POLIMOD Pipeline: step-by-step tutorial

Motion Capture, Visualization & Data Analysis for gesture studies

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

Abstract

This open-access tutorial describes step-by-step how to use motion capture for gesture studies. 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 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 of contents

Pipeline Overview

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

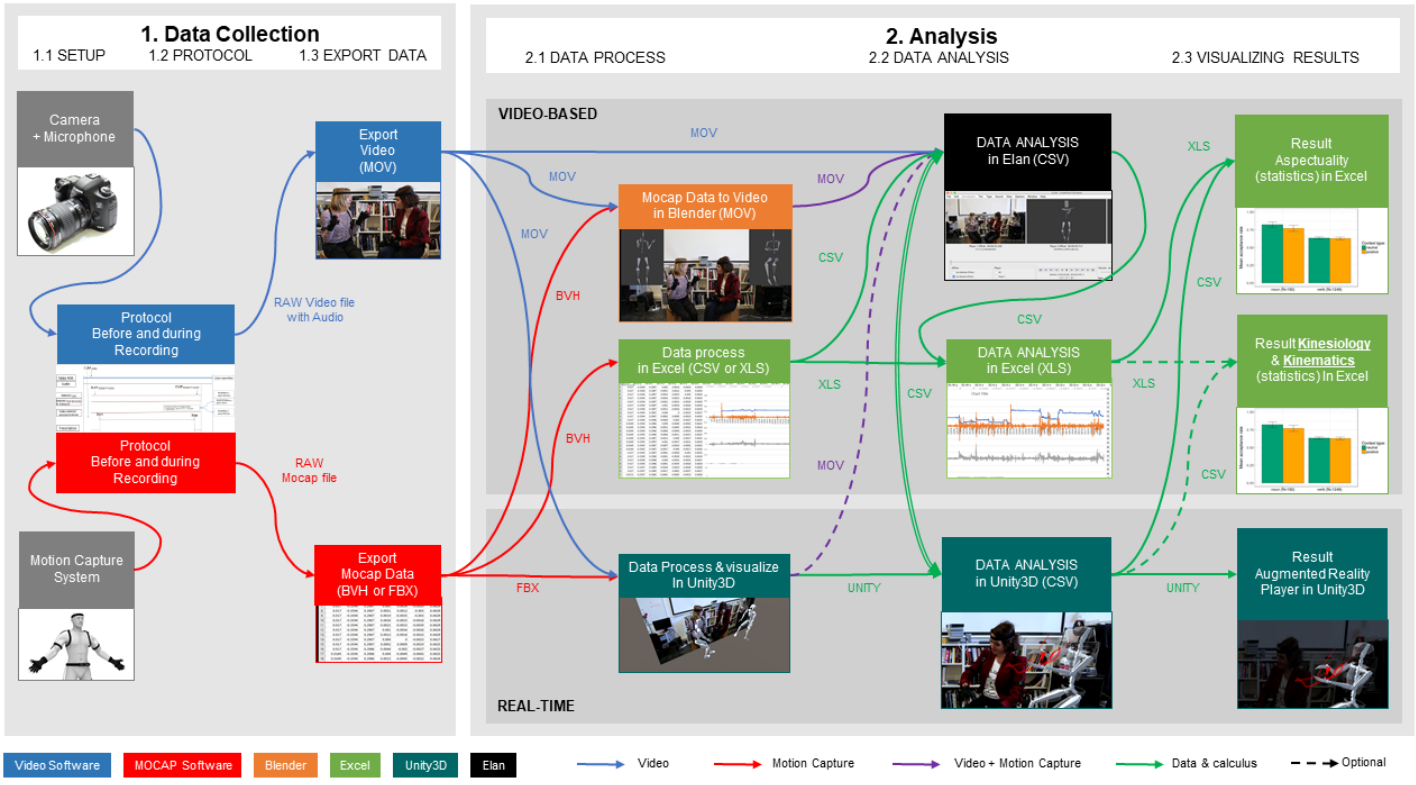


table 2: detailed pipeline

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.

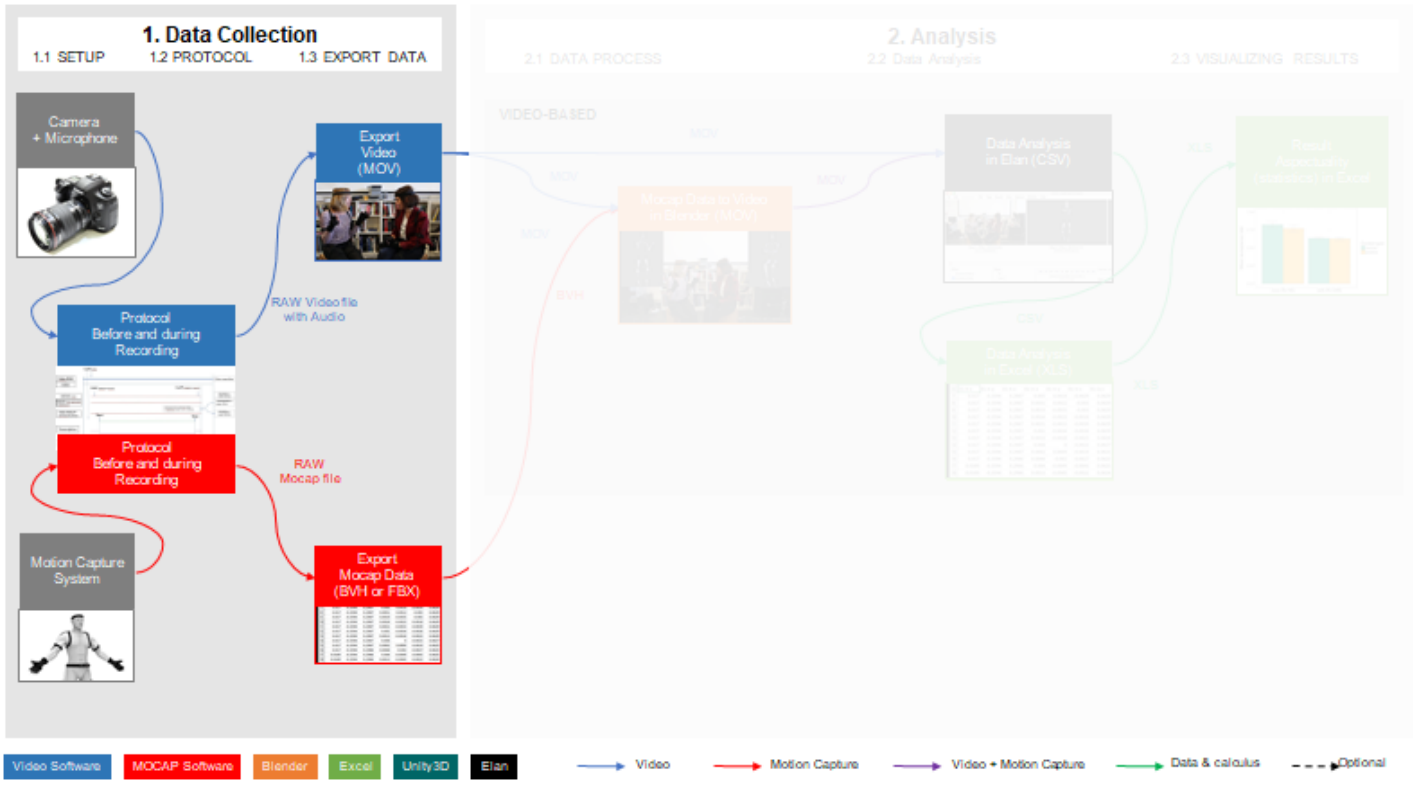


figure: data collection step

1.1 Setup & Device

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

1.1.A Select the Device

Camera : 25 FPS, or 50 FPS (more accurate) or 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.

1.1.B Conditions and Environment

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). To sum up, here is the check list:

- Magnetism of the environment (general recommendations, how to measure it)
- Minimal knowledge about how the IMU work (Magnetism, Gyroscope, accelerometer)
- (Re)-Calibration of the sensors
- What we have to do with sensors which present a latency (putting them on the body parts we won't need for analysis, ex. legs and feet, avoiding critical joints such as root, hip, chest, head...)

1.2 Protocol

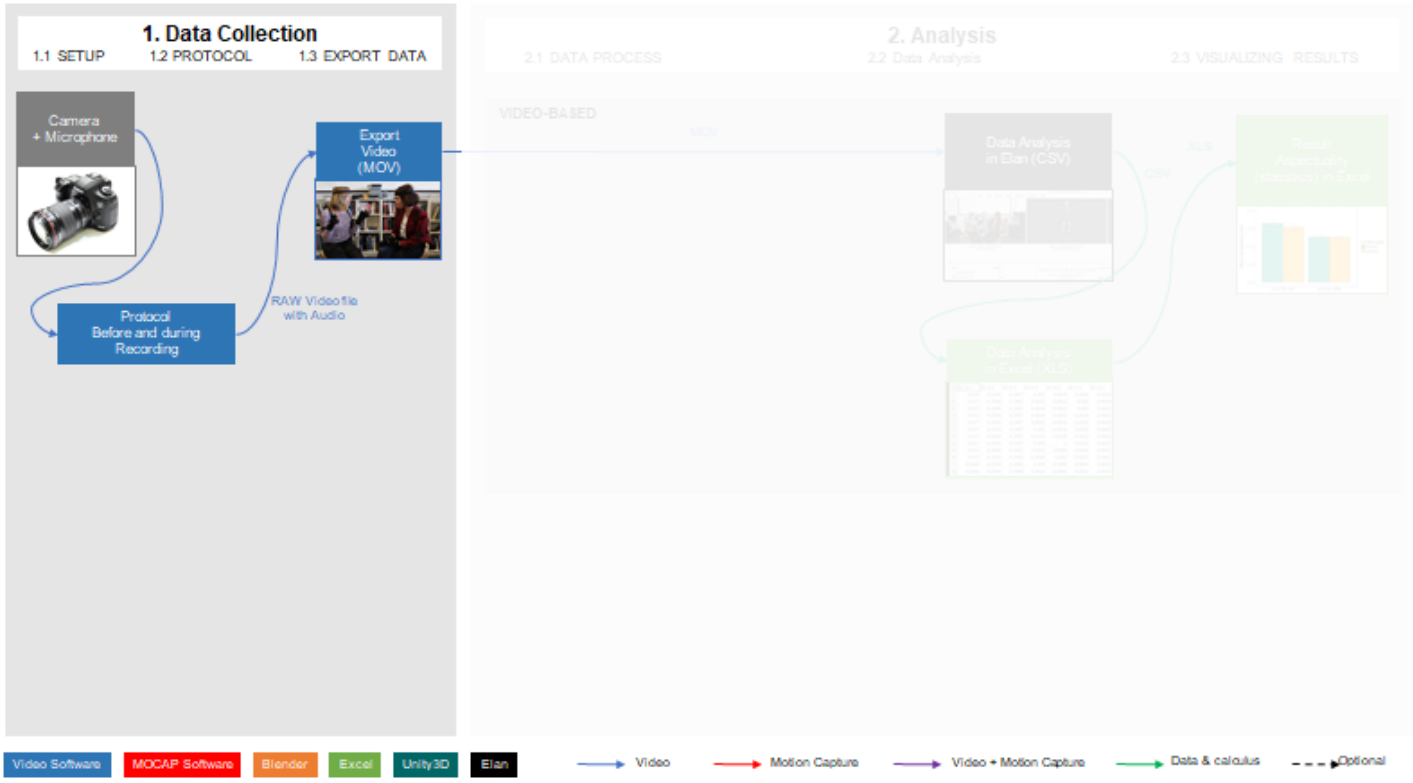


figure: synchronizing audio/video with mocap

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

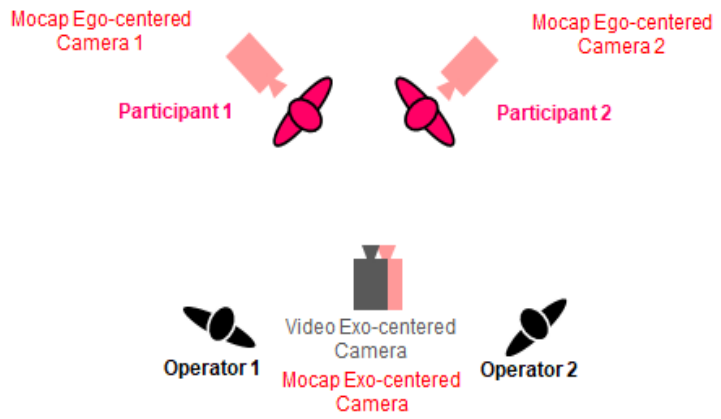


figure: data collection of two participants using video camera (in gray) and motion capture "virtual" cameras (in red)

1.2.A Before the recording

- The material (table, chairs, background)
- The clothes for the subjects and their personal environment (no black cloth, no black background, solid background…)
- General Counseling about the equipment of the suit (strap placement)
- The wearing of the device, i.e. exact placement of the sensors on the body, tightness of the straps (not too loose neither too tight).
- What you have to know about the subjects (sex, height)
- Calibration phase (How to make it, where to do it, locking of the positions)
- Naming (Protocol, height)

1.2.B During the recording

- Sync with the video (first claps at the beginning and a second at the end of the recordings)
- First Evaluation for the accuracy of the recordings

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.

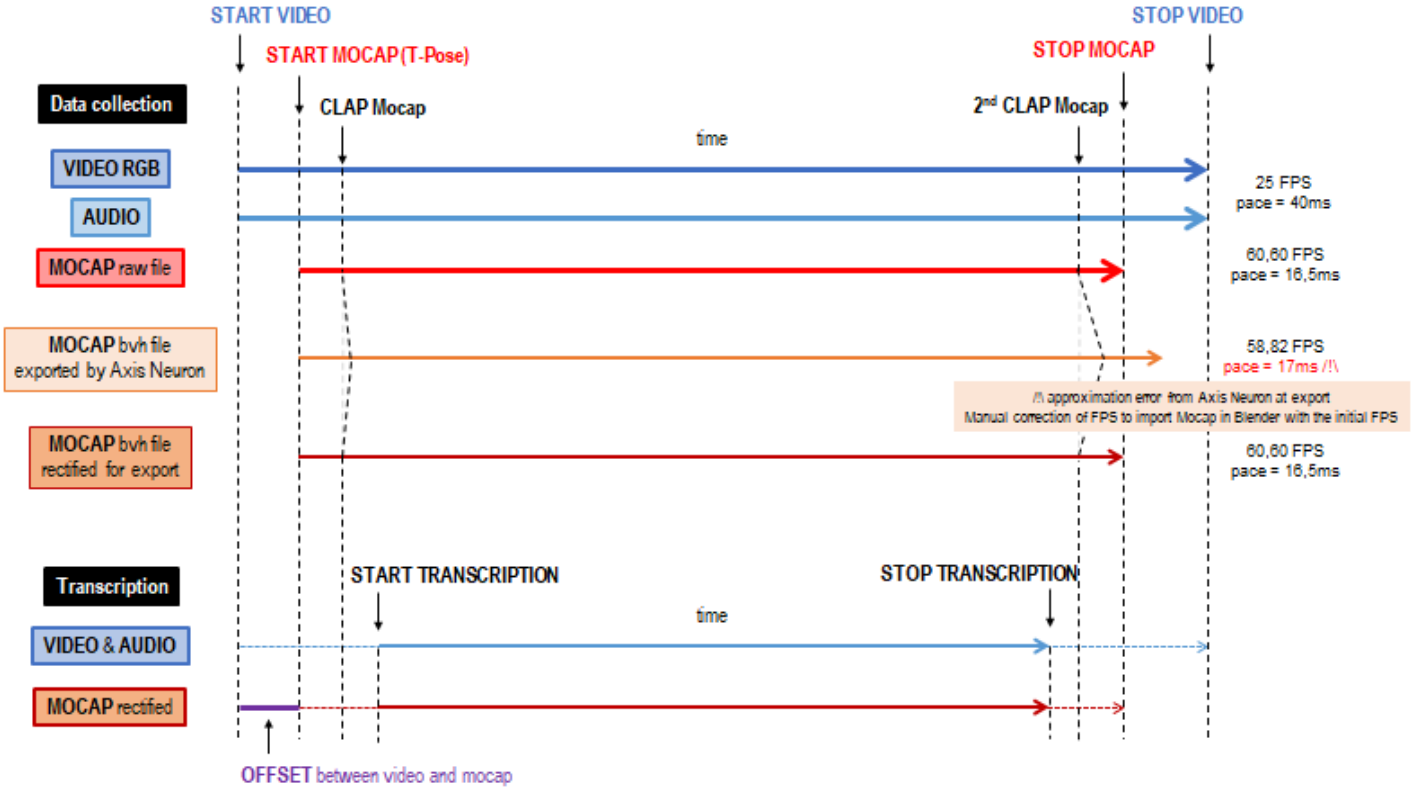


figure: synchronizing timelines with posture, claps and explanation of data asynchrony

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.

1.3.A Filetypes

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.

1.3.B Filenames

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.

- Video Recording **VR**
  (VR1 VR2 if multiple cameras)
- Sound Recording **SR**
- Mocap Capture recording **MC**
  (also add a suffix the heights of the subject ex. MC-FRFR01-1m66)
- Video Motion capture **VM**
  (VM1 VM2 if multiple points of view)

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. Then we proceed to the **Data Analysis** itself. 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.

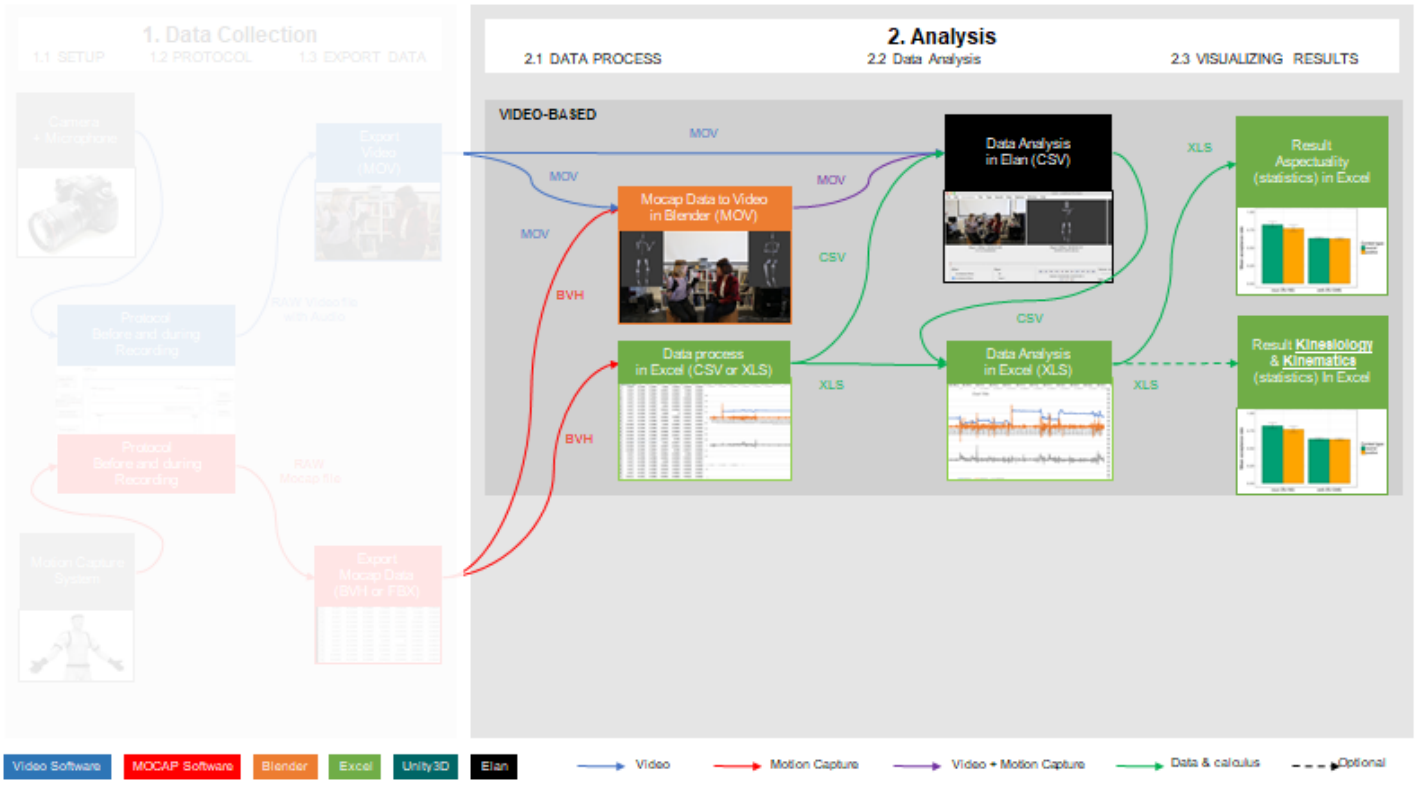


figure: part 2. Analysis

2.1 Data Process

The data process is dependent on the Data export and is necessary to prepare the Data analysis. 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.

2.1.A Extracting Data: Save the raw file into .bvh

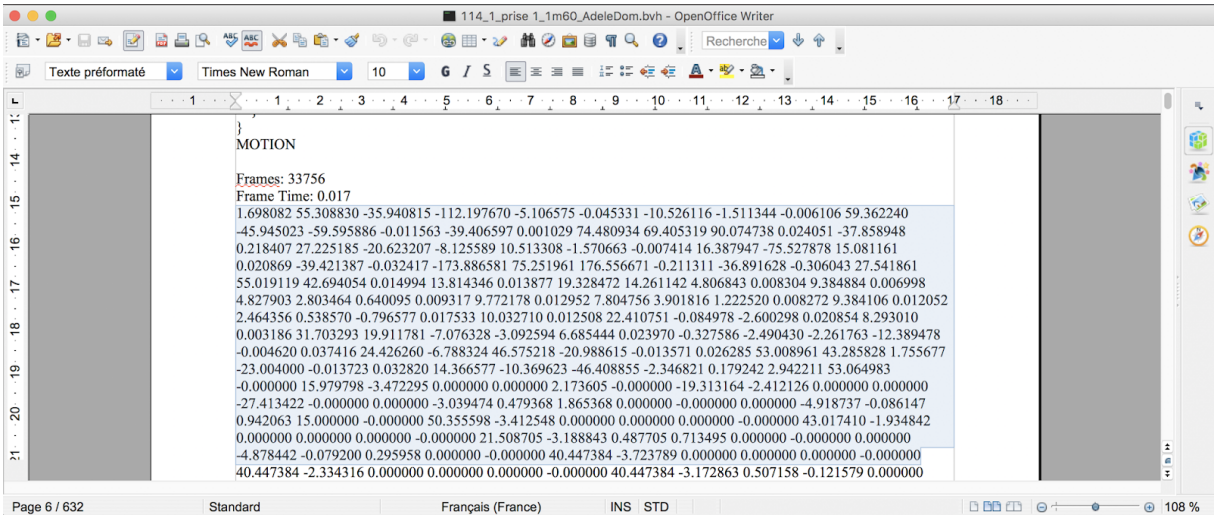
- Open the raw file in Axis Neuron (File, Open)
- File > Export ; filetype: Biovision BVH
- Select rotation order Rotation: **YXZ** ; Tick the box “Displacement”

Data information in Excel sheet format

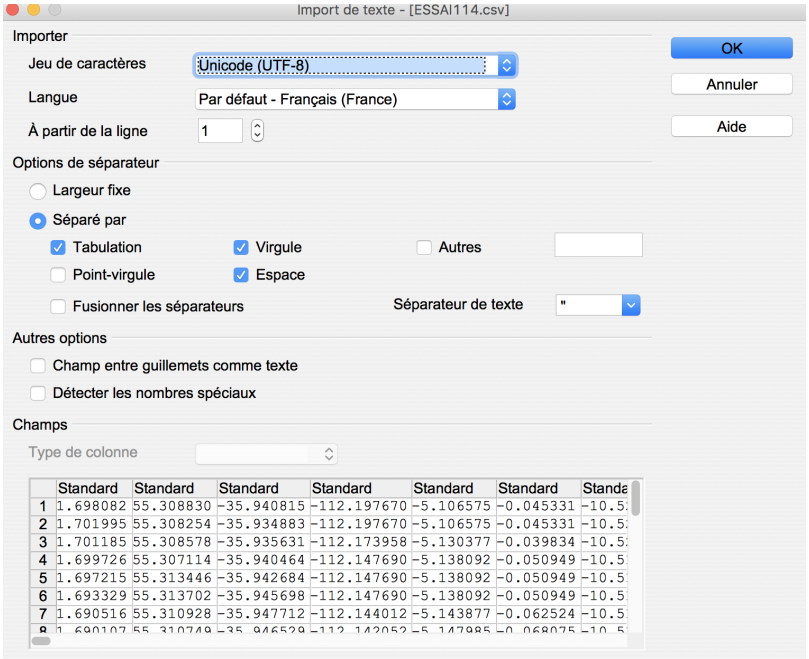
- Open the BVH with OpenOffice (it could be long). You will have this kind of text :

```
HIERARCHY
ROOT Hips
{
   OFFSET 0.000 85.175 0.000
   CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
   JOINT RightUpLeg
   {
      OFFSET -10.500 -1.535 0.000
      CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
      JOINT RightLeg
      {
         OFFSET 0.000 -39.400 0.000
         CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
         JOINT RightFoot
         {
            OFFSET 0.000 -37.320 0.000
            CHANNELS 6 Xposition Yposition Zposition Yrotation Xrotation Zrotation
            End Site
```

- Select the data (more or less at the 6th pages), beginning at the end of the hierarchy information and ending to the end of the file.

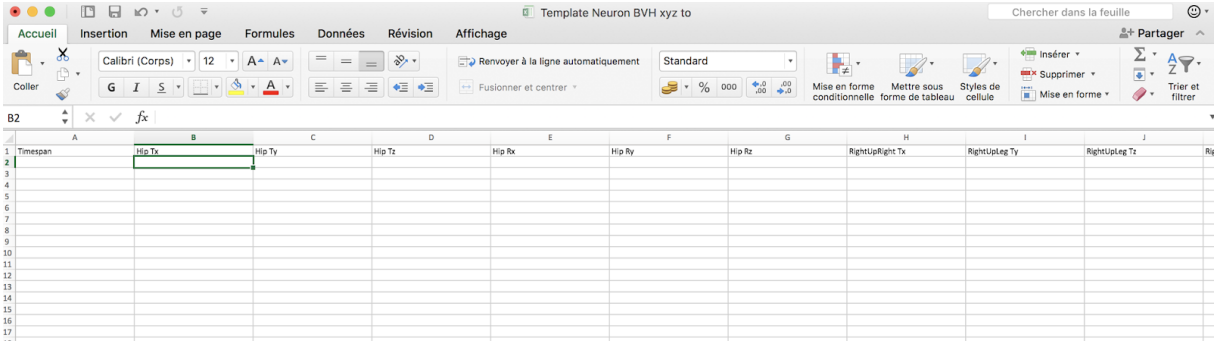


- Copy and paste this data in a new OpenOffice file (.TXT) and change the extension of this file into ".csv".
- Open this .csv file into Open office. You should have this first window:

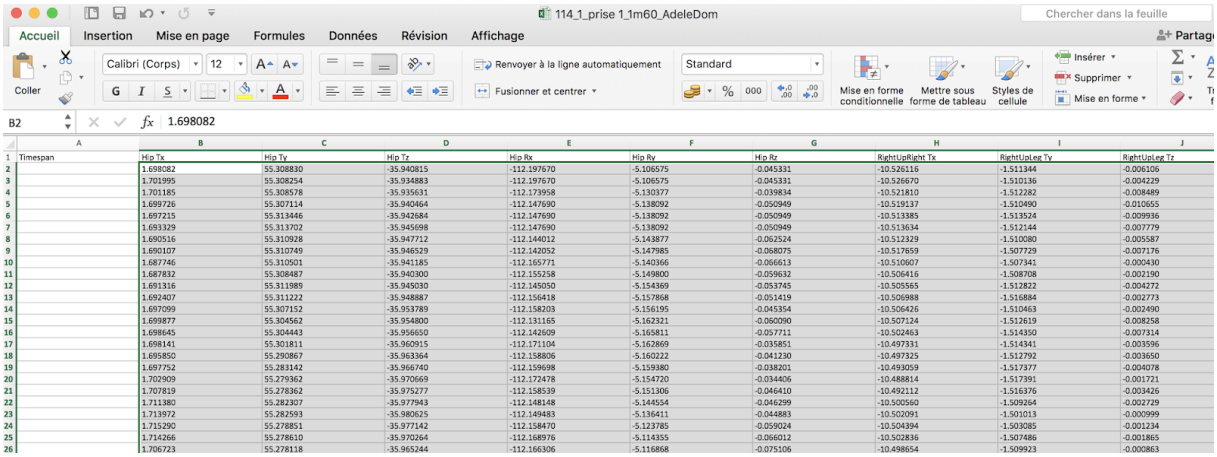


- Do not forget to save as .xlsx (or .xls). Close it. You can re-open it from Excel.

- Open the File “Template Neuron BVH yxz” which contains the right naming for each column. This template is built for 6 dimensions per sensor.



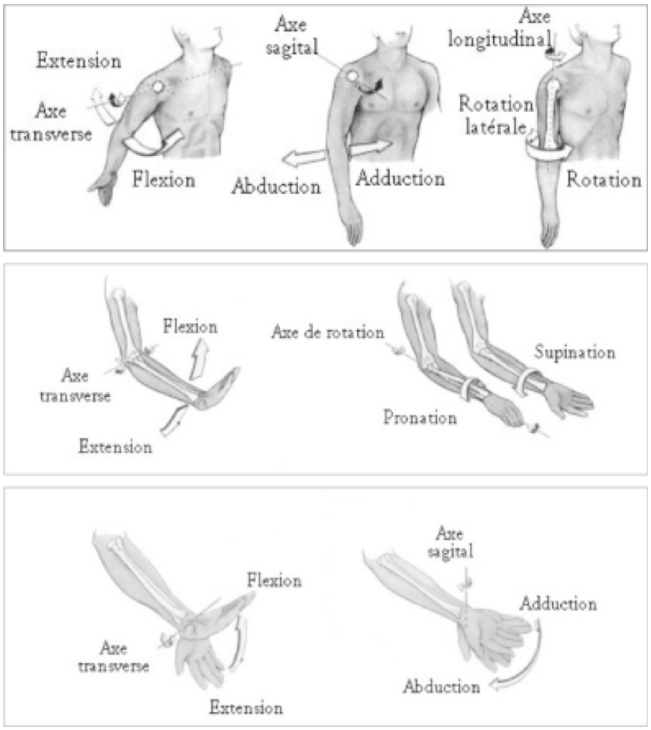
- Paste the content of the file .xlsx (or .xls) you just save in this template file.



- Now you have the data in the right name for each column. You only need to generate the timespan. The time of the first frame is 00, the second 17ms and so on. You can write the first 5 timespan : 0 - 17 - 34 - 51 - 68. Select this first five lines and double-click on the little Cross (right of the last Cell).

2.1.B Explanation about the kinesiological attribution and the orientation of the XYZ axis

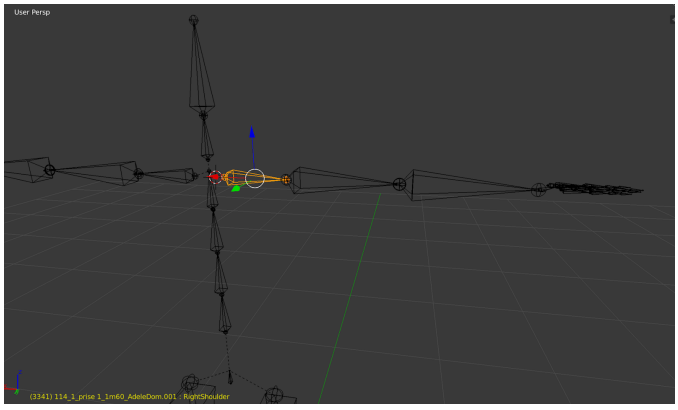
The first step is the naming of the kinesiological movement.
In the figure just below the top bloc figures the Arm Movement (Flexion/Extension, Abduction/Adduction and Interior/Exterior Rotation). The middle bloc concerns the forearm (Flexion/Extension and Pronation/Supination). The bottom bloc figures the movements of the Hand (Flexion/Extension and the Abduction/Adduction).
Each Movement is turning around a joint. For instance, the Flexion/Extension of the arm is due to a circular movement around this transverse Axis, as you can see (top bloc).



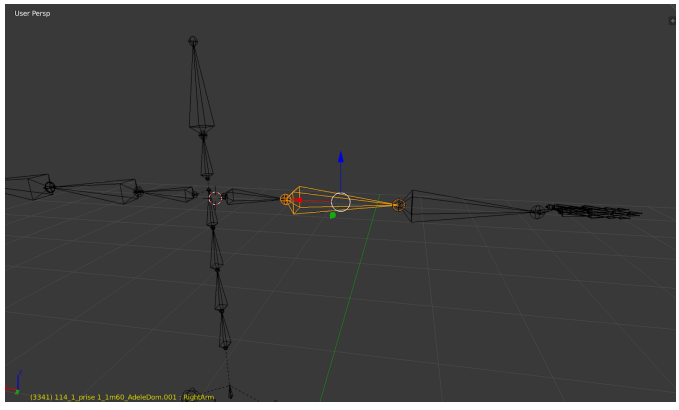
The axis is not always perpendicular to a joint. Some of them are along bones. For the rotation of the arm, the axis is along the humerus. Another movement has to the same feature: the pronation/supination. The axis is along the radius and the ulna.

- How can we label the right movement in a XYZ system of coordinates?
- In T-Pose, we can determine the orientation of the XYZ axis for each segment. These are the Orientation of the Shoulder, the Arm, the Forearm and the Hand.

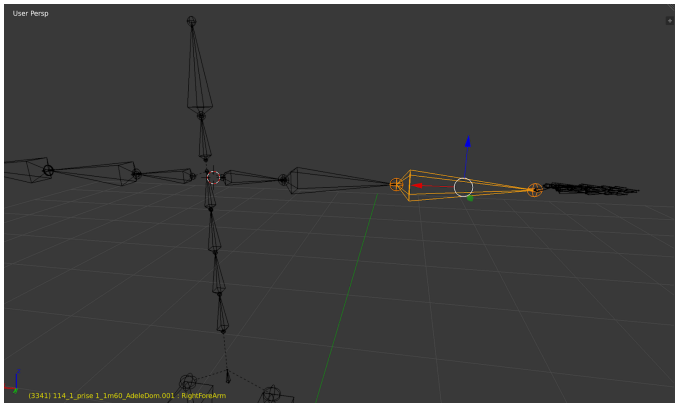
The RED ARROW represents the X Axis
The GREEN ARROW represents the Y Axis
The BLUE ARROW represents the Z Axis.



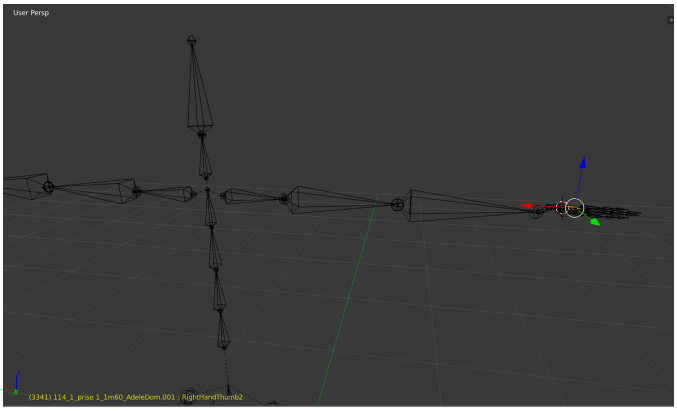
The right shoulder in Blender



The right arm in Blender



The right forearm in Blender



The right hand in Blender

This is a table which sums up the correspondences between the kinesiological naming and the XYZ coordinate system. For each Movement, the first pole is positive in the Axis, the second corresponds to negative values.

XYZ coordinate system	**Right Shoulder**	**Right Arm**	**Right Forearm**	**Right Hand**
X axis		Exterior/Int Rot.	Pro/Supi	
Y axis	Add/Abd	Add/Abd	Flex/Exten	Flex/Exten
Z axis		Flex/Exten		Abd/Add

For the Left upper limb, the values are in the other way around. It is Exterior/Int Rot. instead of Interior/Ext Rot. for the right upper limb.

For the arm sensor, we have the X axis along the arm, the Y axis is towards the back and the Z axis is towards the left
This means: the rotation axis X corresponds to the rotation of the arm, the rotation axis Y corresponds to the abduction/adduction and the rotation axis Z corresponds to the Flexion/Extension

For the forearm the X axis is along the forearm, Y is next to Ulna and Z pierces the forearm perpendicularly.
This means: the rotation axis X corresponds to the prognostication, the rotation axis Y corresponds to the extension bending and the rotation axis Z corresponds to the abduction/adduction

2.1.C Merging Data (video and mocap) BVH Data to Video (Blender)

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 explained previously.
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 Unity3D 3D to Elan software for annotation (table 2)

We use this software:

- Axis 3.8.42.6503 - calculation engine 3.2.2.7251
- Blender version 2.78c

1. **Preparing the BVH for Blender**

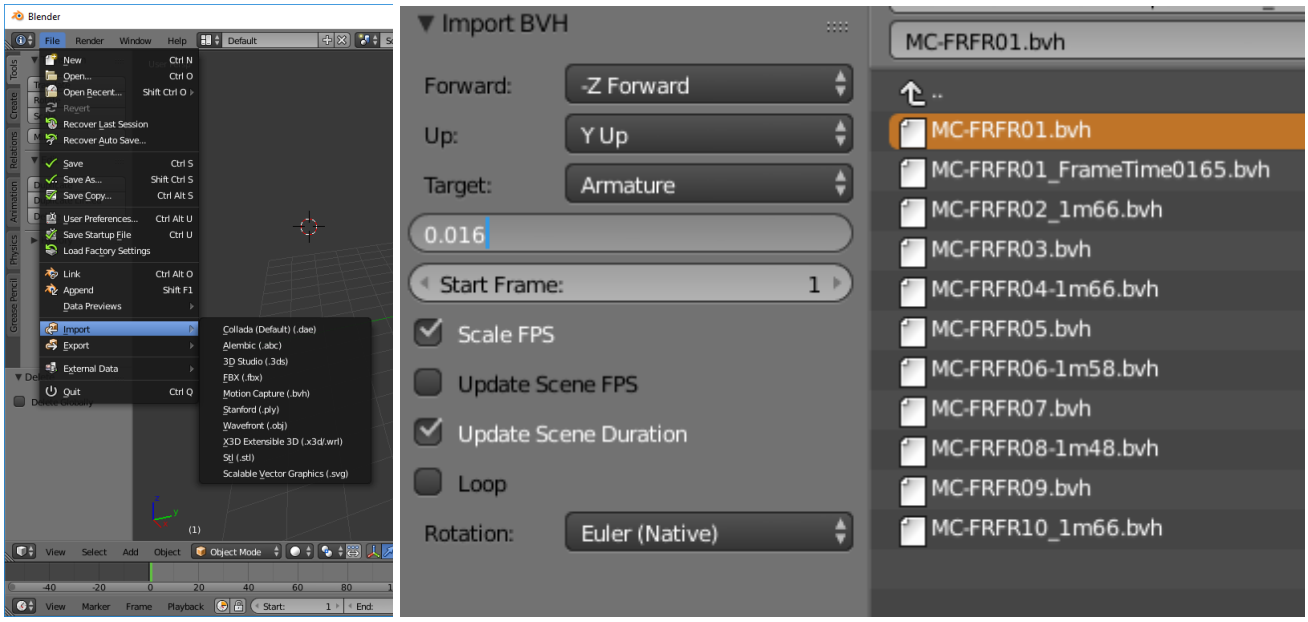
IMPORTANT: when exporting a BVH file from Axis Neuron, the default timeframe is 0.017. However it should be 0.0165. The first thing to do is to edit the BVH file using a text editor (or OpenOffice).

2. **Create a New scene**

File> Create a New scene

WARNING: first set the scene frame rate to 60 FPS (right panel: Properties > Dimensions > Framerate select 60)

File > Import BVH

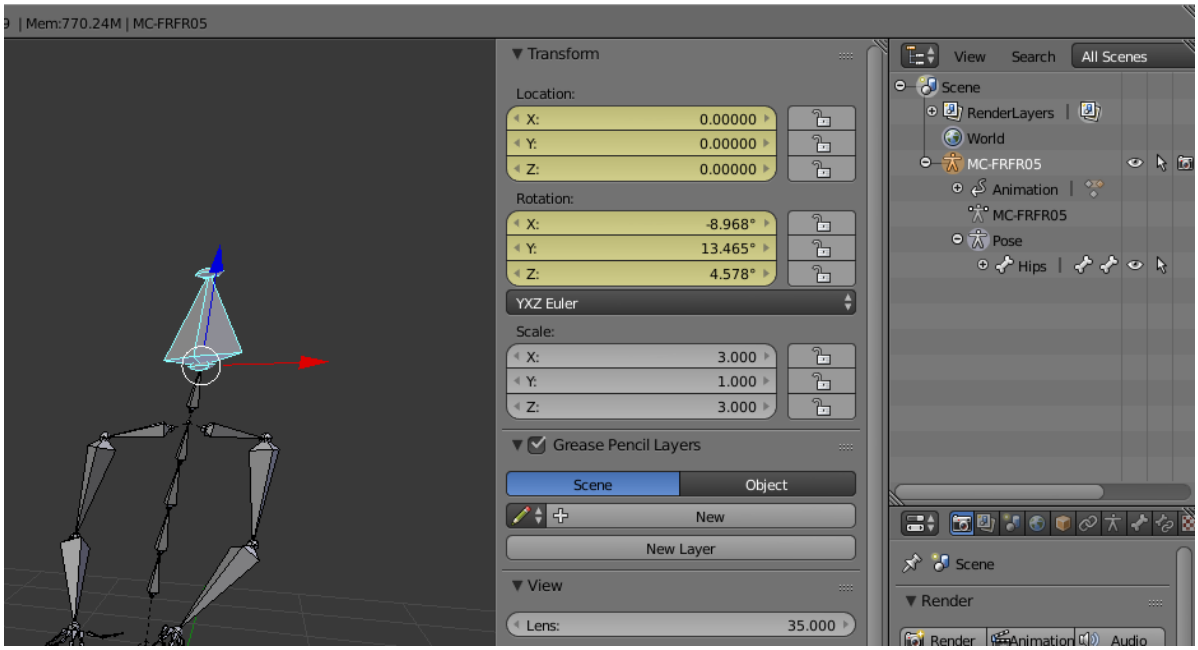


Select the file
On the left panel:

- Tick Scale FPS
- **IMPORTANT Don't** tick Update Scene FPS
- Tick Update Scene Duration
- Change the scale from 1 to 0.016 (means the height of the subject is 1.60m)

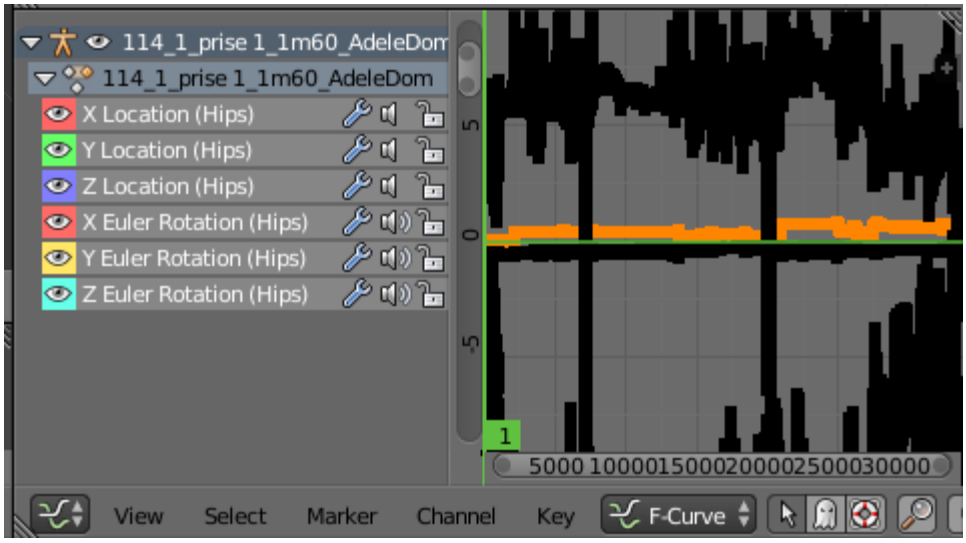
Optional: Scaling the head for better visibility

On the hierarchy menu (right panel): select Pose
One the viewport: Select the head (right click)
On the transform panel set the scale values of X and Z axis to 3.0



Optional: Customizing animation

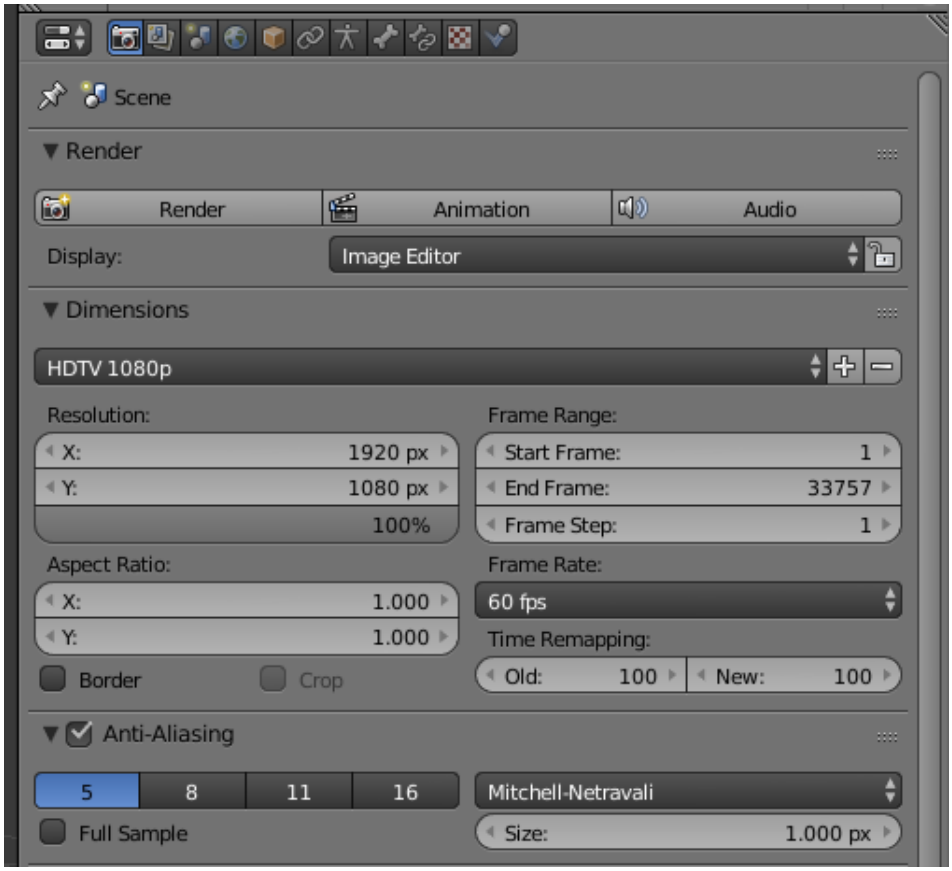
- On the **hierarchy**, select the **hip**
- on the **Graph Editor**, **uncheck the speaker icon** in order not to play the Hips XYZ locations (i.e. positions) animation



Rendering animation

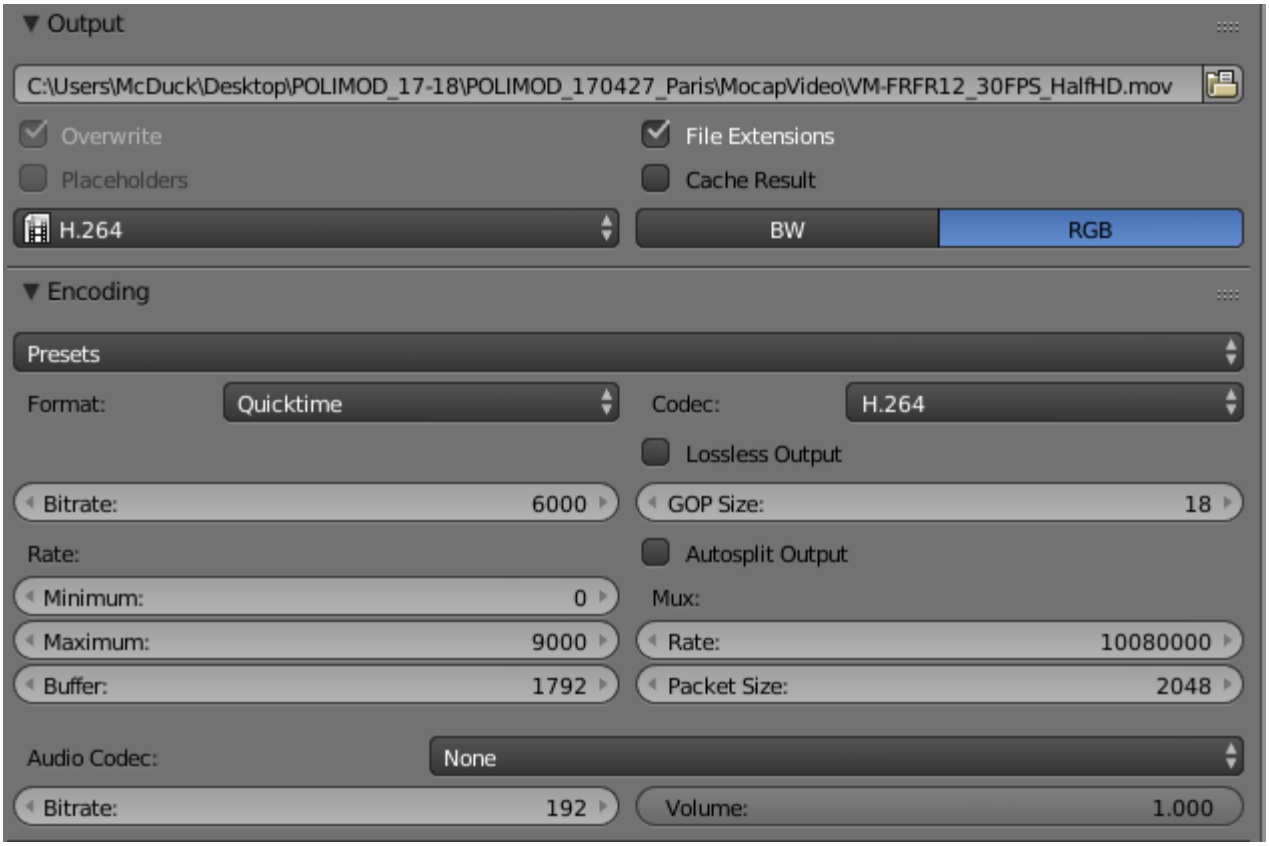
Check the resolution

- Preset HDTV 1080p can be reduced to a lower resolution: 720p
- Frame rate to 60fps (smoother animation) can be reduced to 30 editing the framestep to 2
- Antialiasing to 5 or 8 (better quality), can be unchecked creating lower edge quality with pixels

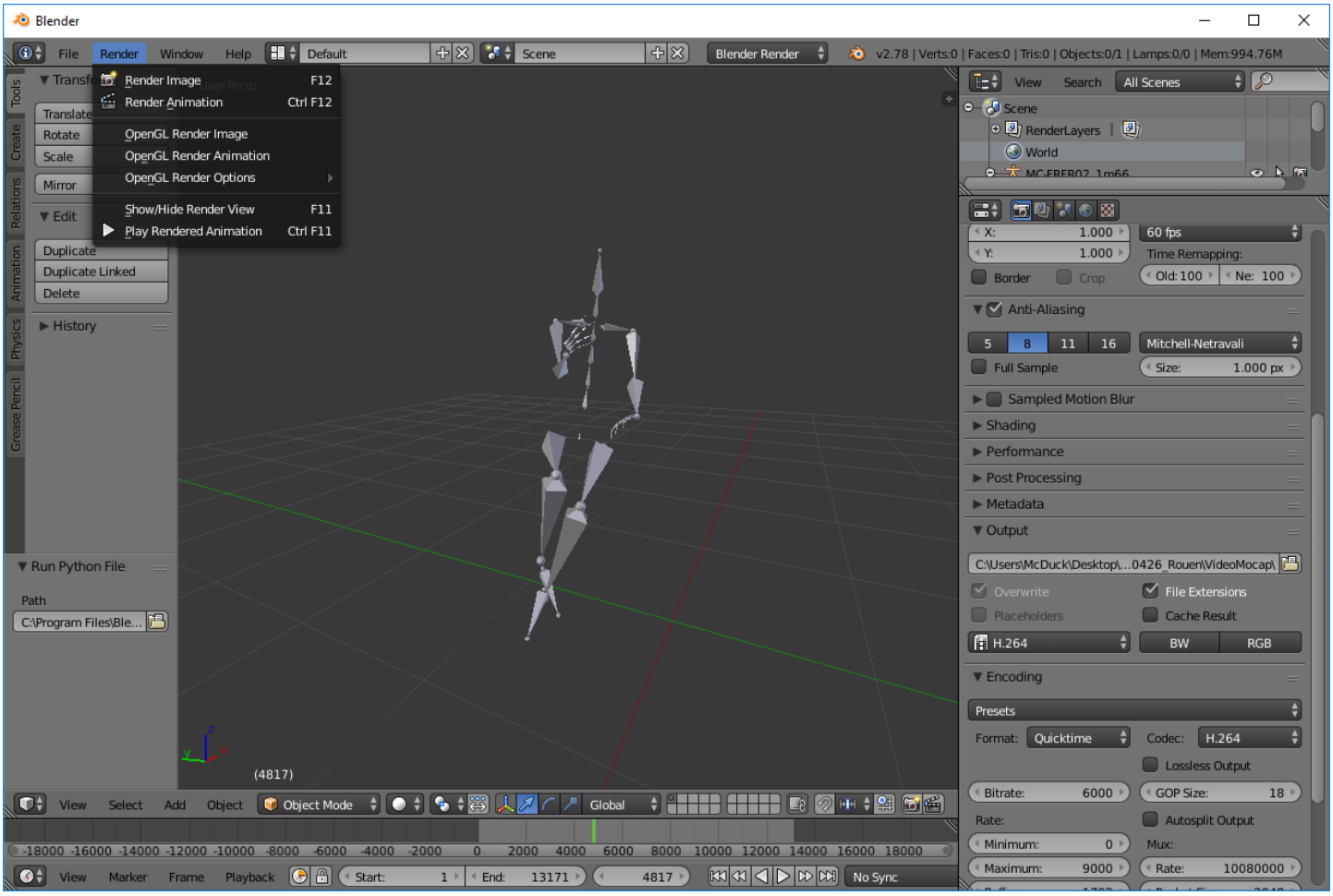


output

- format QuickTime
- h.264



Render the video



1. First, check the virtual body fits the image size

top left menu: **Render > OpenGL Render image**

If not, adjust the view of the viewport

2. if it's correct, launch the video rendering

top left menu: **Render > OpenGL Render Animation**

IMPORTANT: in windows, do not reduce or interact with the blender window during the rendering, it makes Blender crash

Creating a video with MOCAP

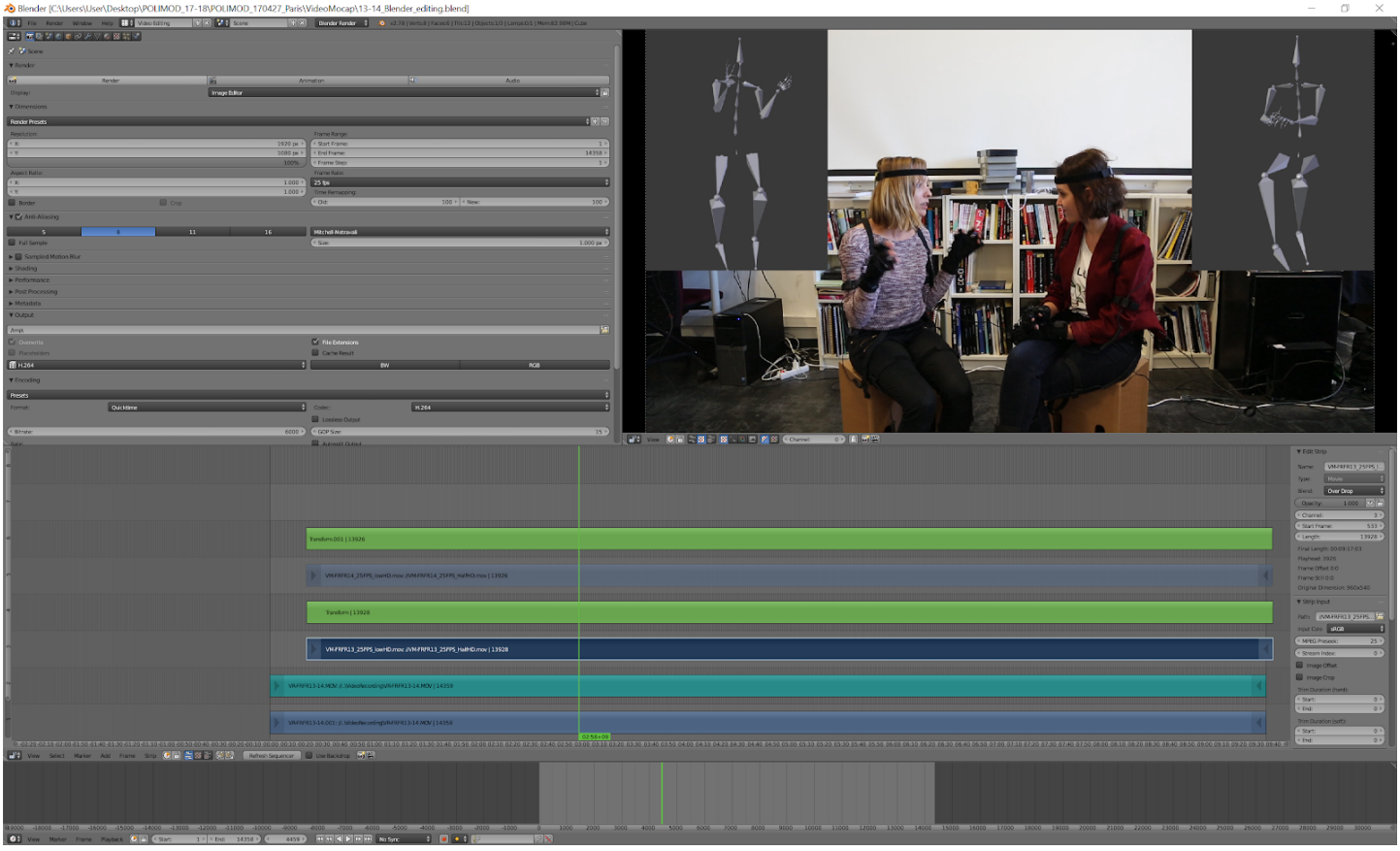
Open **Blender**

Change Default Layout to Video Editing

- On the menu on top of the timeline, go to:

Add > Movie > Select the RGB Video from the camera

then Add the 2 Mocap Video rendered previously:
Add > Movie > Select the RGB Video from the camera



2.1.D Mocap Video and Unity

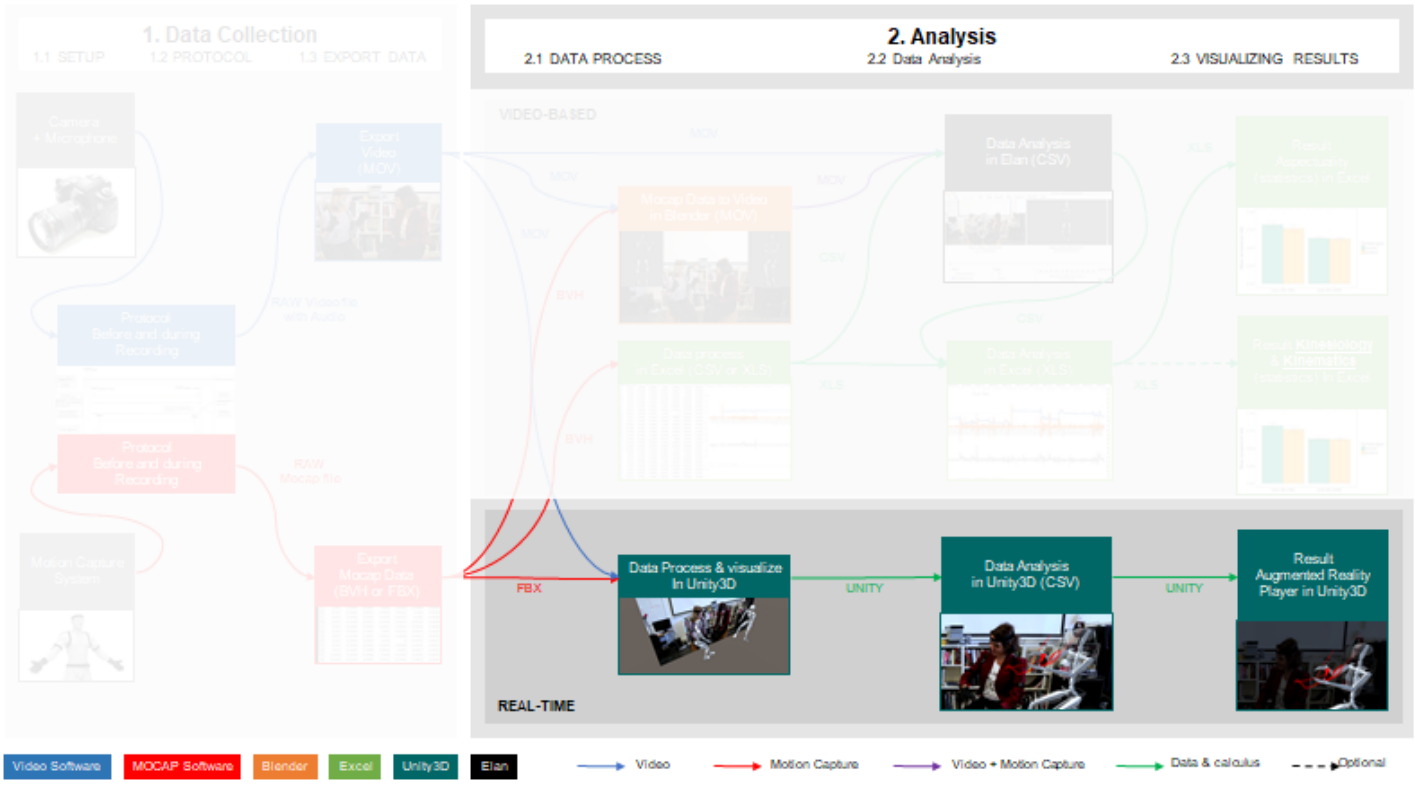


figure: Import Mocap and Elan in Unity

We propose a tool to import and merge in Unity3D Audio/video with Elan annotation and Mocap

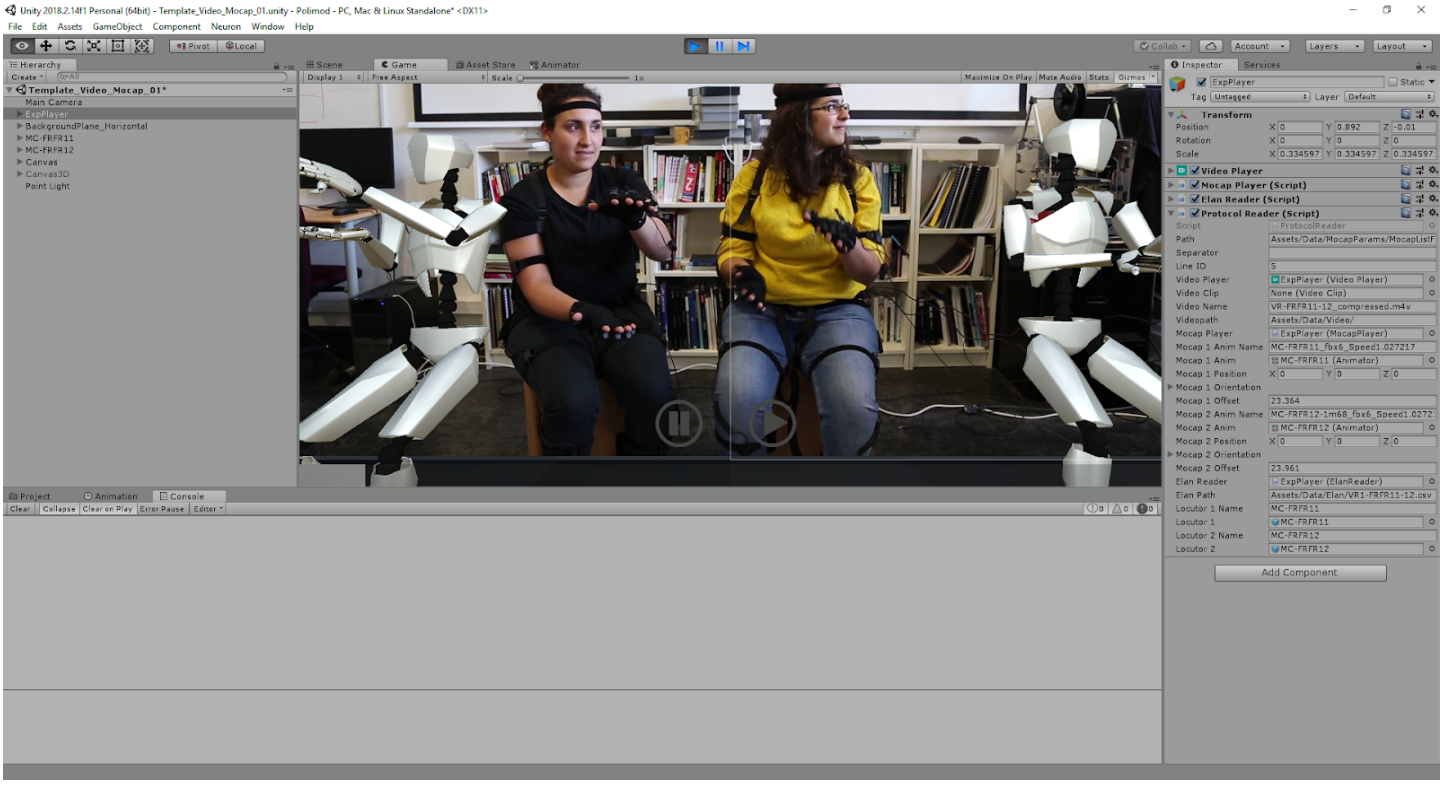
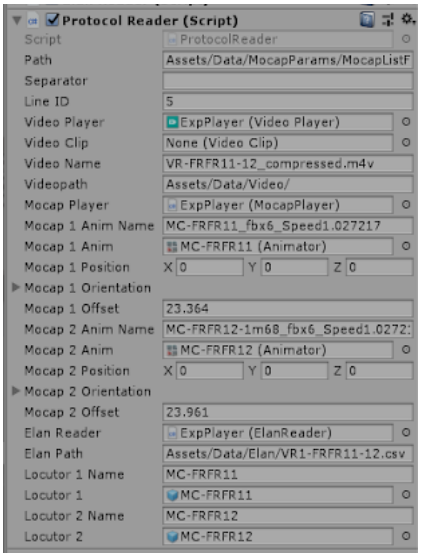


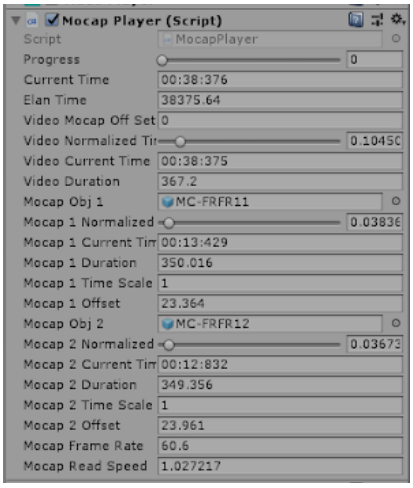
figure: Unity3D Interface

Script description:



Protocol reader:
This script connects all the elements (video, Motion Capture and Elan coding) for a given experimentation it also configures all the other scripts

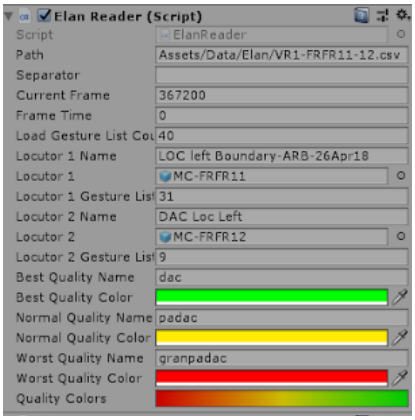
The user just has to choose the number of experiments and click on play button



Mocap Player: (Managed by Protocol reader)

This script synchronizes the video of the experimentation and the different mocap recording at the same time base

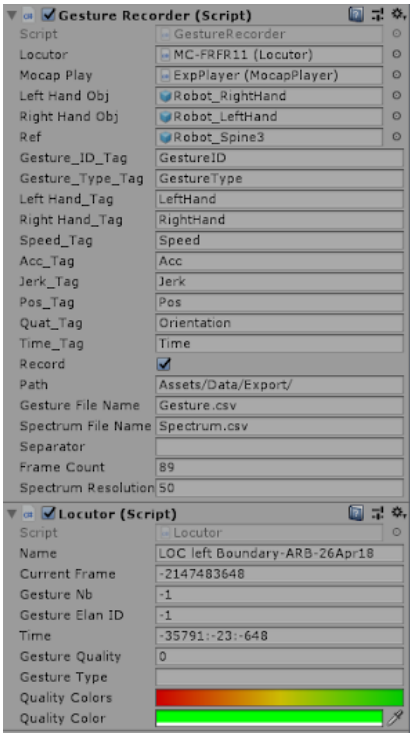
It also gives the global control of the progress of the experimentation and allows to choose different key points for a contextualized gesture analysis



Elan Reader: (Managed by Protocol reader)

This script reads the exported coding file from Elan and transcodes the information to a locutor script specific to each speaker

It also transcodes the consensus information (DAC, PADAC, GRANPADAC) in an indexed value on a color gradient.



Gesture recorder: (Managed by Protocol reader)

This script retrieves the gestural information (Velocity, Acceleration, Jerk) for both hands of a speaker and writes them both to a specific and a global file in order to be able to do the statistical analysis

Locutor: (Managed by Elan reader)

This script triggers gesture on global time and gives information specific to the identification of each gesture coded with Elan

Motion view: (Managed by Gesture Recorder)

This script calculates and filters the different information for the kinematic (Velocity, Acceleration, Jerk) analysis of the gesture

It also allows to add colored VFX in order to visualize the information in real time on the avatar hands

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 on each type of analysis, we had to generate specific mocap file exports.

2.2.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. We can then import this video in Elan and annotate gesture with mocap files superimposed in the video.

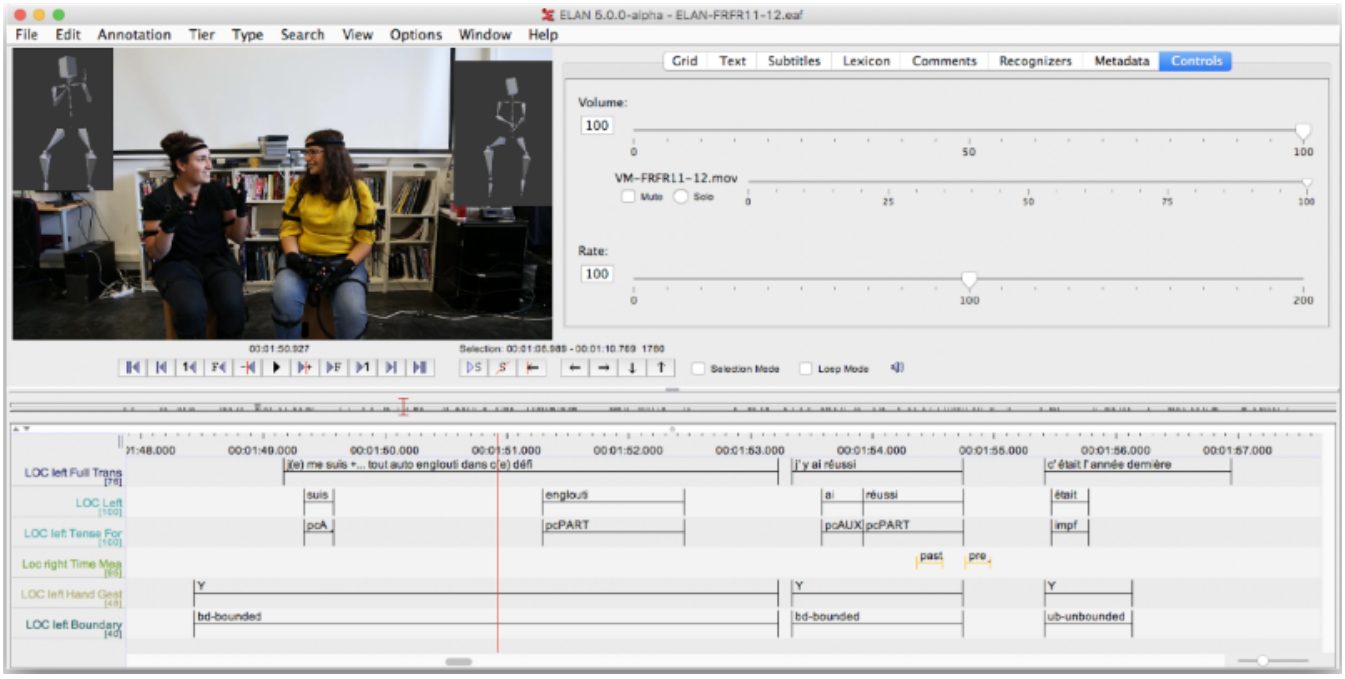


figure: coding bounded and unbounded gesture in Elan with video and mocap rendered picture in picture in the video

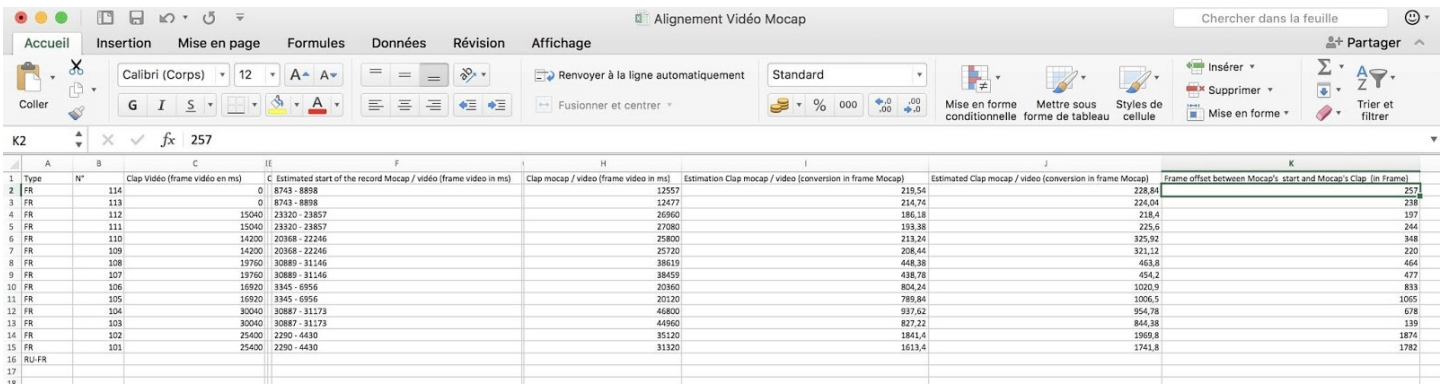
Synchronization in ELAN

ELAN allows to synchronize up to 4 media files (video or audio). So we can synchronize each video file with the video of the two MOCAP recordings. It allows to check visually sequences that we analyze.

Prerequisites

You have to keep in mind that the velocity of the recordings are different between the video (25 frames per second) and the Mocap with Neuron (60 frames per second). To be sure of the synchronization in ELAN, we need to be sure of the time in AXIS NEURON.

Build an excel sheet (proposed name: “Alignement Video-Mocap_Name Project_Month-Year”). The goal is to put together all of the sync information of the start (video, Mocap 1 and 2 and eventually the Audio), and of the claps. Some claps are not visible because the media is not started or because we cannot hear or see it.



Type	N°	Clap Vidéo (frame vidéo en ms)	Estimated start of the record Mocap / vidéo (frame video in ms)	Clap mocap / video (frame video in ms)	Estimation Clap mocap / video (conversion in frame Mocap)	Estimated Clap mocap / video (conversion in frame Mocap)	Frame offset between Mocap's start and Mocap's Clap (in Frame)
FR	114	0	8743 - 8898	12557	219,54	228,84	257
FR	113	0	8743 - 8898	12477	214,74	224,04	238
FR	112	15040	23320 - 23857	26960	186,18	218,4	197
FR	111	15040	23320 - 23857	27080	193,38	225,6	244
FR	110	14200	20368 - 22246	25800	213,24	325,92	348
FR	109	14200	20368 - 22246	25720	208,44	321,12	220
FR	108	19760	30889 - 31146	38619	448,38	463,8	464
FR	107	19760	30889 - 31146	38459	438,78	454,2	477
FR	106	16920	3345 - 6956	20360	804,24	1020,9	833
FR	105	16920	3345 - 6956	20120	789,84	1006,5	1065
FR	104	30040	30887 - 31173	46800	937,62	954,78	678
FR	103	30040	30887 - 31173	44960	827,22	844,38	139
FR	102	25400	2290 - 4430	35120	1841,4	1969,8	1874
FR	101	25400	2290 - 4430	31320	1613,4	1741,8	1782
RU-FR							

Column C : Clap vidéo (frame vidéo en ms)

formula to switch hh:mn:s.ms format in ms format: hh*3,6*10^6+mn*6*10^4+ss*1000+ms

Column F: Estimated start of the record Mocap / video (frame video in ms)
Values find in ELAN transformed in ms. This is an estimation regarding to the noise made by the starting beep of each Mocap recording system (we do not know which beep corresponds to which Mocap recording). We do not know the triggering moment during the noise. We put the two values of the beginning of the beep.

Column H: Clap mocap / video (frame video in ms)
These values are extracted from the video put in ELAN, as well. If the clap of a locutor is crushed during several frames, then we take into account the first frame in which the hand is stabilized.

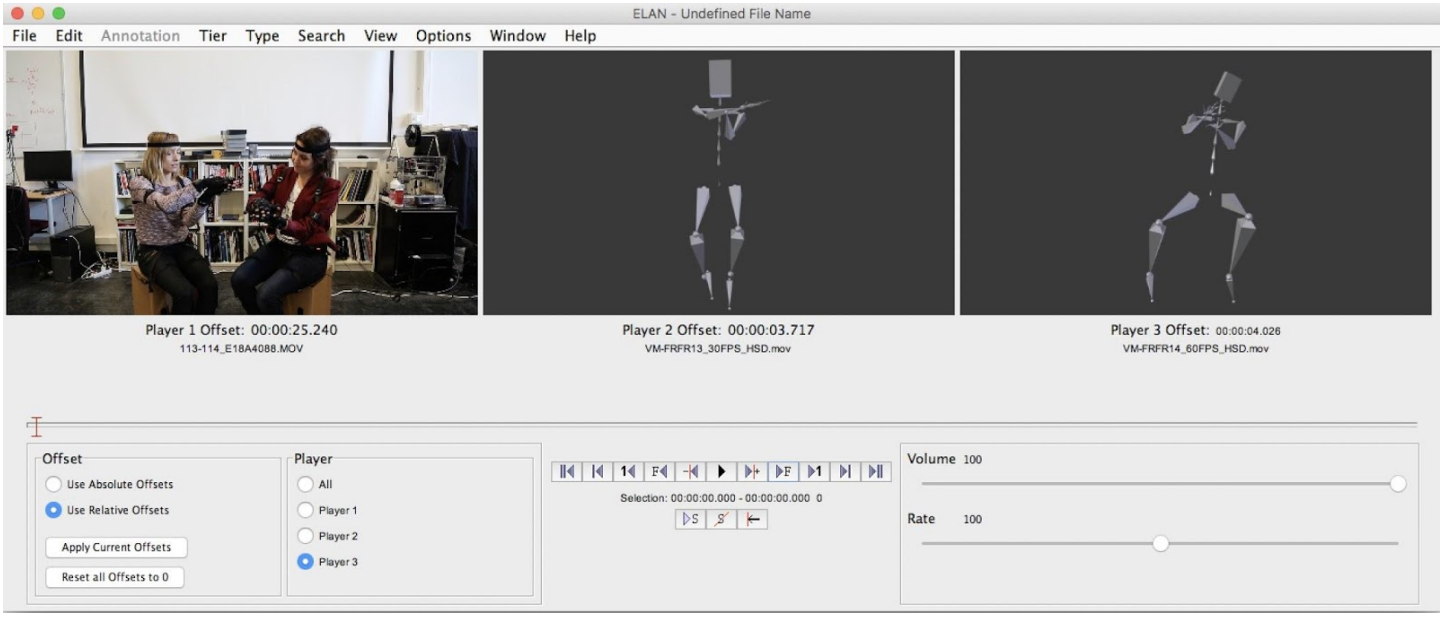
Column I: Estimation Clap mocap / video (conversion in frame Mocap)
Formula : =(H$2-highest value of F$2)/(1000/60) — "(1000/60)" to calculate the exact duration for each frame in a 60f/s system.

Column J: Estimated Clap mocap / video (conversion in frame Mocap)
Formula : =(H$2-lowest value of F$2)/(1000/60). These two columns have no accuracy. They are useful to approximate the value of the last column.

Column K: Frame offset between Mocap's start and Mocap's Clap (in Frame)
In Axis neuron, load the file.raw and search the exact moment of the Mocap clap for each locutor. The value is given in frame.

Now we have checked information about the different sync. We are able to synchronize the files in ELAN.

ELAN sync
Charge the video and the video of the Mocap in ELAN (New > Add media files). Once these video are loaded, go to the "Media Synchronization Mode" (in the tab Options). FIRST of all select the button "Use Relative Offsets".



For each video (Player 1, 2 and 3) select the same moment (the clap of the latest locutor). Sync the video Mocap. Then click on "All" and re-select the "Annotation Mode". Register the file as such with your naming protocol. Close and check if everything is OK. Be careful, Elan crops the video up to the shortest one.

2.2.B Pipeline to approach the Flow in Excel

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 ;

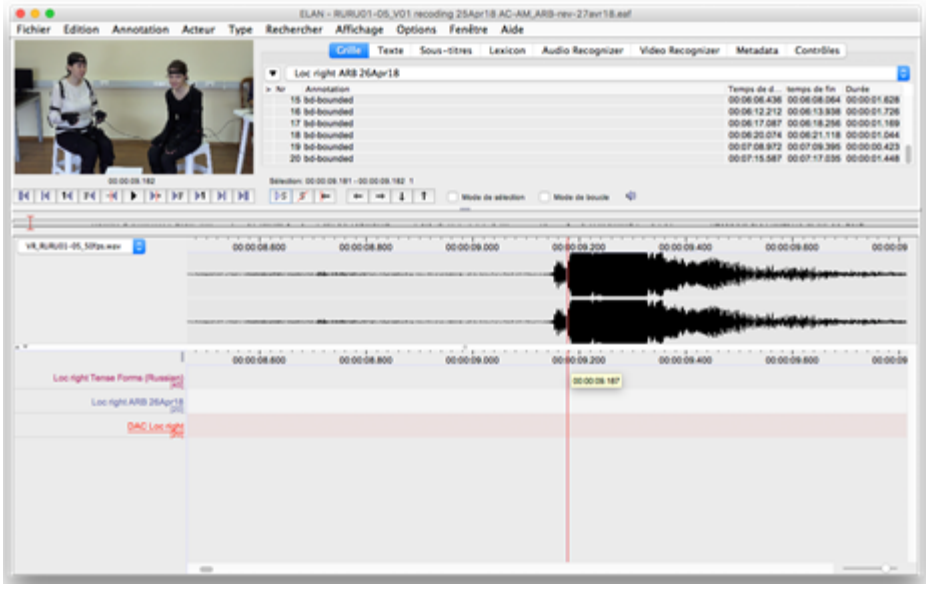
2/ mixing ratio between the degrees of freedom (dof) ;

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 next paragraph.

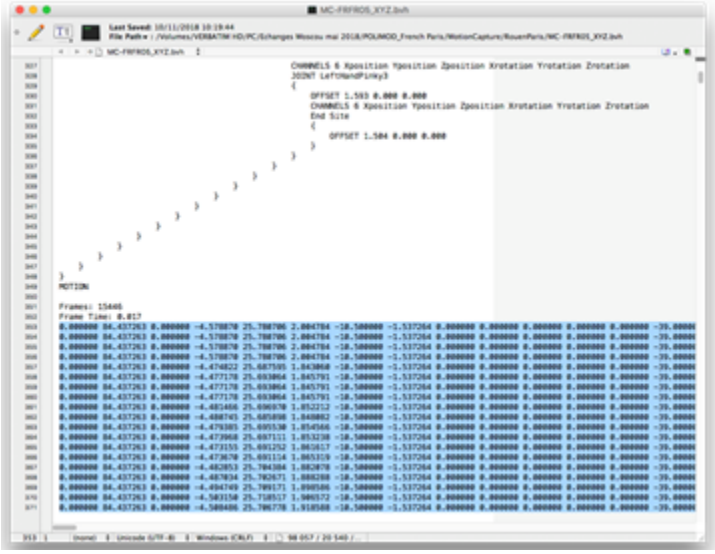
1/ Determine the exact offset between the video and the Mocap data.

Our advice for that is to upload Audacity software in order to “import” the .mov file ant to extract the audio (“export audio” in Audacity with a .wav format) to display it in ELAN into a wave like form (In ELAN: Edit > Linked Files > Choose the .wav file and “Add” it and “Apply” it). Then, you place the cursor in ELAN just before the “beep” of the Mocap recording for each locator and you have the offset in ms (see the offset on the screenshot below: 9182ms).

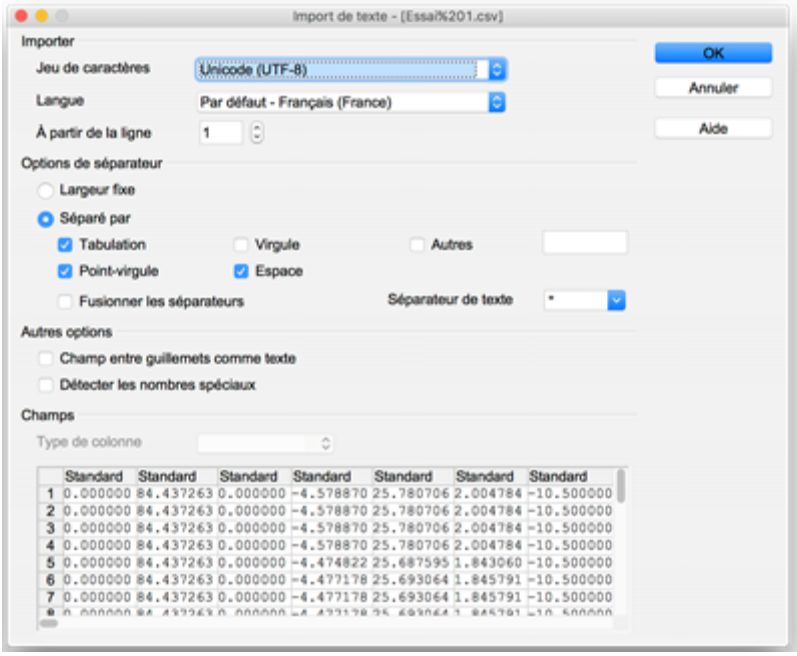


The value of this offset or the nearest value in the excel file giving the timecode is selected until the end of this column. These values will be pasted in the master file where we will add the data issued from ELAN and the mocap, as you will see in the step 6/a/ and following.

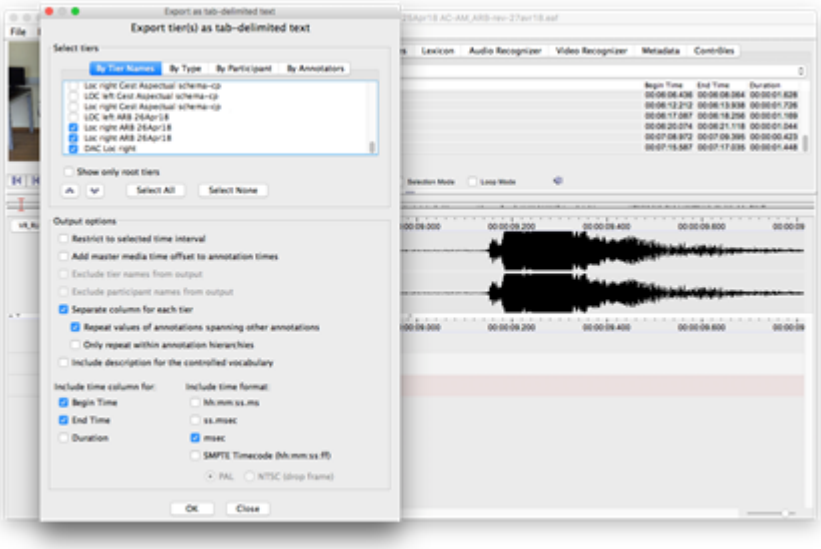
2/ Open the file .bvh coming from the Mocap and exported in XYZ order of rotation with TextWrangler or openedit. Copy Paste the data in a new file and leave the header with the chainings (as you can see below)



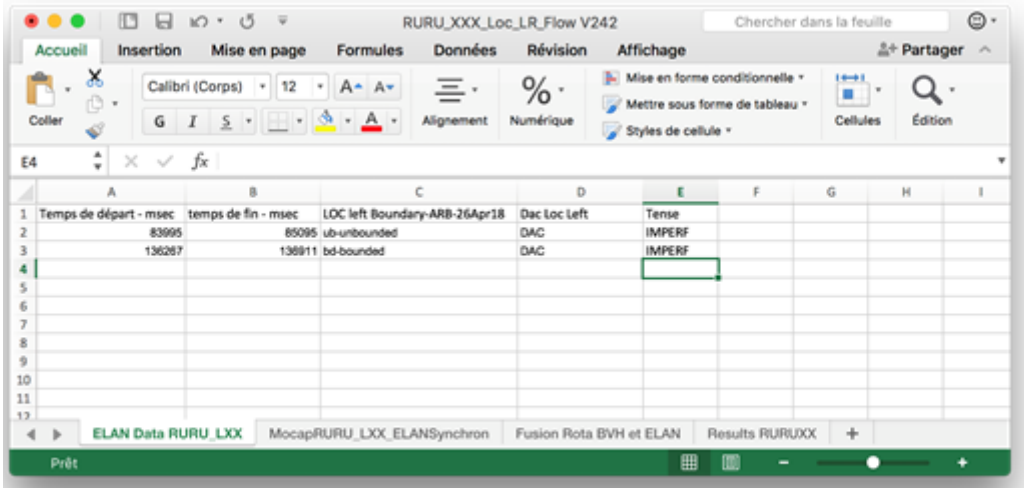
3/ Change the extension of the file from .txt to .csv and open it with OpenOffice. Choose the space as delimiters and save the file in an .xls format in order to open it with excel.



4/ Export the transcription you need from ELAN (File > Export as > Tab delimited Text…). In the pop-up window, select the Tiers you want to export, dispatch them in separate column repeat values of annotation and select the Begin Time, the End time in ms (as you can see below). At this step, e have transformed in Excel format data issues from ELAN, data coming from the Mocap and the Synchronization data.



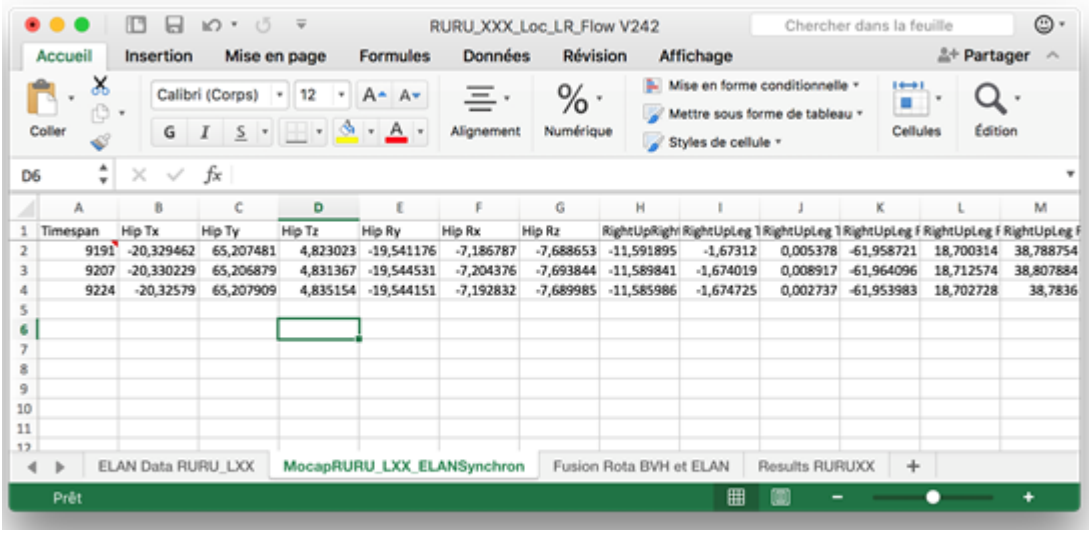
5/ From that point, we are going to put all the data presented in the last step into an excel template (.xltx) whom we detail in the next steps.



6/ The template is divided in 4 tabs.

a/ Tab one “ELAN Data RURU_LXXX” has four columns active and related to the other tabs. Beliw the labels, it is two lines pre-filled on which you can paste the data coming from your ELAN file (see step 1). This data will be reused in the tab 3 “Fusion Rota BVH et ELAN” (Merge Rotation BVH data and data coming from ELAN).

b/ Tab two “MocapRURU_LXX_ELANSynchron”. The two other kind of data is pasted in this tab: the synchronization and the Mocap data. On the column “A”, the timespan is pasted according to the offset determined in step 1/ (be cautious, the values taken from the first excel file contain a formula, you can copy them but paste them with the menu “Paste Special”, choose “value”). In the second column Paste the Mocap value copy from the file created in the steps 2 and 3 (it could be long according to the duration of your recording). All of the data is pasted from the line 2. The first line contains the labels.



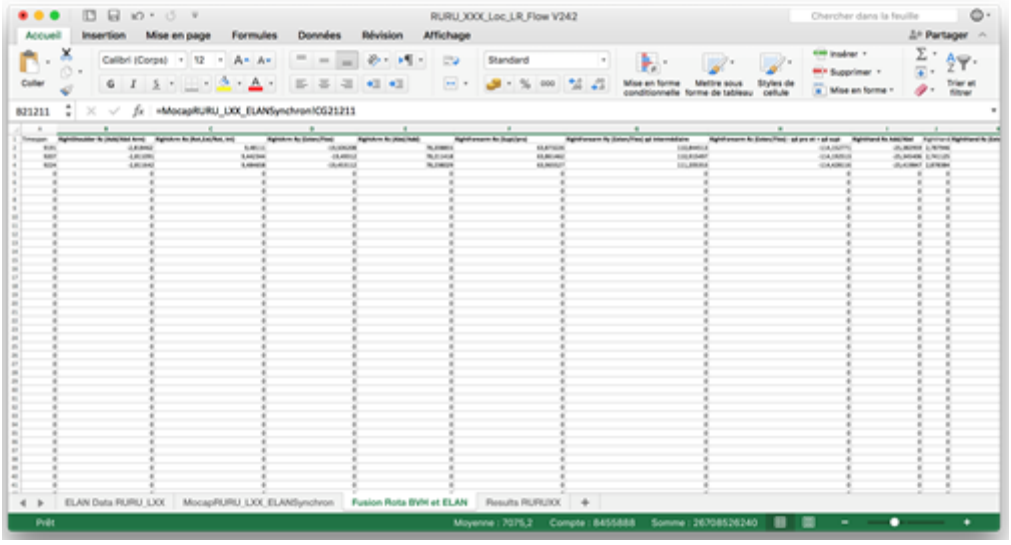
c/ The tab 3 named “Fusion Rota BVH et ELAN” is the core of the computation. Several functions are made automatically in this file:

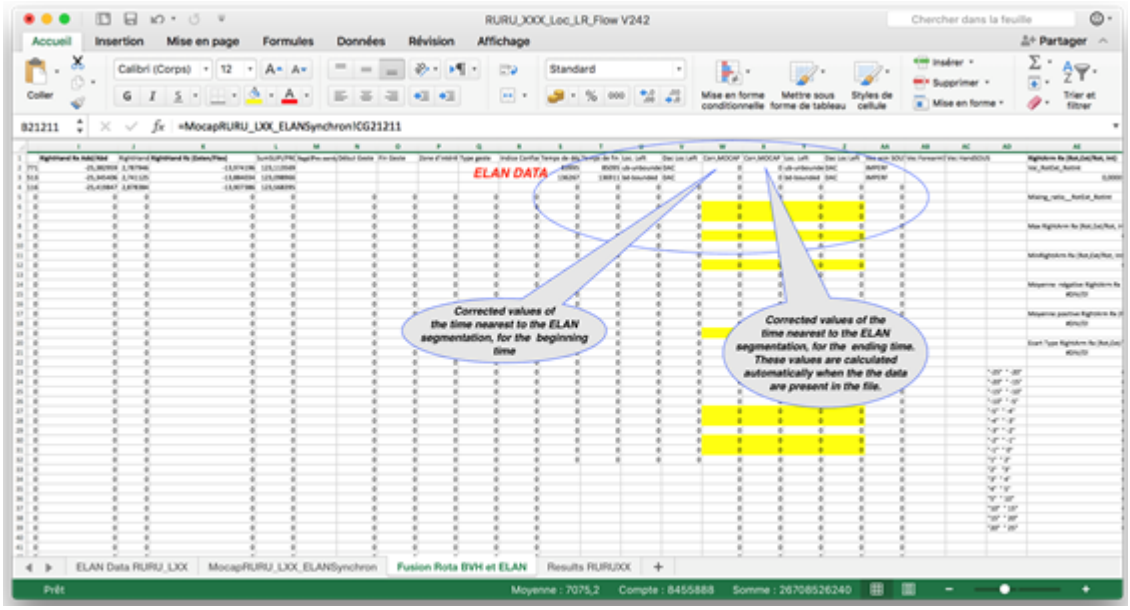
i/ the alignment between the ELAN data and the Mocap Data, extracting just the data during the Gestures segmented in ELAN (first figure below)

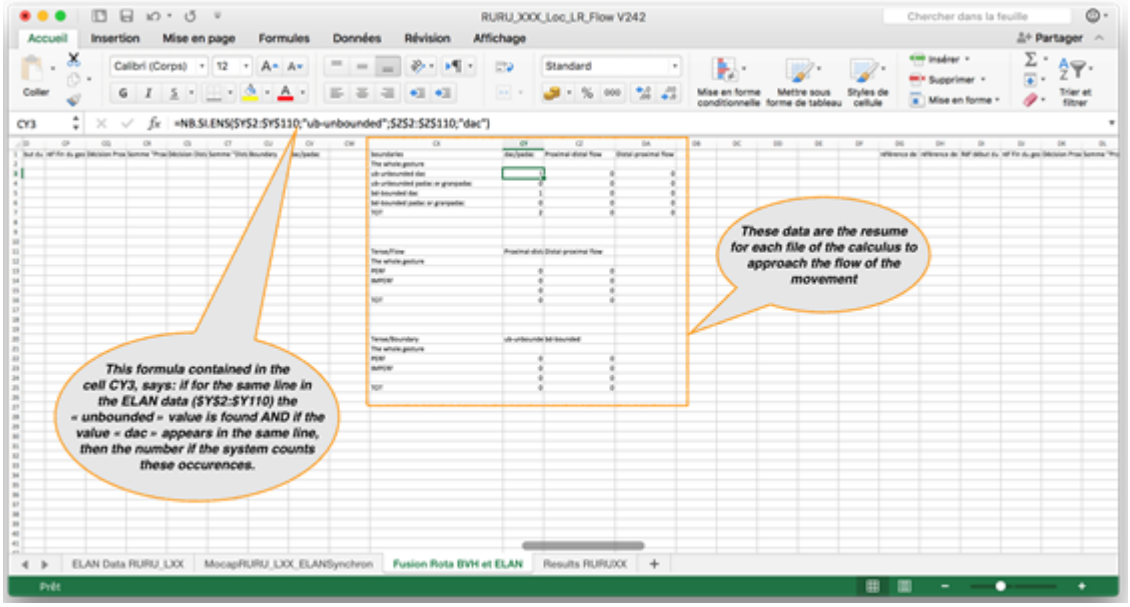
ii/ the selection of the data we need for the analysis: only the rotation values (the values of translation are left out, the values of the right upper limb are interesting there and then copy from the Tab 2 (second figure below).

iii/ Anisgorithm involving the columns AE to EK are required to approach the flow ws th in three ways.

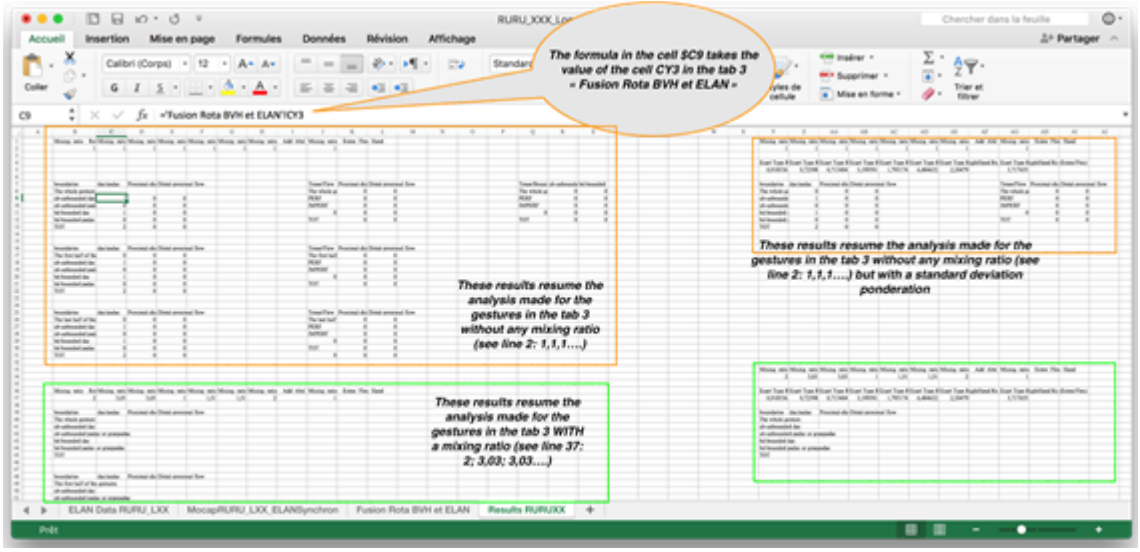
iv/ All the calculation made in this tab are resumed in the sector CK 1 to DA26 (for whole gestures); DR1 to DU26 (for the first half part of the gestures with and without a mixing ratio), EL1 to EP26 (for the last half of the gestures with and without a mixing ratio) and GH1 to GK26 (for the whole gestures with a mixing ratio AND with a ponderation according to the standard deviation of each degree of freedom) (third figure below).







d/The tab 4 is composed of the resume for all the gestures of the file. In other words, the four sectors viewed in 6/c/iv/ are grouped in this tab (see the figure below)



2.2.C Pipeline to determine how to find the kinematics in data using Unity3D and Excel

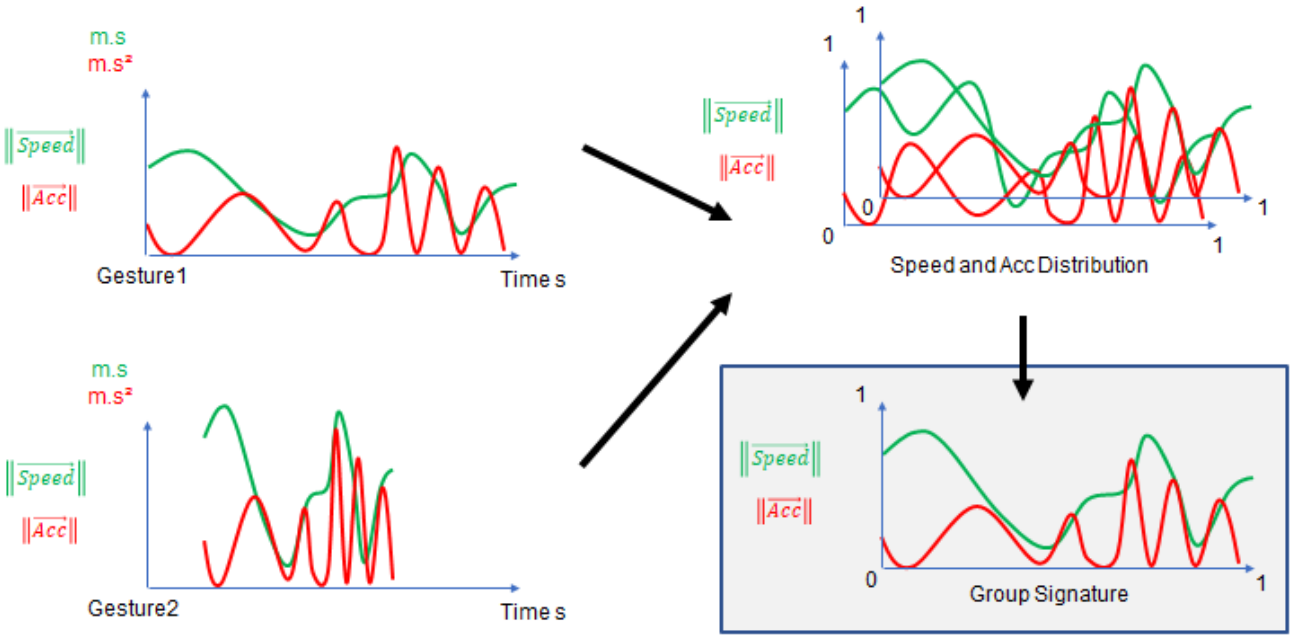


figure: Gesture comparison and development of a group signature

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 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" 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 (figure below). The tool also standardizes and formats the data so that it can be analyzed in a second step with statistical tools.

2.3 Visualizing results and interpretation

Depending on each analysis, the results can be presented in plots with statistical analysis or curves. We also developed a visualization system in Unity3D 3D designed to visualize gesture descriptors such as kinematics in real-time. We detail the different results in the next paragraphs.

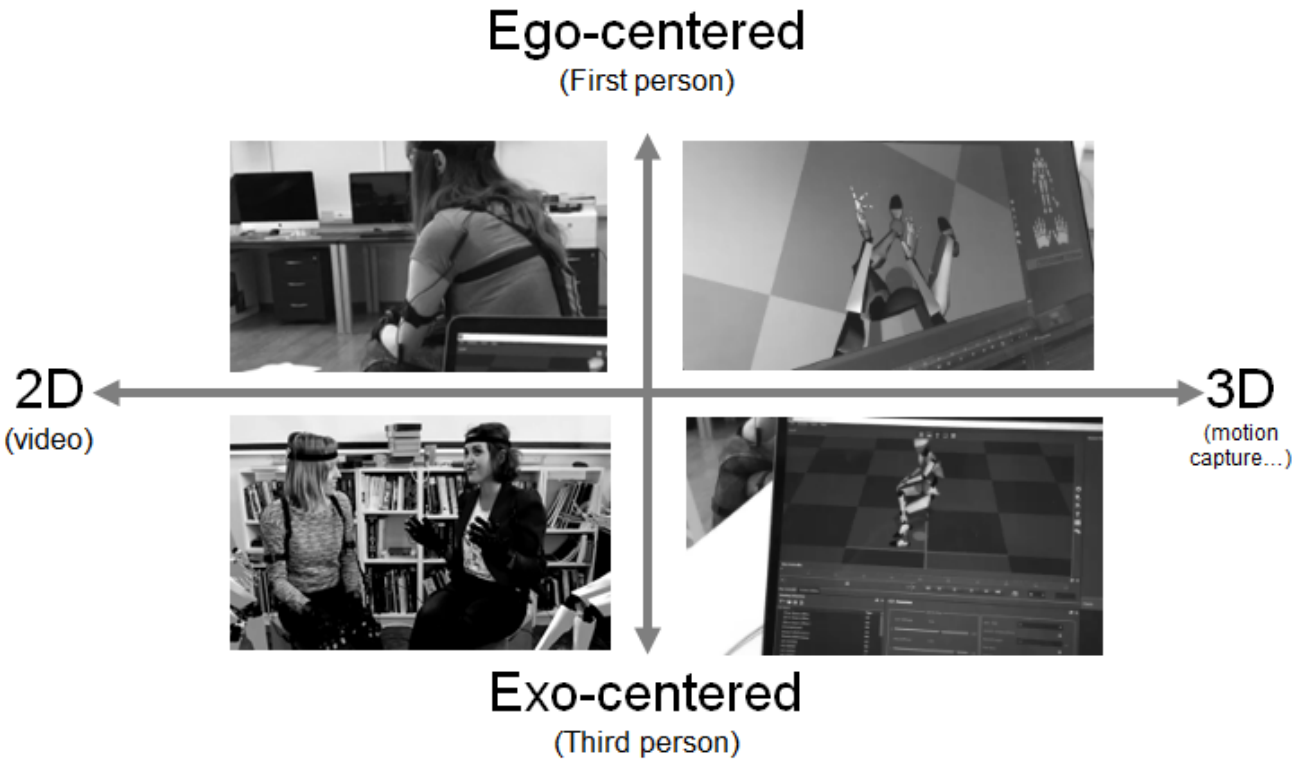


figure: ego or exo-centered points of view (Berthoz, 2000) regarding 2D video or 3D mocap data collections

2.3.A 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.

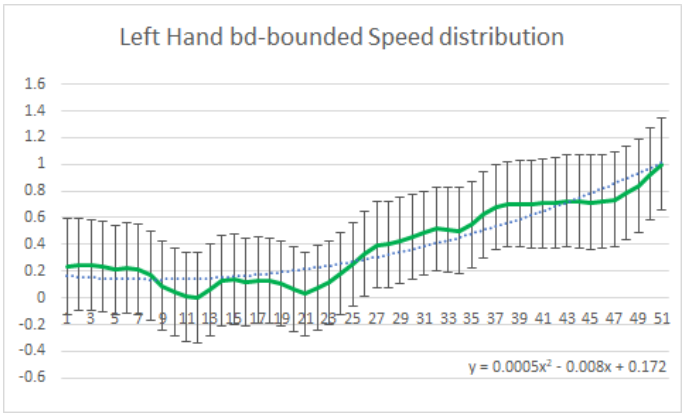
Also, mocap allows to generate multiple points of view. For instance, an exocentered (third-person) point of view close to the one of the video camera. We can also use the advantage of the virtual cameras used in mocap to render for instance a mocap video fith a wider point of view or a close-up on specific body part of the participant. Finally, we can adopt an ego-centered (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.

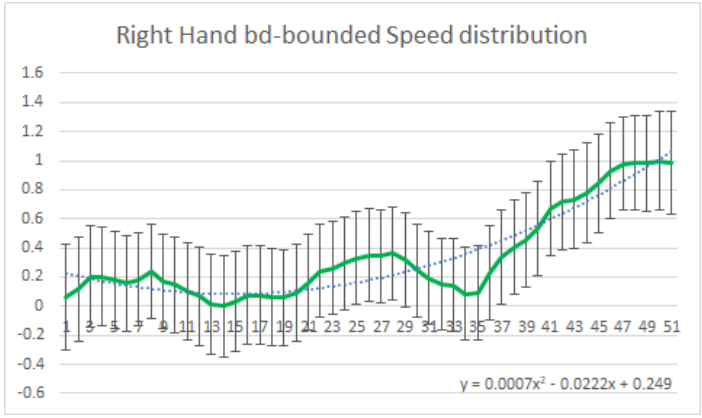
2.3.B Kinesiology & Kinematics: Statistics and Curve Analysis in Excel

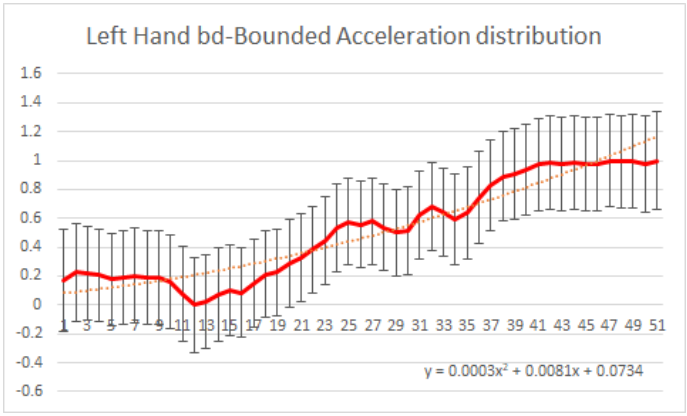
Regarding Kinesiology, we are able to estimate the range or motion of each degree of freedom (dof) for each segment (arm, forearm, hand). On thiinbasis, we extract the biggest amplitude at every frame. The dof thus marked is the one on which the second-largest amplitude determines finally the flow. The 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

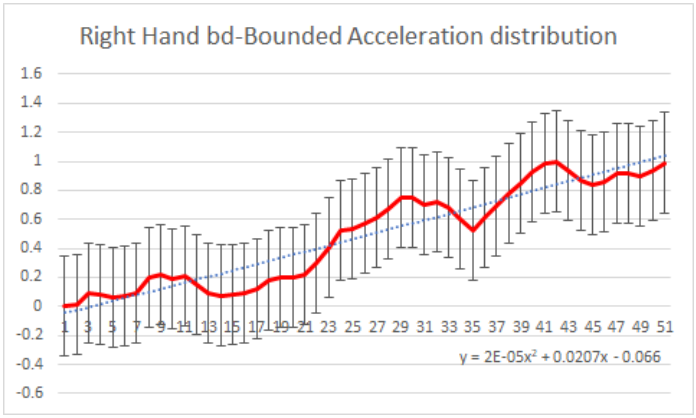
ranges of motion. The mixing ratio overstates the motions for these dof. We have selected this mixing ratio.
A third method to approach the flow introduces the standard variation for each dof. The dof follows a bell curve. We expect that every time a value is outside the standard variation, it is meaningful. 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.

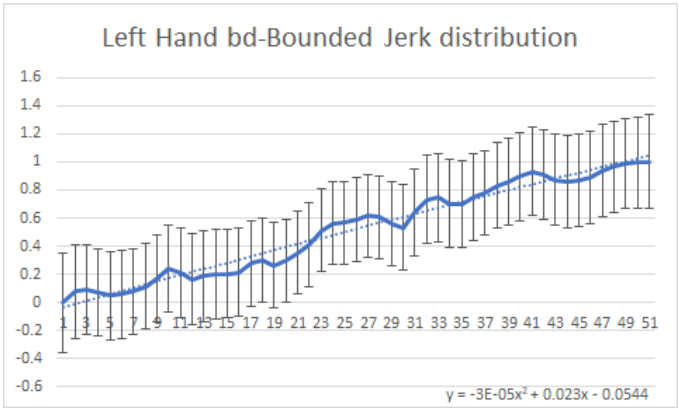
Analyses are conducted on 15 pairs of participants. The analysis of the Motion capture is made on the basis of the video coding data under the Elan software by 3 different coders. We focus on the gesture analysis of gestures with the best triple consensus tagging which represents 348 gestures noted bounded and 148 gestures noted un-bounded for all the user panel.
In the context of bounded gestures coded with Elan, we obtain the following curves for both Hands. Each curve is expressed with the standard deviation on each sample and à polynomial trend curve of degree 2.

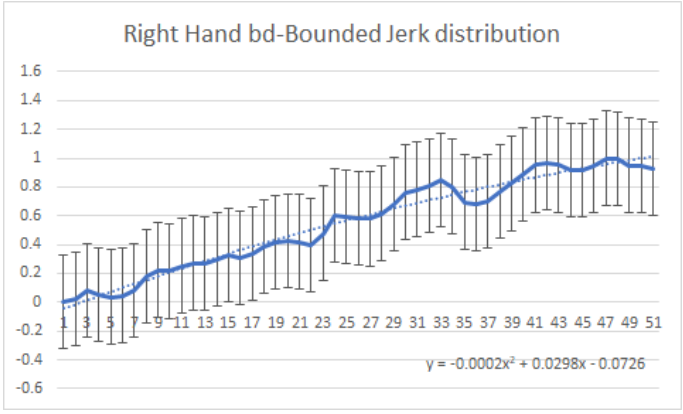






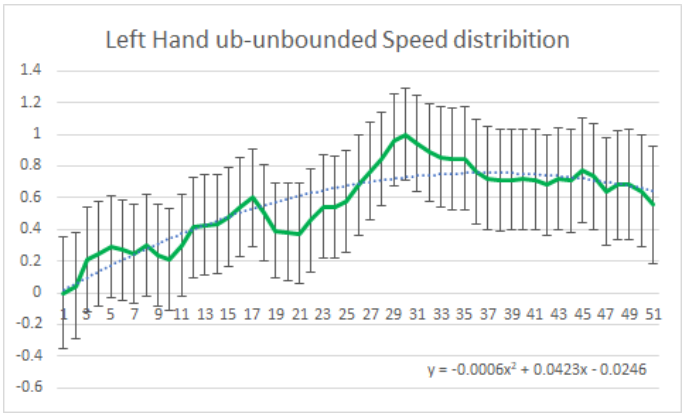


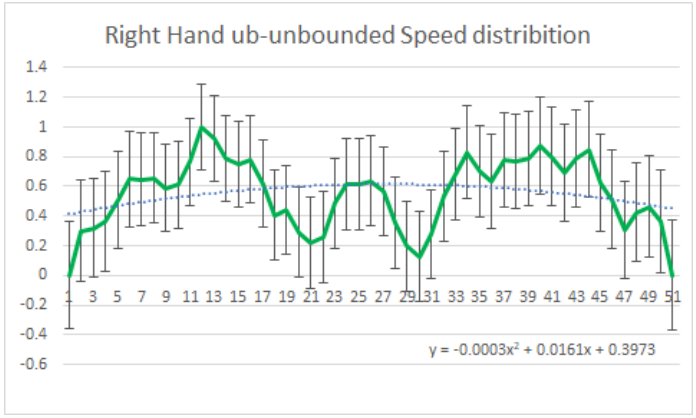


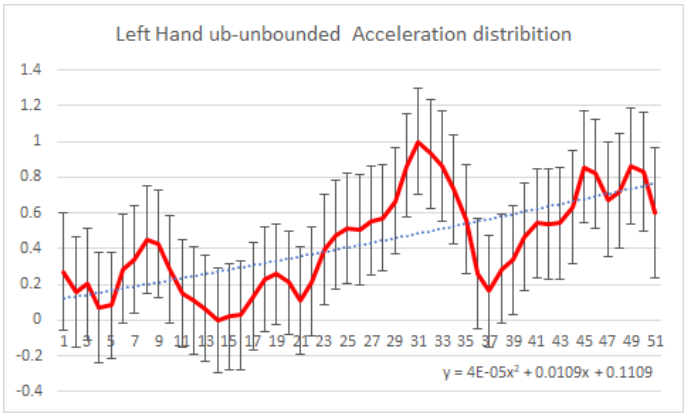


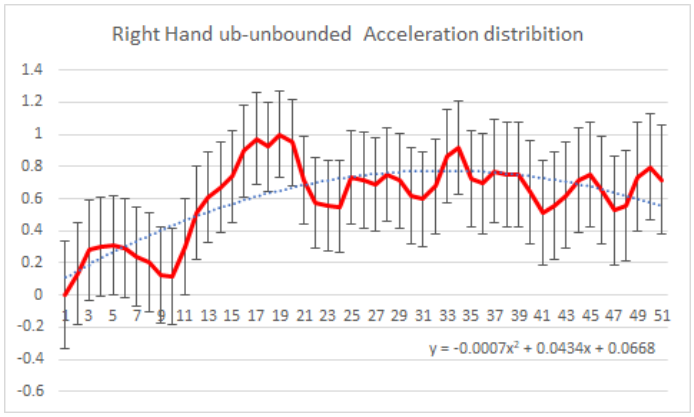
The motion capture analysis shows that both hands have the same behaviors, the bounded gestures have an increasing velocity, acceleration and jerk distribution profiles due to the nature of their shape. Given the number of samples, the standard deviation remains more or less constant on each sample.

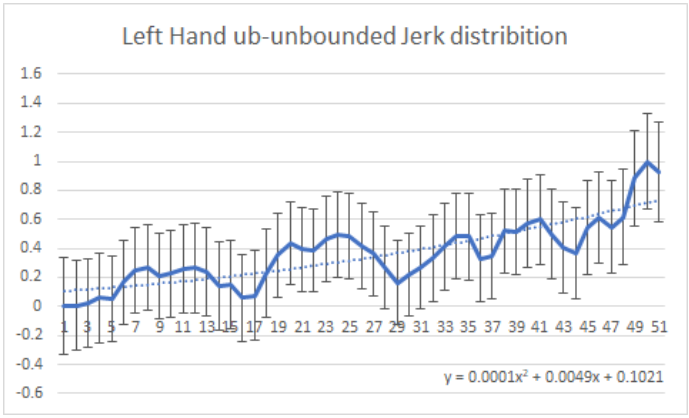
In the context of un-bounded gestures coded with Elan, we obtained the following curves for both hands. Each curve is expressed with the standard deviation on each sample and polynomial trend curve of degree 2.

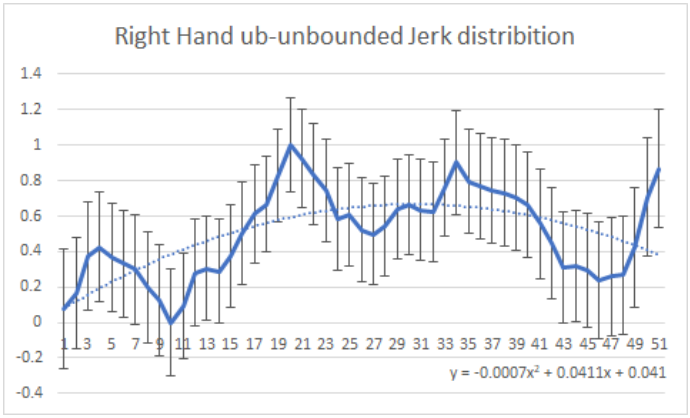












The motion capture analysis shows that the un-bounded gestures have velocity, acceleration and jerk distribution profiles less pronounced than the bounded gesture ones due to less affirmed gesture dynamic. Given the number of samples, the standard deviation remains more or less constant on each sample.

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.

2.3.C 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. 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. 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.

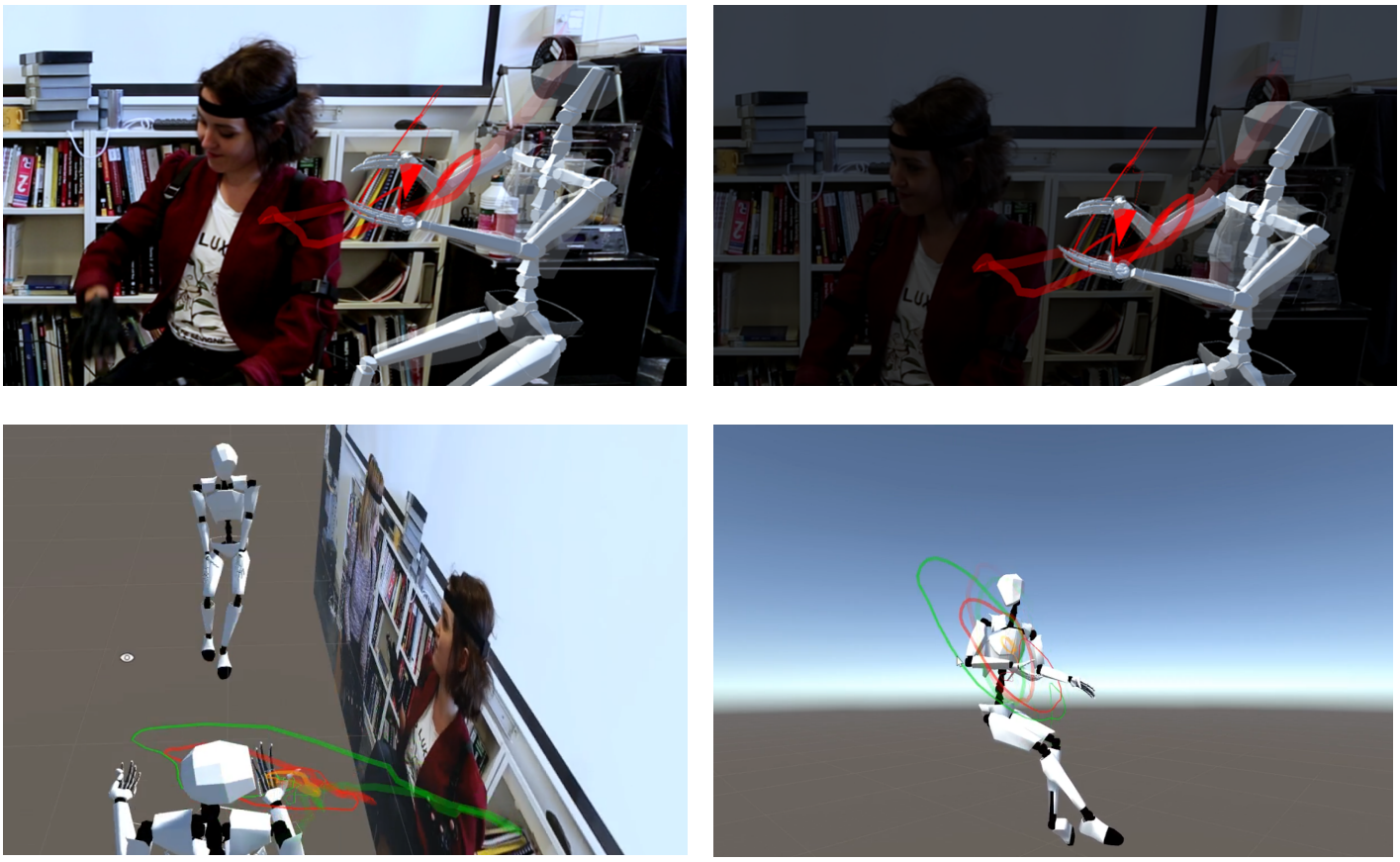


figure: Top: augmented reality visual rendering of gesture kinematics, focus can set on the video or the avatar. Bottom: capture of different points of view in the real-time viewport. Legend of the trails of the right hand of the participant in the right: the position is in orange, the velocity is in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

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.