POLIMOD Pipeline: documentation

Motion Capture, Visualization & Data Analysis for gesture studies

Dominique Boutet, Université de Rouen, France, dominique.boutet@univ-rouen.fr
Jean-François Jego, Université Paris 8, France, jean-francois.jego03@univ-paris8.fr
Vincent Meyrueis, Université Paris 8, France, vincent.meyrueis@univ-paris8.fr

Abstract

We propose a pipeline to collect, visualize, annotate and analyze motion capture (mocap) data for gesture studies. A pipeline is "an implementation of a workflow specification. The term comes from computing, where it means a set of serial processes, with the output of one process being the input of the subsequent process. A production pipeline is not generally perfectly serial because real workflows usually have branches and iterative loops, but the idea is valid: A pipeline is the set of procedures that need to be taken in order to create and hand off deliverables" (Okun, 2010). The pipeline designed here (table 1) presents two main parts and three subparts. The first part of the pipeline describes the data collection process, including the setup and its prerequisites, the protocol to follow and how to export data regarding the analysis. The second part is focusing on data analysis, describing the main steps of Data processing, Data analysis itself following different gesture descriptors, and Data visualization in order to understand complex or multidimensional gesture features. We design the pipeline using blocks connected with arrows. Each block is presenting a specific step using hardware or software. Arrows represent the flow of data between each block and the 3-letter acronyms attached refer to the data file format (table 2). The development of the pipeline raises three main questions: How to synchronize Data ? How to pick data and transform it? And what is changing in annotation? To solve the question of data synchronization, we design a protocol where we detail how hardware has to be properly selected regarding the type of measures and the protocol to follow implies specific steps for the participant such as adopting a T-Pose, or clapping their hands once at the beginning and the end of the recording to facilitate data synchronization and export in the next steps. About picking relevant data and transforming it, we propose to select and prepare files to export regarding the analysis and software expected. Thus, mocap files could be converted to videos to be visualized for instance in Elan to enhance gesture coding or converted to text files to be analyzed in Excel, or processed in Unity to explore the flow of movement, new gesture features or kinematics. We detail all these processes in a step-by-step tutorial available in an open access. Finally, we question what a pipeline involving Mocap is changing in annotation. We notice mocap allows no more a single point a view but as many as required since we can use virtual camera to study gesture from the "skeleton" of the participant. For example, we show it is possible to adopt a first-person point of view to embody then better understand participants gestures. We also propose an augmented reality tool developed in the Unity3D software to visualize in real-time multidimensional gesture features (such as velocity, acceleration, jerk) or a combination of them in simpler curve or surface representations. As future direction, data collected here could be used for a machine learning algorithm in order to extract automatically gesture properties or automatically detect and tag aspectuality of gestures. At last, an embodied visualization tool using virtual reality could thus offer newer possibilities to code and understand gestures than using a 2D video as a reference or study material.

Introduction & overview

We propose a pipeline or workflow to collect, visualize, annotate and analyze motion capture data for gesture studies. The purpose of this documentation is to offer an open and accessible workflow for gesture studies involving motion capture which often generate huge amount of data. Also, regarding the recurrent question of experimentation reproducibility, this approach would help in designing reproducible experiments. The term pipeline "comes from computing, where it means a set of serial processes, with the output of one process being the input of the subsequent process. A production pipeline is not generally perfectly serial because real workflows usually have branches and iterative loops, but the idea is valid: a pipeline is the set of procedures that need to be taken in order to create and hand off deliverables" (Okun, 2010).

The pipeline designed here (table 1) presents two main parts and three subparts. The first part of the pipeline describes the data collection process, including the setup and its prerequisites, the protocol to follow and how to export data regarding the analysis. The second part is focusing on data analysis, describing the main steps of Data processing, Data analysis itself following different gesture descriptors, and Data visualization in order to better understand complex or multidimensional gesture features.

1. Data collection			**2. Analysis**		
1.1 Setup Device and characteristics	**1.2 Protocol** before & during Recording	**1.3 Export Data** and filenames	**2.1 Data Process** (Extraction & Merging)	**2.2 Data Analysis** Elan / Excel / Unity	**2.3 Visualizing results** and interpretation

table 1: pipeline overview

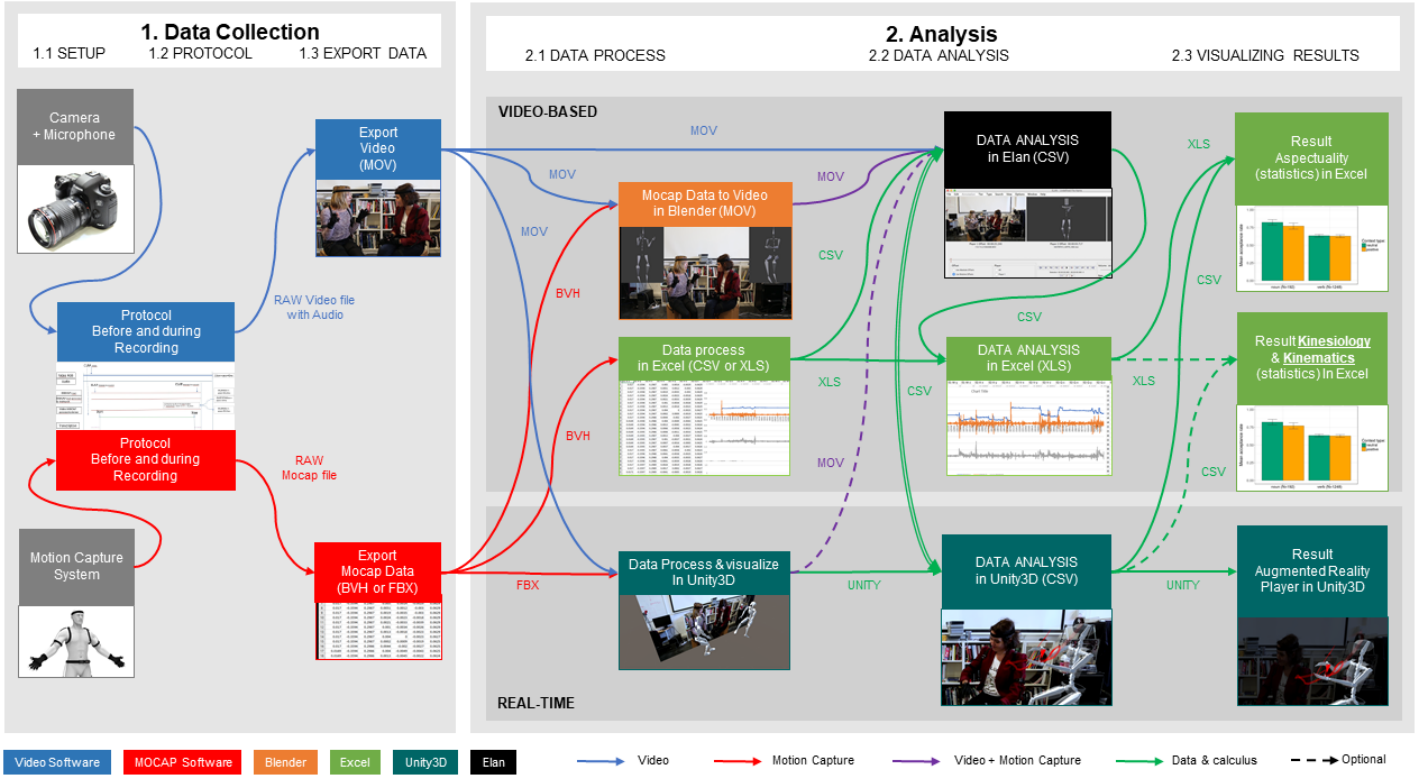

table 2: detailed pipeline

The table 2 presented above describes the whole pipeline design using blocks connected with arrows. Each block is presenting a specific step using hardware or software. Arrows represent the flow of data between each block and the 3-letter acronyms attached refer to the data file format. To facilitate its understanding, we propose to describe first the data collection. We will describe then the Analysis regarding each type of results expected.

1. Data Collection

The data collection process is the first part of the pipeline, we detail the **Setup & device** and the prerequisites. Then we propose a **Protocol** to follow with specific guidelines and finally we describe how to **Export data** depending on the analysis tools and outcomes.

1.1. Setup & Device

During the data collection process, video, audio and mocap have to be recorded simultaneously. We recommend selecting the recording device depending on its frame rate, especially for the camera, the higher frame rate (or FPS for Frame Per Second) is the better, keeping in mind 25 FPS is a minimum to understand movement, 60 FPS gives a better accuracy especially in slow motion. Since video and audio channels are now recorded with the same device, there is no special requirement for synchronizing data, be sure the sound is recording properly and the frame rate is correct.

Regarding Mocap, the selection of the device is important. Indeed, many motion-capture system exists (mechanical, optical, inertial…), with different use cases, accuracy and price. In our case, we opt for an inertial system (Perception Neuron) which appears to be more convenient in terms of accuracy (millimeters), frequency (from 60hz to 120hz), and this type of mocap suit solves occlusion problems compared with an optical system. We, however, had to be very careful regarding the experiment conditions and environment. Actually, inertial motion-capture system uses IMU (Inertial Measurement units) which are very sensitive to the ambient magnetism of the environment. We suggest to first learn more about the way IMU are working (Magnetism, Gyroscope, Accelerometer). If the magnetic environment is totally different from the one when the sensors were calibrated we recommend to a full calibration of the sensors. If some sensors present a latency despite a proper calibration, we suggest avoiding placing them on critical joints for the recording (such as root, hip, chest, head...) putting them on the body parts not needed for analysis (for instance in the legs of the feet).

1.2. Protocol

The protocol involving motion capture is slightly different from a traditional data collection using a single video camera.

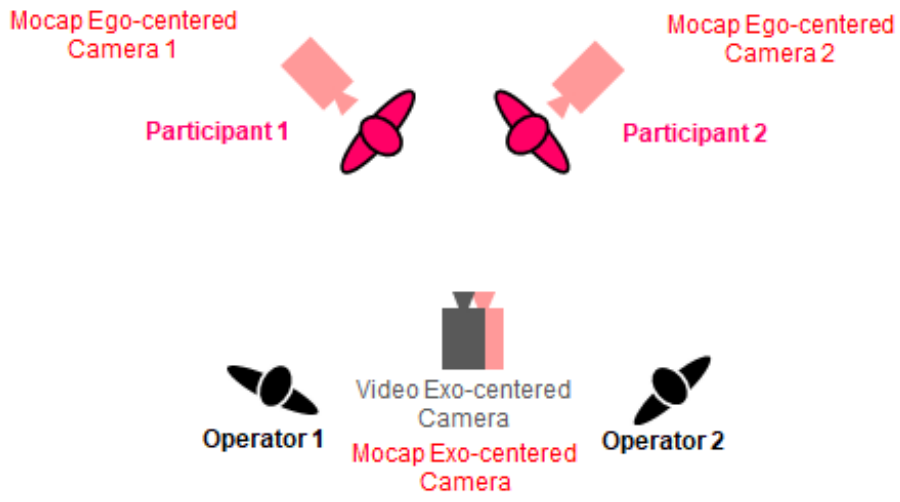

figure 1: data collection of two participants using video camera (in gray) and motion capture "virtual" cameras (in red)

Before the recording, since in our case the inertial mocap system is magnetic sensitive, be sure the material (table, chairs, background) does not contain metal or magnets. Participants clothes and their personal environment (no black cloth, no black background, prefer solid background…) has to be compliant with the general counseling about the equipment of the suit and the straps placement. Dressing the participant with the mocap system has to follow the correct placement of the sensors on the body, also check the tightness of

the straps (not too loose neither too tight). To calibrate the virtual body of the user, information such as gender, or height have to be collected. At last, participants have to be instructed about the calibration phase which consists of a sequence of postures in order the motion-capture system calibrates the virtual body (avatar) to fit the participants body measures. Also, as any data collection process, be sure to naming the files correctly and the folders especially since mocap involves generating lots of files and data.

During the recording phase of the data collection, we recommend the participant to adopt a specific posture called T-Pose. This pose is required then to set up properly the virtual body (avatar) in a 3D engine. It is important to synchronize the mocap with the video and the audio recordings. To do so, we suggest the participant after the T-Pose to clap their hands once at the beginning of the recording and also one more time at the end of the recordings. In this way, operators can do a quick evaluation for the accuracy of the recordings and check if data is not missing or drifting.

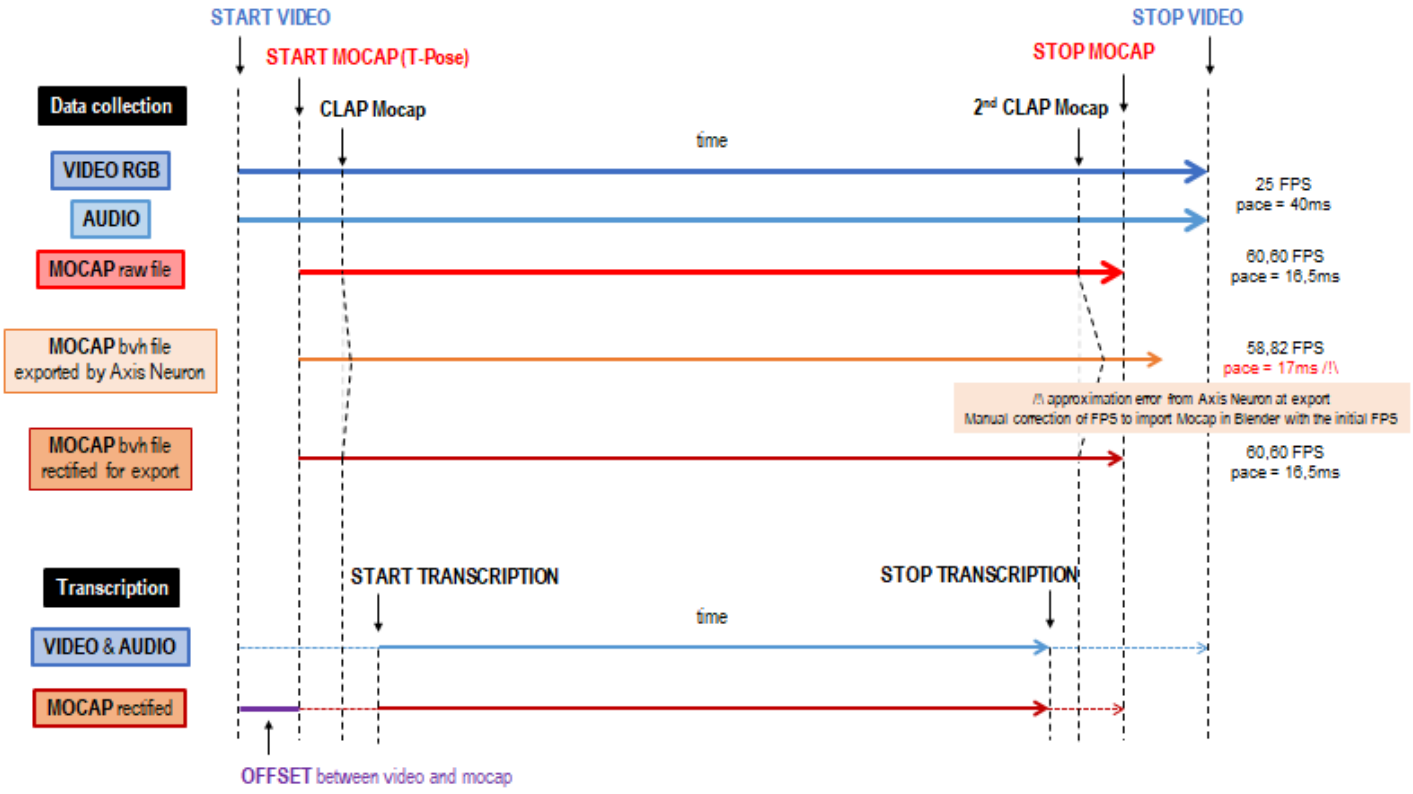

figure 2: synchronizing timelines with posture, claps and explanation of data asynchrony

1.3. Export Data

After the data collection, we need to synchronize properly the audio/video with the motion recorded. To do so, we have to be sure of the frame rate of each file. For instance, a video file has typically a frame rate of 25 FPS, 30 FPS or 50 FPS. Regarding the motion capture, the device we use (Perception Neuron) record movement at 60,6060 FPS (pace = 16,5 ms). However we note a simple export to BVH sets in the exported file a pace at 17 ms which appears to be an approximation of 16,5 ms. This results in obtaining a longer motion capture file (in time) than the raw mocap file since the same number of frames is played at a slower pace. In this case, we have to correct the file export manually in order to generate for instance a video in the Blender 3D software. In this way, the total length is correct and can be synchronized properly with the audio and video files from the camera.

Regarding the file types, video files and mocap can then be exported to different file formats. Please note there are sometimes some compatibility issues (for instance MacOS MOV file types are not always readable for Windows PC). In our case, we opt for the H264 open source codec in MOV video files.

Finally, about the filenames, different templates exist (DCNC, 2018) but some of them could be too exhaustive

which makes more difficult to read. In our protocol, we choose uppercase initials, numbers and dash for separation. Thus, we write VR for Video Recording (VR1, VR2, if multiple cameras), SR for Sound Recording if the sound track is separated from the video, MC for Mocap Capture recording, and VM for Video Motion capture (VM1, VM2, if we render multiple points of view). We associate then to each participant's language initials of the group (for instance RU for Russian, FR for French participants) and a number. We also add to the end of the filename the heights of the subject as a suffix for specific software export where this information is needed. We obtain for instance MC-RUFR01-1m66 for the Motion Capture file of the participant 01 speaking Russian as a native language and French as a second language and whose height is 1.66 meters.

2. Analysis

The analysis part is the second main part of the pipeline we design. This part is also divided in three steps. The first one in the **Data Process** which consists of extracting or merging data from the raw file of the data collection. The we proceed **Data Analysis**. This step is dependent on the type of studies or output expected. In this study, we wish to both conduct statistical analysis but also understand movement and find new gesture descriptors in action, i.e. using real-time playback and visualization. The last step consists of **Visualizing results** and its interpretation.

2.1. Data Process

The data process is dependent on the Data export and is necessary to prepare the Data analysis. In this project, we wish to obtain 3 types of results (statistics, gesture analysis and gesture descriptors) which consist of analyzing gesture using Excel, using Elan and using Unity 3D. This implies each software doesn't use the same file format.

The most standard and open mocap file format is BVH (Biovision Hierarchy). A mocap file is generally composed of a header and a table of data. The header of the mocap file defines the skeleton, the height or size of each bone and the pace to play the file. In the table, each column corresponds to specific coordinates of each joint of the body and each row is the value of the coordinate at each frame. When exported as an ASCII text file (such as a BVH) the mocap file could be open and manipulated or transformed in Excel. But we could also import the mocap file in Unity3D in a binary format (such as FBX file format) which includes extra feature to simplify the import and, thus, makes import more convenient in 3D software. In this study, we need to prepare data regarding our three main outputs: Generating an Excel file from the mocap, Generating a skeleton in video from the mocap and Generating a real-time skeleton (avatar) from the mocap.

To generate a video with the mocap skeleton, we use the Blender software which can import natively BVH file format and allow to render sequences of images and, so, videos. Be sure the frame rate of the BVH file is correct. Indeed, in our workflow, we discover while using Blender that Axis neuron software creates an approximation error during export. This results in generating a video with an incorrect length. To solve this issue, we had to manually correct the BVH file as we detailed in the previous figure (figure 2).
We also develop a real-time tool using Unity3D to visualize skeleton and gesture descriptors. Unity3D software is mainly developed and used for real-time data processing and visualization but it also allows to export videos, we propose then in the pipeline an optional MOV export from unity 3D to Elan software for annotation (table 2)

2.2. Data Analysis

In this project, we conduct three types of data analysis: A. coding bounded and unbounded gesture in Elan, B. Determining the way to find the flow in data computing and C. Determining the way to find the kinematics in data computing. Depending each type of analysis, we had to generate specific mocap file exports (as detailed in table 2).

A. Coding aspectuality (bounded and unbounded gesture) in Elan

This part is following a previous study investigating gesture aspectuality. Bounded and unbounded gestures have been coded using video. Using our pipeline which adds mocap, we have created a new corpus to analyze gesture aspectuality across two languages : Russian participants speaking Russian as a first language, Russian participants speaking French as a second language, and French Participant speaking French as a first language (namely RURU, RUFR, FRFR). For this study, we design in the pipeline a workflow to generate videos from the mocap files using the open source software Blender. As we detailed in the previous part (2.1 Data process) Blender is a 3D rendering software that allows to generate sequence of images thus to generate video. Blender also integrates a video editing tool which allows to create picture in picture videos with the original video and audio from the camera and the mocap files rendered (figure 3). We can then import this video in Elan and annotate gesture with mocap files superimposed in the video.

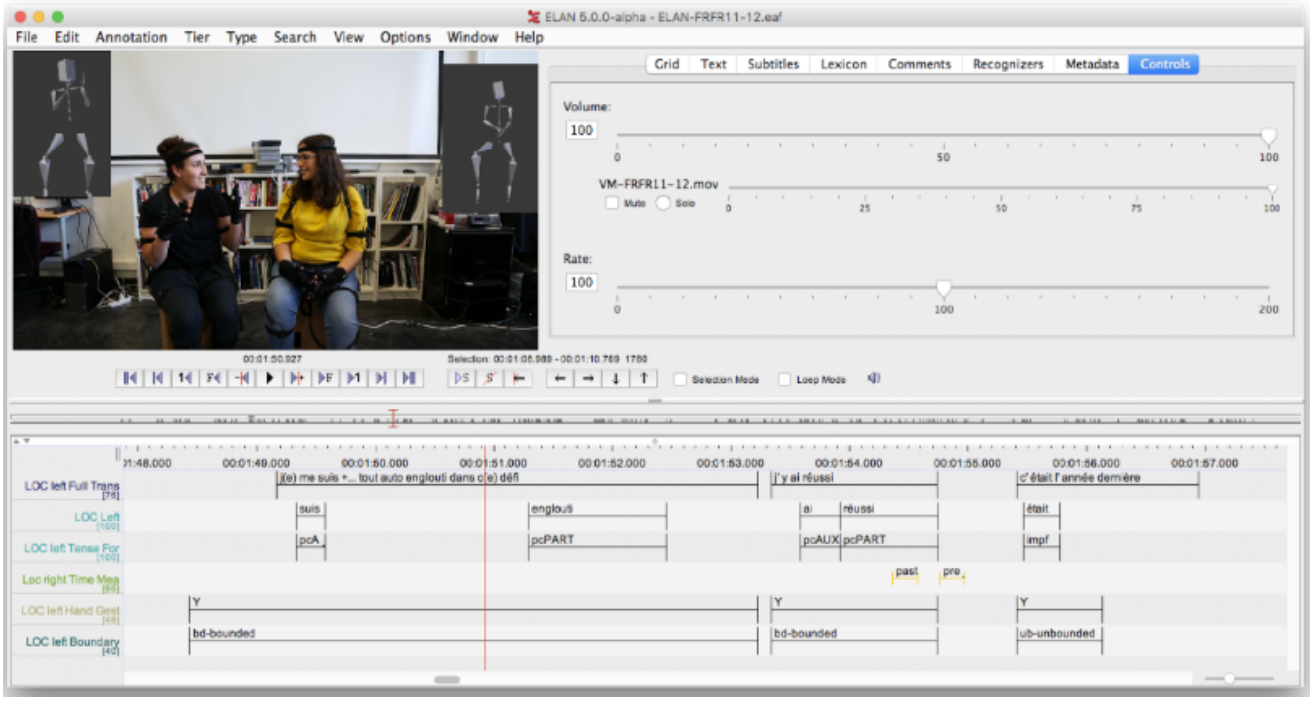

figure 3: coding bounded and unbounded gesture in Elan with video and mocap rendered picture in picture in the video

B. Determining the way to find the flow in data computing

A second analysis we chose to explore is the flow of movement. The flow is a transfer of a movement from one dof to another which determines the expansion of a shape on the upper limb. Regarding the Kinesiological criteria, we have:

- (quasi-)Co-linearity between Mvts ;
- (quasi-)Co-temporality between these Mvts :
    - with a temporal lag ;
    - without any temporal lag.

Unfortunately, at this moment, we are not able to measure the flow according to the three methods exposed above. We are able to approach the flow through three methods on one base. Our approach of the flow is calculated at every frame. The flow of each gesture is determined by the sum of every frame (simple arithmetic) according to:

1/ sliding windows ;
2/ mixing ratio between the degrees of freedom (dof) ;
3/ estimating the thresholds on the flow.

All the process is done with Excel. This software is not the most convenient to analyze data, but it presents some advantage. i/ its accessibility, ii/ a treatment more transparent for researchers who do not know how to code with C language or Python, iii/ The data coming from the Mocap, and from ELAN are compatible with Excel. The numbers of steps are numerous and are detailed in the step-by-step tutorial.

C. Determining the way to find the kinematics in data computing

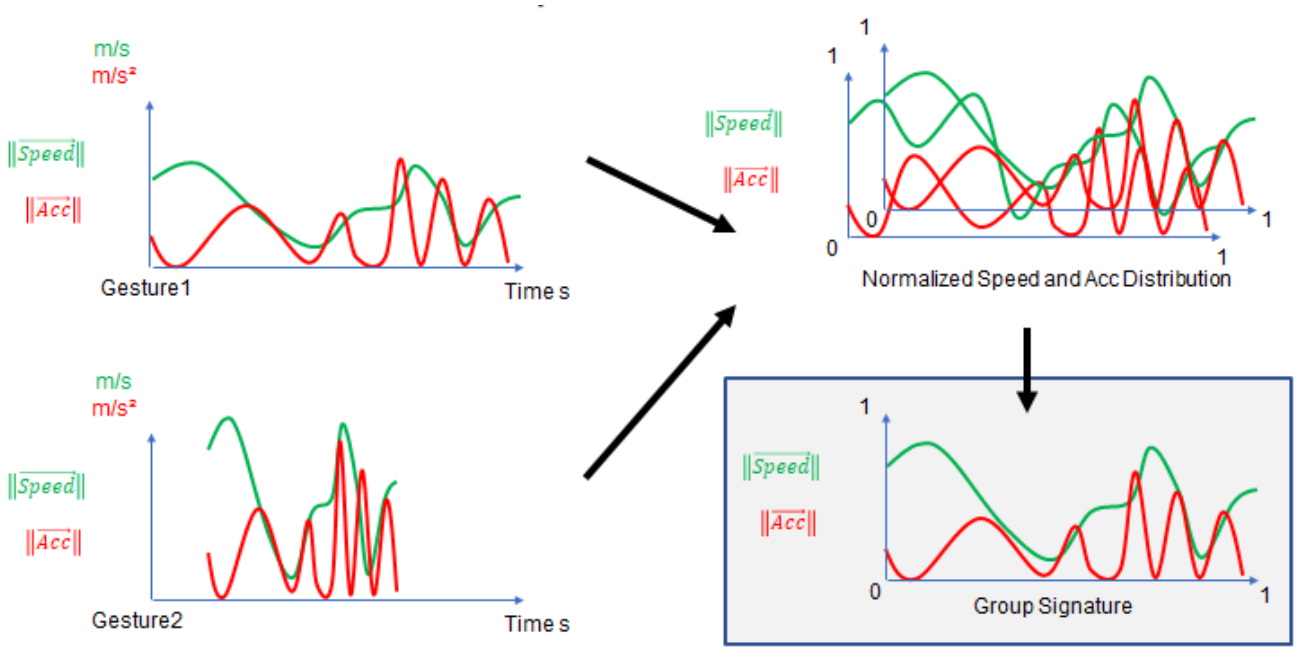

figure 4: Gesture comparison and development of a group signature

In order to compare the gestures between them, we chose to study the normalized distributions of the vector's magnitudes of velocity, acceleration and jerk (Alemi 2014) on the basis of the normalized gesture execution time with 50 samples. By combining all these values, we can compute an average distribution that allows us to characterize the "gesture dynamic" (figure 4) according to unbounded and bounded video coding and the populations studied here (FRFR, RURU, RUFR).

To do this, we have developed in addition to the pipeline our own information extraction and visualization tools that compute the velocity, acceleration and jerk vectors on the basis of the positions expressed in the Cartesian reference frame used for motion captures. The tool also standardizes and formats the data so that it can be analyzed in a second step with statistical tools. Results are discussed in the next part.

2.3. Visualizing Results and Interpretation

Depending each analysis, the results can be presented in plots with statistical analysis or curves. We also developed a visualization system in Unity 3D designed to visualize gesture descriptors such as kinematics in real-time. We detail the different results in the next paragraphs.

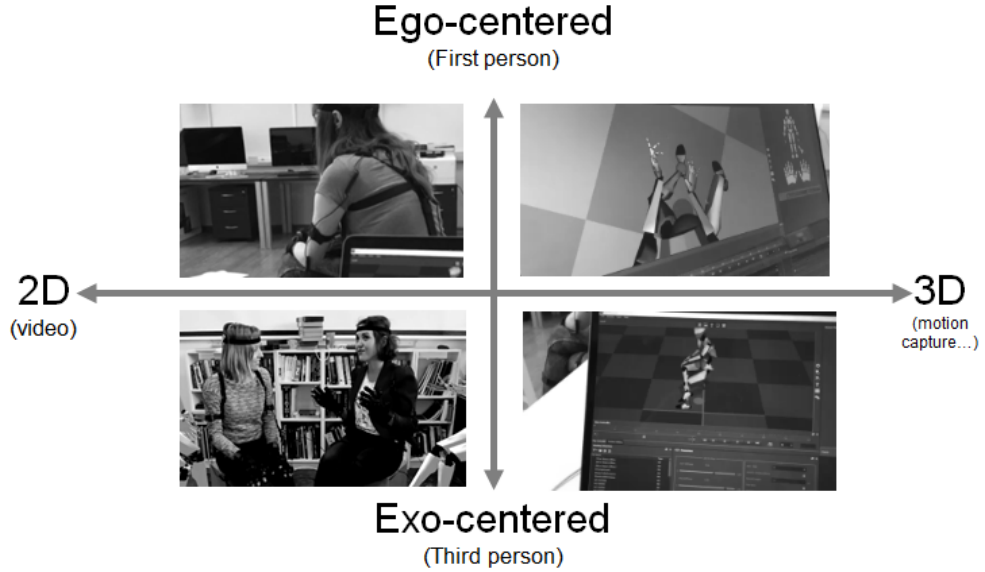

figure 5: ego or exo-centered points of view (Berthoz, 2000) regarding 2D video or 3D mocap data collections

Visualizing aspectuality and statistics in Excel

After the gestures were coded in Elan, we are able to export the tire to a CSV file to analyze data in Excel. This part of the pipeline is not different from a classic gesture analysis except the analyze itself has been done with video and mocap video which allows a better understanding of gesture.

Indeed, using the mocap video we are able to "separate" the body envelop filmed by the camera from the Skeleton. Also, the mocap system we use doesn't capture the facial expressions of the participants. Some motion-capture system exists to capture facial movements, but in our study, we focus mostly on arms and hands movements. Facial expressions were not taken into account and then couldn't interfere in the process of gesture annotation in Elan.

Also, mocap allows to generate multiple points of view (figures 1 and 5). For instance, an exocentered (third-person) point of view close to the one of the video camera. We can also use the advantage of the virtual cameras used in mocap to render for instance a mocap video with a wider point of view or a close-up on specific body part of the participant. Finally, we can adopt an egocentered (first-person) point of view of the participant to visualize and embody the gesture which is hard to achieve with a classic setup using a video camera.

Kinesiology & Kinematics: Statistics and Curve Analysis in Excel

Regarding Kinesiology, we are able to estimate the range or motion of each degree of freedom (dof) for each segment (arm, forearm, hand). On this basis, we extract the biggest amplitude at every frame. The dof thus marked is the one on which the second-largest amplitude determines finally the flow. The flow is calculated for every frame and the sum up is done for 1/ the whole gesture, 2/ the first half and 3/ the last half gesture. We note there are no significant differences between these three moments of each gesture. We decided then to focus on the whole gesture.
A second way we introduce to be more realistic is a weighting according to the motion of each segment. The assessment relies on an estimation of the range of motion. For instance, the complete amplitude of the flexion/extension of the hand is more than twice the one of the abduction/adduction of the same segment. The weighting is the double for the latter dof. The amplitude of the dof of the arm does not cover their full ranges of motion. The mixing ratio overstates the motions for these dof. We have selected this mixing ratio.
A third method to approach the flow introduces the standard variation for each dof. The dof follows a bell curve. We expect that every time a value is outside the standard variation, it is meaningful. We overrated the dof every time and this is critical to estimate the flow. All of our analysis have followed this last two ways to calculate.

Regarding Kinematics, analyses were conducted on 15 pairs of participants. The analysis of the Motion capture is made on the basis of the video coding data under the Elan software by 3 different coders. We focus on the gesture analysis of gestures with the best triple consensus tagging which represents 348 gestures noted bounded and 148 gestures noted un-bounded for all the user panel. All of the results are presented in the pipeline step-by-step tutorial.
The motion capture analysis shows that both hands have the same behaviors, the bounded gestures have an increasing velocity, acceleration and jerk distribution profiles due to the nature of their shape. Regarding the un-bounded gestures, their velocity, acceleration and jerk distribution profiles are less pronounced than the bounded gesture ones due to less affirmed gesture dynamic. In both analysis, given the number of samples, the standard deviation remains more or less constant on each sample. Overall, there is a significant visual difference between the curves that makes possible to distinguish bounded and unbounded gesture on the basis of this gesture analysis approach.
The main outlook of an approach to automatically distinguish each bounded and unbounded gesture from the mocap data could be the use of a learning machine algorithm based on these curves and the gesture video coding as a ground truth.

Augmented Reality Player in Unity3D for gesture descriptors

We developed in the pipeline a new visualization system for gesture descriptors in order to explore gesture kinematics in real-time. This tool helps in perceiving and understanding participants gesture. But it is also questioning how such a pipeline involving Mocap is changing in annotation (figure 6). We show in the previous part how mocap allows no more a single point a view but as many as required since we can use virtual camera to visualize the skeleton of the participant (figures 1 and 5). We can now adopt first person point of view to better understand participants gesture. With the tool developed in the Unity3D software, we are now able to visualize in real-time multidimensional gesture features (such as velocity, acceleration or jerk) or a combination of them in simpler curve or surface representations. We can then study gesture in an immersive and embodied point of view of the participant and visualizing its gesture in a real-time simulation using augmented reality.

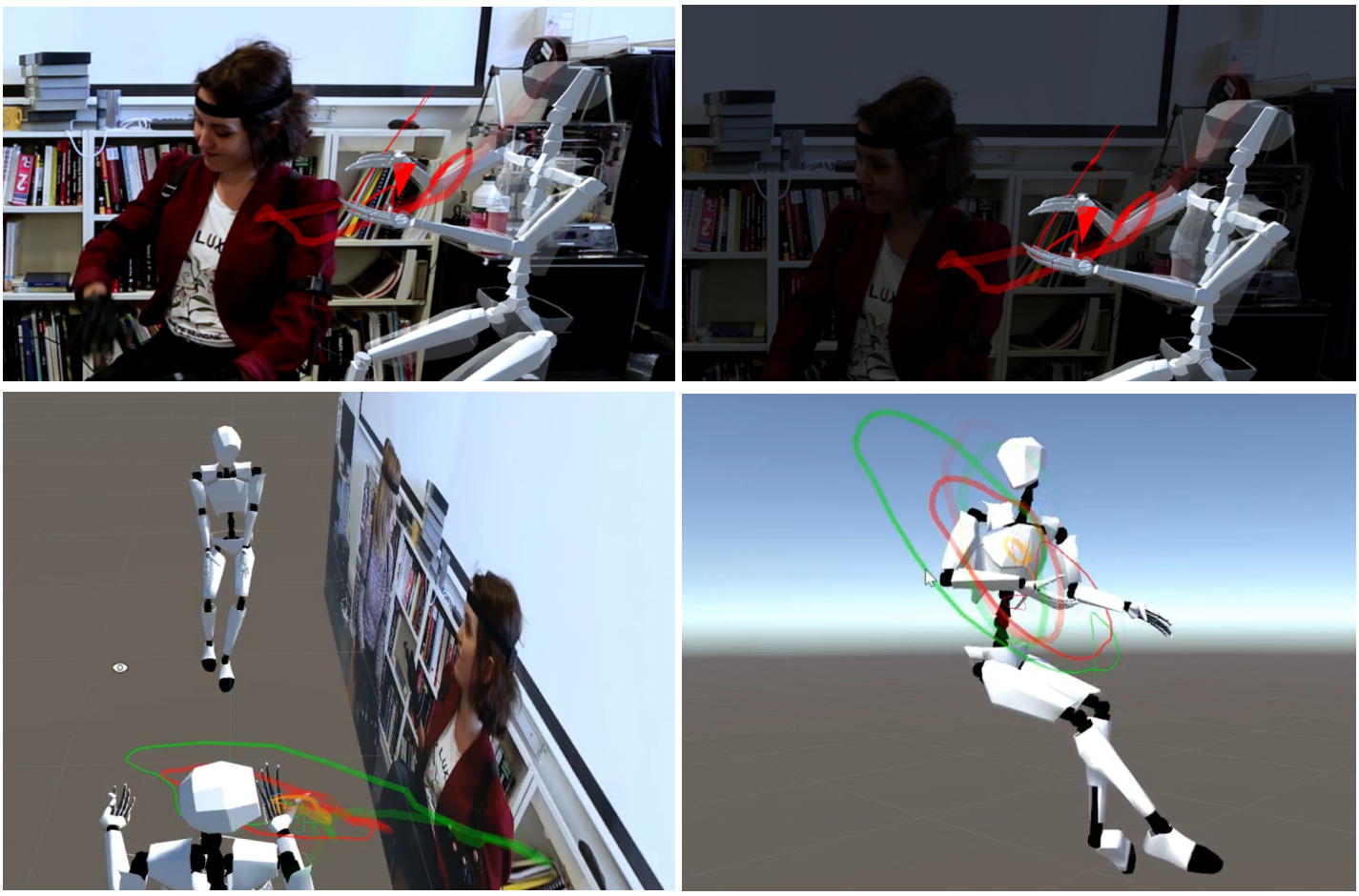

figure 6: Top: augmented reality visual rendering of gesture kinematics, focus can set on the video or the avatar. Bottom: capture of different points of view in the real-time viewport. Legend of the trails of the right hand of the participant in the right: the position is in orange, the velocity is in green and the acceleration is in red.

Conclusion and Perspective

The development of the pipeline raises three main questions: How to synchronize Data ? How to pick data and transform it? And what is changing in annotation? To solve the question of data synchronization, we design a protocol where the hardware has to be properly selected regarding the type of measures and the protocol to follow implies specific steps for the participant such as adopting a T-Pose, or Clapping their hands once at the beginning and the end of the recording to facilitate data synchronization in the next steps. About picking relevant data and transforming it, we expose how to select and prepare files to export regarding the analysis and software expected. Thus, we propose a process to convert mocap files to videos to be for instance visualized in Elan in order to enhance gesture coding. Mocap can also be converted to text files to be analyzed in Excel or processed in Unity to explore the flow of movement, new gesture features or kinematics. We detail all these processes in a step-by-step tutorial available online. Finally, we question what such a pipeline involving Mocap is changing in annotation. We notice mocap allows no more a single point a view but as many

as required since we can use virtual camera to observe the skeleton of the participant. We can now adopt first person point of view to better understand participants gestures. With the tool developed in the Unity3D software, we are now able to visualize in real-time multidimensional gesture features using augmented reality. We could then imagine combining immersive or embodied the point of view of the participant and visualizing its gesture in a real-time simulation in virtual reality is the content is no more presented on a screen but inside a VR headset.
We finally propose a few future directions investigating new gestural descriptors for kinematics. The data collected here could be used for a machine learning algorithm in order to extract automatically gesture properties or for instance to automatically detect and tag aspectuality of gestures.
At last, an embodied visualization tool using virtual reality could thus offer newer possibilities to code and understand gestures than using a 2D video as a reference or study material.

Acknowledgment

We wish to thank all the participants of the pre-tests and the study. The research was carried out at Moscow State Linguistic University and supported by the Russian Science Foundation (project No. 14-48-00067П).

References

Alemi, O. & Pasquier, P. & Shaw, C. (2014). Mova: Interactive Movement Analytics Platform. ACM International Conference Proceeding Series.
Berthoz, A. (2000). The brain's sense of movement (Vol. 10). Harvard University Press.
DCNC Digital Cinema Naming Convention V.9.5 (2018). Retrieved from http://isdcf.com/dcnc/
Okun, J. A., & Zwerman, S. (Eds.). (2010). The VES handbook of visual effects: industry standard VFX practices and procedures. Taylor & Francis.