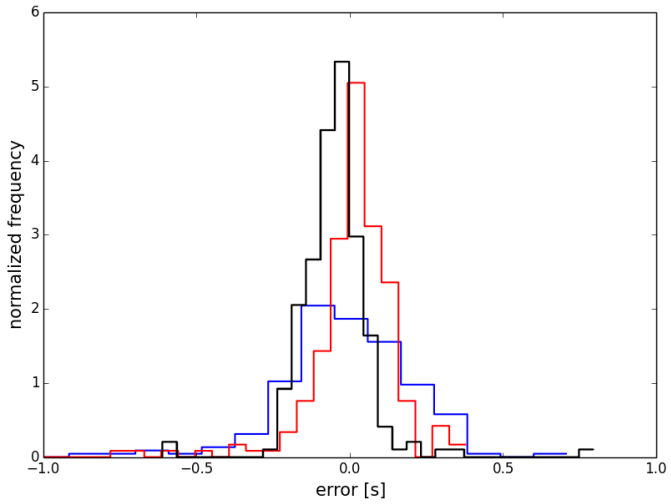

Figure 7: Resulting error distributions from three different subjects.

6.2 Dynamics

We also looked for relationships between movement and loudness in the performance. In this case, we built linear regression models to predict loudness from motion descriptors.

We started from a general model, trained from the data of all subjects. As expected, the resulting linear model was poor in terms of predicting loudness from motion capture data (only 35% of the variability was explained from this model).

Next, we tried to overcome the difficulties of a general model by looking for different tendencies by which users could be classified. In fact, by just looking at the descriptors that were highly correlated to loudness we were able to split subjects into two categories: those for which the Quantity Of Motion (QOM) appeared as a good indicator for loudness and those for which the highest hand position (Ymax) was. These two sets of subjects correspond to those that, in loud parts, moved more or raised their hands higher, respectively. Figure 8 shows how subjects were split into two categories following this criterion. The models for these groups account for more variability (41% and 46%, respectively) than the general model. In addition, if QOM is normalized across users (to adapt to different dynamic ranges across users), the models have still some room for improvement (accounting for 46% and 47% variability, respectively).

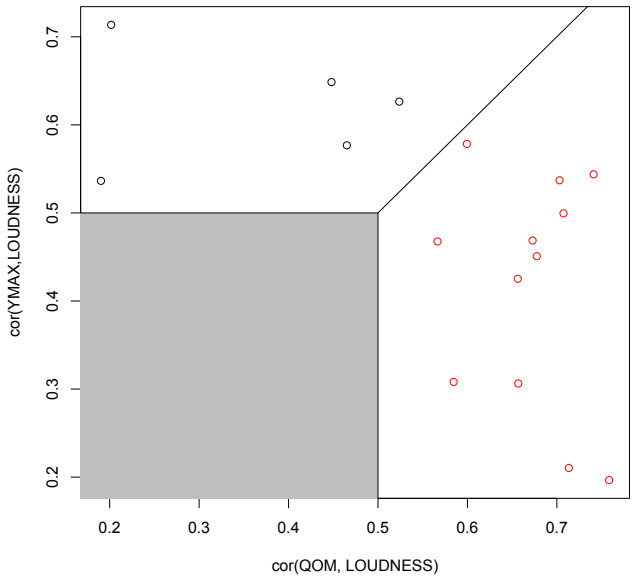

Figure 8: Classification of subjects according to different styles reflected in the correlation analysis.

Finally, it is also possible to train subject-specific models in which, as expected, the explained variability is clearly increased (62% average across subjects).

In summary, our study showed that trying to come up with a general model may not be the right approach in cases where we want to allow users to control the expressive aspects of a performance in an intuitive way. Instead, we can either automatically classify subjects into a set of predefined models for different styles or even train the system to be able to understand how the specific user is embodying expressive information. More specific information on this can be found in (Sarasúa and Guaus, 2014b).

6.3 Work to be done

The conclusions from these studies have to be applied in the real-time scenario where users actually control the performance. Different strategies include using a similar approach as a learn-by-example stage or mapping gesture characteristics to sound qualities (Françoise et al., 2012). A system for conducting a virtual orchestra in real-time is currently being developed to perform new experiments in this sense.

7 GESTURE IDENTIFICATION

With the goal of creating a musical performance reinterpretation game-like experience, we identified two different gestural information levels that could be identified both in a professional conductor level and in a nonprofessional player level. The first one addresses how the conductor's generic movement features are related to high-level and perceptive features of the music, such as loudness, complexity or tempo variation, and how are they perceived and replayed by the users. The other one is symbolic gesture identification: the actual symbols that are drawn in mid-air by the conductor that have a specific meaning to be transmitted to the orchestra. After having covered the first in the previous sections, we address the second in this one.

Hand gestures (symbols) used by conductors in real performances vary greatly. Apart from some isolated instances through the piece, most of the time they are a translation of the high-level perceptive musical properties into a movement that can be understood by the orchestra. This language is not entirely defined, but a construction that the orchestra must learn through the rehearsals with the conductor. In fact, every conductor has his own language (or conducting style), which may differ from other conductors, and dependent to the social and musical context of each concert (Konttinen, 2008).

In contrast, a set of formal, well-defined gestures for conducting exist, mainly used in teaching and rehearsal contexts. These are used to transmit the tempo and beats and have defined rules of how gestures must be performed. We think as them as good candidates for our game-like context, as they would not have to change (and be learned again) with every concert or conductor.

The recognition of the individual beat patterns from a well defined dictionary will allow us not only to achieve the game-like experience in the non-professional setting, but also to create methods to evaluate the quality of gesture performing in the context of conducting classes. We take the objective of building a game-like directing gesture rehearsal program as the motivation to build such recognition methods and testing them.

7.1 Approach

Many previous approaches to identify beat patterns (and other conducting symbols) use machine learning techniques: from the most used methods are neural networks (Ilmonen, 1999) and Hidden Markov Models (Kolesnik and Wanderley, 2004). These techniques use a set of annotated samples to train a model that will be used to classify the incoming live data. This is very appropriate to detect which of the trained symbols is more likely to represent the captured data, but totally unable to estimate the correctness of the performed symbol according to a formal definition.

To identify the gesture symbol type from the data according to a formal pattern we must use an analytic approach that checks the user hand position and movement against this pattern. First, we study the selected gestures set to identify their mandatory components, and build a recognizer that checks if those components are present.

We acquired a corpus of performed gestures to be used as ground truth. Two sets of recordings were acquired: one of an expert performing the beat pattern gestures and another one of a real conducting teaching session. Examining those recordings we asserted that in the class setting, this particular session was focused on practicing real concert gestures and not the canonical beat patterns, rendering the recording as not useful for our case. The expert recording was then selected as the only source of ground truth, leaving out the teaching session set.

A gesture recognizer for every beat pattern was coded to match the formal definition of each gesture and then tested against the recorded data.

7.2 Implementation

We chose GestureAgents (Julià et al., 2013) as our developing framework. It is a framework designed to implement competing recognizers and supports this analytic approach as its default strategy.

Although it was created in a Tangible multi-user tabletop context it is device-agnostic, meaning that it does not assume a particular technology or kind of interaction to be used. Its design allows programmers to implement the gesture recognizer for each particular gesture separately, instead of a general gesture classifier, while it automatically enforces that every input event is consumed at most by only one recognizer. Input data that is not recognized by any of the recognizers is considered not to be a particular gesture. This architecture allows us to optionally add control gestures, other than the beat patterns, to interact with the application, with minimal effort.

For the initial set of gestures we choose these beat patterns (Figure 9, extracted from Gustems and Elgström (2008)), that we will label 1T, 2T, 3T and 4T.

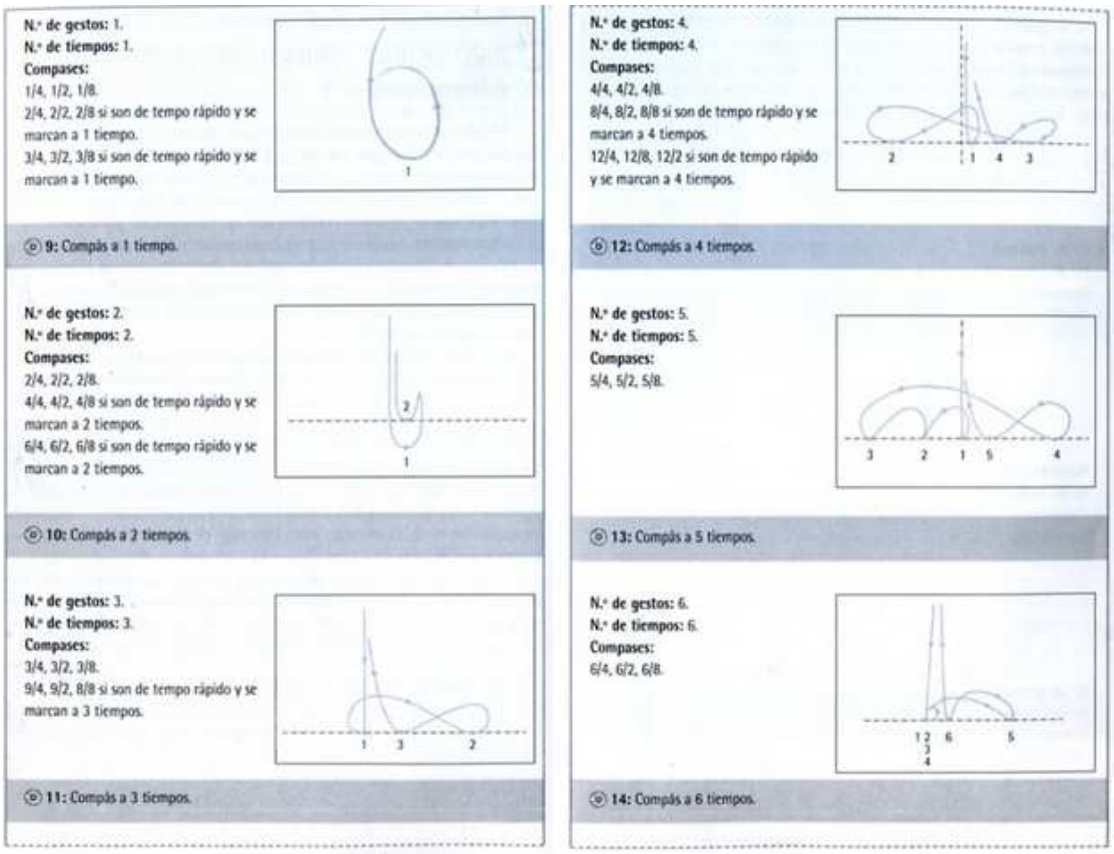

Figure 9: Graphical description of the gestures.

Those gestures are mainly characterized by a sequential vertical movement in two spaces: the upper space, where the gesture begins and ends, and the lower space, where the intermediate parts of the symbol are performed (see Figure 10). We started by characterizing each gesture by the relative positions (not distances) of their alternating vertical local minima and maxima, obtained with a simple sliding window of size w.

$$\text{minima} = \{i \mid y_i \leq y_n \; \forall n \in \{i-w, i+w\}\}$$

$$\text{maxima} = \{i \mid y_i \geq y_n \; \forall n \in \{i-w, i+w\}\}$$

$$\text{keypoints} = \text{maxima}_0, \text{minima}_0, ..., \text{maxima}_m, \text{minima}_m, \text{maxima}_{m+1}$$

As an example, 3T beat pattern gesture consists of 7 key points, characterized by the conditions of Listing 1. Note that by using relative positions instead of distances the recognizer becomes scale invariant.

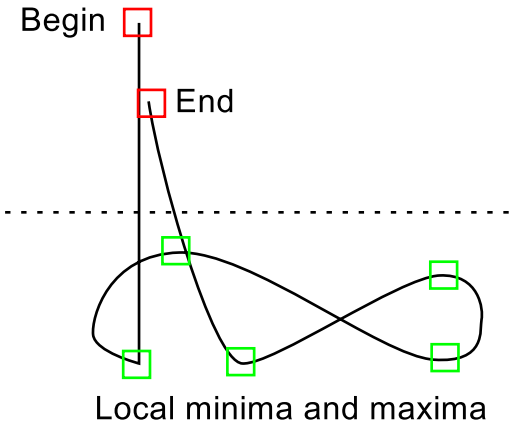

Figure 10: Key points, consisting of vertical local minima and maxima, in a beat pattern gesture (3T).

```
k[1].y < k[0].y      k[2].y > k[1].y      k[2].y < k[0].y
k[3].x > k[2].x      k[3].x > k[1].x      k[3].x > k[0].x
k[3].y < k[2].y      k[4].y > k[3].y      k[4].y > k[1].y
k[4].y < k[0].y      k[4].x > k[2].x      k[4].x > k[1].x
k[4].x > k[0].x      k[5].x > k[1].x      k[5].x < k[3].x
k[5].x < k[4].x      k[5].y < k[2].y      k[5].y < k[4].y
k[6].y > k[2].y      k[6].y > k[4].y      k[6].x < k[4].x
k[6].x < k[3].x
```

Listing 1: Conditions for 3T recognizer.

Another possible mechanism would be using a temporal pattern to match the gesture instead of its spatial trajectory (Bergen, 2012). This strategy, however, requires to use the actual score to predict the expected key points in time, and does not take into account the actual hand position, something essential when evaluating if the shape of a performed gesture also fits the ideal pattern.

After programming the gesture recognizers and testing them separately, we started testing them at the same time within the framework, that would have to automatically manage the disambiguation between the recognizers. We found compatibility problems between our recognizer strategy and the framework's own design preventing successful disambiguation from happening, that required some additional recognizing steps or a slight change on the strategy:

- In some cases the final part of one gesture could be recognized as the beginning of another. This is the case, for instance for 4T and 3T, shown in Figure 11. Because GestureAgents recognizers compete to have the exclusivity over the input events as a mechanism of disambiguation, a recognizer can't successfully recognize a gesture until all other recognizers give up. This confusion, 3T recognizer trying to identify a 3T gesture in the middle of a 4T gesture, delayed the the recognition of the correct 4T gesture, preventing then the following gesture to be recognized at all (because of missed input

events caused by the delay). This string of half failed recognitions has shown a flaw of GestureAgents when dealing with gestures that appear in mid-interaction, in oposition to the gestures in Tabletops, that usually start with the introduction of a new finger on the surface. We solved this problem by segmenting the hand trace using a common start/end gesture feature present on all the recognized gestures: the different vertical position.

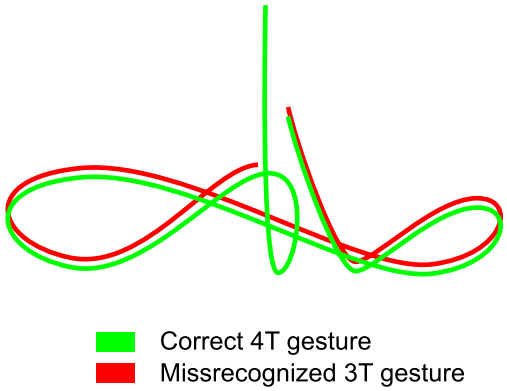

Figure 11: Missrecognition of a partial 4T gesture as a 3T gesture.

- A related problem appeared with the concept of this type of gestures in GestureAgents: with our conducting gestures the last point of one gesture is also the first point of another. This, sharing input events between recognizers, is strictly forbidden by GestureAgents. We changed the segmentation code to repeat the same event at the end and beginning of the following segments to be able to feed different recognizers.
- Several other bugs of GestureAgents were fixed during the implementation. Most of them were simply errors never caught in a tabletop setup, as most of its gestures always start with the introduction of a new interaction Agent, instead of appearing inside an already started interaction trace.
- Also, using recognizer composition (defining a gesture in terms of another one) for segmentation prevented the gestures to compete due to the gesture isolation strategy created by GestureAgents to allow gestures from different programs to compete regardless of its composition. It seems clear that this strategy is problematic in this context, and does not always allow having composition inside the gesture isolation. We faced this problem by extracting the segmentation from the gesture recognizer category and treating it like a sensor (a recognizer that is used as a primary means of interaction events). Another possible solution would have been implementing the segmentation inside the recognizer avoiding composition altogether.

After resolving the aforementioned issues, the final implementation works well and identifies all the gestures that adhere to the definition from the recording.

7.3 Testing

For the ease of testing we implemented an alternative input method based on Leap Motion (Leap Motion), a depth sensor designed to capture hand and tool movement, instead of full body. This implementation allows rapid testing without having to set up a full-body Kinect scenario.

Informal testing shows that it is possible to recognize correct gestures in real time.

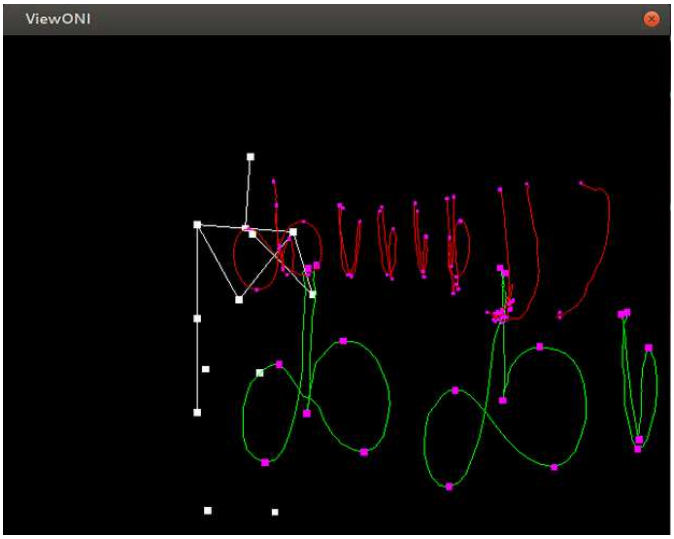

Figure 12: Program recognizing valid gestures from Kinect recordings.

7.4 Work to be done

To complete the game-like directing gesture rehearsal program it is needed to implement the quality evaluation of the gesture. Input from a field expert will be used to quantify and weight the factors that contribute to the quality of a gesture.

8 CONCLUSIONS AND FUTURE WORK

Task T4.2 “Methods for recognising performer’s and conductor’s gesture” primary goal, now addressed in this deliverable D4.4, was to provide the needed technological and methodological foundation for the upcoming task T6.4 “Interactive systems for the performer impersonation”, mainly focusing on its gestural recognition part. We have done that by creating the methods to recognize the conductor gestures (from both real conductors and potential users of the goal application) and their relation with some aspects of the music.

Particularly, we reviewed and selected the sensing technology and tested its efficacy, concluding to use Microsoft Kinect as our primary device, and OpenNI as the middleware to extract the skeleton data. We tested the device capabilities against several performer profiles, including instrumentalists and conductors and we concluded that the most viable approach was focusing on the conductor. By doing this we have further defined the scope of task T6.4, making it more focused and concrete.

To address the specific goal of recognizing the conductors gestures we have defined two different approaches: Recognition of the symbolic part of the gestures, defined by a well defined dictionary used by conductors in several situations, using top-down approach (from the definition to the gesture instance), and the study of the relationship between body movement and the expressive properties of the performance, from the perspective of both conductors and potential users, using a bottom-up approach (from the observations to the creation of a model). The results show a promising potential for both of the approaches, allowing a combination of both methods to be used in T4.2 to achieve a gesturally rich experience. Recordings made for this phase are made available in accordance with the technical requirements described in deliverable D2.1.

The following work that will be done in task T6.4 includes extracting the possible mappings according to the studied conductor and user’s movements, evaluating the variability extracted from symbolic gesture recognition, and overall using the methods described to create a game-like experience to control the music performance by impersonating the conductor.

9 REFERENCES

9.1 Written references

Bergen, S. (2012). Conductor Follower: Controlling sample-based synthesis with expressive gestural input. Master's thesis.

Dahl, S. and Friberg, A. (2007). Visual perception of expressiveness in musicians' body movements. *Music Perception: An Interdisciplinary Journal*, 24(5):pp. 433–454.

Françoise, J., Caramiaux, B., and Bevilacqua, F. (2012). A Hierarchical Approach for the Design of Gesture–to–Sound Mappings. In *Proceedings of the 9th Sound and Music Computing Conference*, pages 233–240, Copenhagen, Denmark.

Godøy, R. I. and Jensenius, A. R. (2009). Body movement in music information retrieval. In *ISMIR 2009*.

Gustems, J. and Elgström, E. (2008). *Guía práctica para la dirección de grupos vocales e instrumentales*.

Ilmonen, T. (1999). Tracking conductor of an orchestra using artificial neural networks. *Master's thesis, Helsinki University of Technology*.

Jensenius, A. R. (2007). *Action–Sound: Developing Methods and Tools to Study Music-Related Body Movement*. PhD thesis, University of Oslo.

Julià, C. F., Earnshaw, N., and Jorda, S. (2013). GestureAgents: an agent-based framework for concurrent multi-task multi-user interaction. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pages 207–214. ACM.

Kolesnik, P. and Wanderley, M. (2004). Recognition, analysis and performance with expressive conducting gestures. In *ICMC*.

Konttinen, A. (2008). *Conducting Gestures: Institutional and Educational Construction of Conductorship in Finland, 1973-1993*. PhD thesis.

Sarasúa, A. and Guaus, E. (2014a). Beat tracking from conducting gestural data: A multi-subject study. In *Proceedings of the 2014 International Workshop on Movement and Computing*, MOCO '14, pages 118:118–118:123. ACM.

Sarasúa, A. and Guaus, E. (2014b). Dynamics in music conducting: A computational comparative study among subjects. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 195–200, London, United Kingdom. Goldsmiths, University of London.

Wanderley, M. M., Vines, B. W., Middleton, N., McKay, C., and Hatch, W. (2005). The musical significance of clarinetists' ancillary gestures: an exploration of the field. *Journal of New Music Research*, 34(1):97–113.

Yao, B., Ma, J., and Fei-Fei, L. (2013). Discovering object functionality. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia.

9.2 Web references

(subjectStudies14) Extra information on subjects recordings and links to Repovizz datapacks. http://alvarosarasua.wordpress.com/research/beat-tracking/

(Microsoft Kinect) http://www.xbox.com/en-US/kinect

(Open Natural Interaction) http://structure.io/openni

(Leap Motion) https://www.leapmotion.com/

9.3 List of authors

Carles F. Julià, Universitat Pompeu Fabra
Álvaro Sarasúa, Escola Superior de Música de Catalunya
Sergi Jordà, Universitat Pompeu Fabra
Enric Guaus, Escola Superior de Música de Catalunya