UNIVERSITÀ DEGLI STUDI DI TRENTO

Computer vision

Gait analysis on OptiTrack

Leonardo Fornalè 224839
Jacopo Donà 229369

A.Y. 2021/2022

1 Introduction

Motion capture is a tool widely used nowadays for the development of virtual characters in movies, video games and in the animation industry. Actors perform stunts and takes in the capture volume area, the skeleton is extracted and finally the animators merge it to a chosen virtual model. A very interesting sub-branch that has been on the rising is the use of motion capture technologies to help doctors and therapists diagnose and treat their patients.
The aim of this project is to acquire the gait for the walk and run of an actor in the capturing volume and so analyze the movement of various body joints through time.

2 OptiTrack

The first phase of the project consists in the takes acquisition. To do so, we used the Creative, Intelligent & Multisensory Interaction Laboratory of DISI. It contains a capturing volume composed by 8 OptiTrack cameras and an additional audio system that for the purpose of this project has not been used.

In order to be detected by the camera, actors must wear a suit composed by a set of markers made of a reflective material. The cameras used by the system emit a infrared light that is being reflected by the markers, this allows the tracking through time.

Since we were only interested in the analysis of the lower body, we used the OptiTrack marker layout *Lower Body* that uses just 20 markers, placed around the hips, thighs, shins and feet of the actor.

Each take starts from an angle of the volume and consists of either of a walk or a run by the actor. In order to analyze and characterize some posture and body movement we acquired takes from 5 different actors. During the takes we noted that one of the actors had a slight anomaly during the walk, in the sense that the actor kept the right foot slightly pointed towards the inside, we will elaborate more on this information in chapter 3 while discussing about results.

Once the take is registered, OptiTrack converts the markers position into body joint position. The final data can be exported in various different formats, each for their own use:

- `.fbx` and `.bvh`: used for animating skeletons with a graphical model.
- `.c3d` and `.trc`: contain only markers information, no body joints.
- `.csv`: this is our export choice and it will be discuss deeper in chapter 3.

3 GUI implementation and results

Once the take is register, OptiTrack analyzes it and return a *.csv* table file. The position and orientation for each frame and joint is described by the column of the file as described in Tab. 1.

		Hip	Hip	Hip	Hip	Hip	Hip	Hip
		Rotation	Rotation	Rotation	Rotation	Position	Position	Position
Frame	Time	X	Y	Z	W	X	Y	Z
0	0.000000	-0.002749	0.298915	-0.000041	-0.954276	2.299.826	0.927871	-2.281.647
1	0.002778	-0.002749	0.298915	-0.000041	-0.954276	2.299.826	0.927871	-2.281.647
2	0.005556	-0.002746	0.298914	-0.000008	-0.954276	2.299.817	0.927878	-2.281.627

Figure 1: Csv format table for the Hip.

The file describes the evolution of the following joints

$$\begin{bmatrix} Hip & LeftArm & LeftHand & RightShoulder & RUArm \\ RfArm & RightHand & LeftThigh & LeftShin & LeftFoot \\ RightThigh & RightShin & RightFoot & LeftToe & RightToe \end{bmatrix} \tag{1}$$

Even though we had markers only from the hip below, OptiTrack still emulates the joints of the upper body joints. These ones (*torso, shoulders, arms, etc...*) maintain a T-pose throughout the take. Since this information is simulated and does not provides any major information for this task, we chose to simply ignore these positions.

Each joint is described by 7 values: $\begin{bmatrix} X & Y & Z \end{bmatrix}$ for the space position and $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$ for the rotation described by quaternions which can be changed by means of a transformation matrix in Euler angle. The GUI implementation analyze only the position of the point therefore the rotation information is not used.

The GUI is split in three windows:

1. Load data: The GUI provide the possibility to load up to 9 *.csv* files. In our directory, in order to provide a fast detection they must be placed inside the `data` folder.

2. 2D plot: In this window there are two modes which are interchangeable by the `Enable multiple file joint` button:

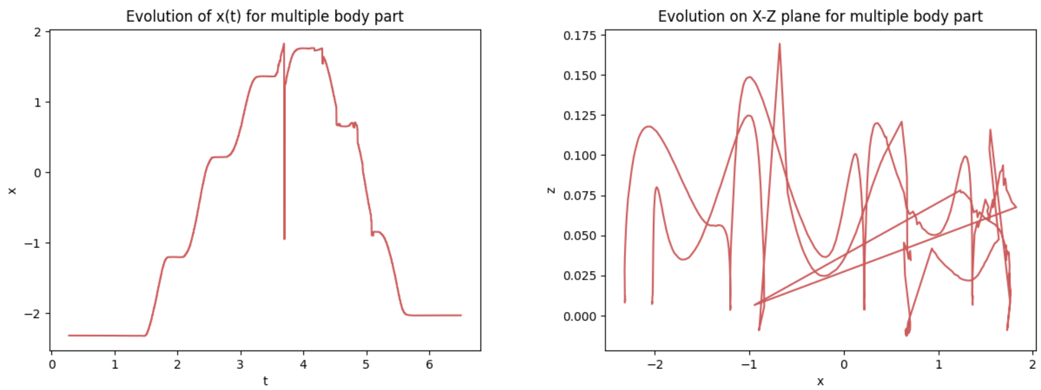
   (a) Single file: In this mode it is possible to select the body parts. By default the middle plot describes the evolution in plane X-Y, this can be changed by using the spinner on the left. The evolution in time of the three coordinates are described by the surrounding plots.

(b) Multiple files: Use the `enable multiple joint` button to enable this mode. From the left panel it is possible to choose multiple files. By default the chosen joint is the *Hip*, it can be changed using the spinner near the `enable multiple joint` button.

3. 3D plot: This window provide a 3D representation of each joint for a single take.

Each of these windows has a `save plot` button, the *.png* plots are saved in the same folder of the python main script. The maximum frame can be selected by means of a slider, using the `auto update` button the graphs can be updated in real time, even if this is a heavy situation for the GUI and it is better to change the slider without the `auto update` and then plot again.

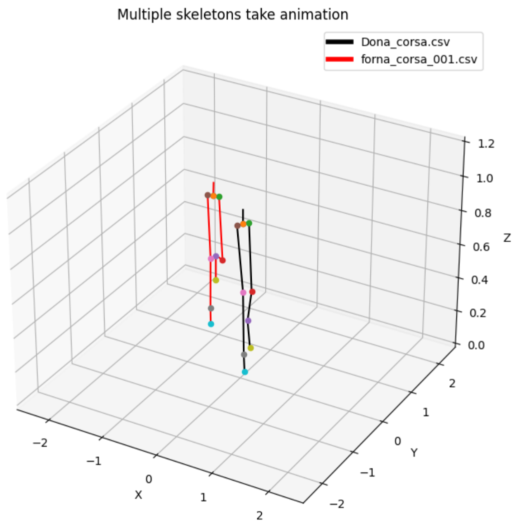
One of the main problem we faced during our take analysis is that in some of the joints vector multiple frames are missing. This issue could be caused by occlusions or the miss position of the actor near the border of the system capturing volume. In order to address this issue without losing the whole take information we opted to truncate the vector, with the assumption that during the display of the skeleton only few frames may be missing. This assumption is mostly true, but in some instances it can leads to a partial skeleton as represented in Fig. 2. This problem is mainly present during the change in direction of the body so at the border of the capturing volume as pictured in 3 (*b*).



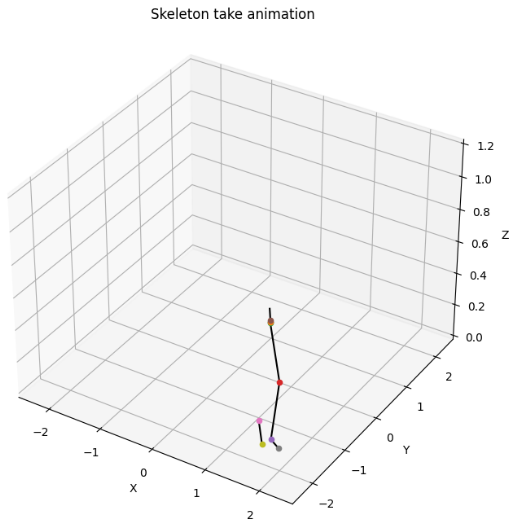
(a) (b)

Figure 2: Evolution in time and in space of a truncated run-take.

Since the library used for the GUI does not allows the refreshing of graph plots, a second python script named `skeleton_matplotlib` has been created. This file can be run by command line, and can choose up to two files for the animation. Joints are displayed by a point in space and nearby joints are connected using lines, thus creating a very simple skeleton model. A design choice made in this program was to refresh the skeleton position with a 10 frame step. This decision was made in order to make the animation move faster, since plot refresh take some time and the takes are taken with a really high frame rate (144 fps). By reducing this value the animation will have smoother movements but consequently move slower. In Fig. 3 we reported a screenshot during the animation with 2 actors, while in the right figure the problem of truncation is highlighted by the incomplete joints of the skeleton. As said before, this situation is mostly limited to some frames and at the borders of the volume. For most of the take the skeleton joints are fully present.



(a)

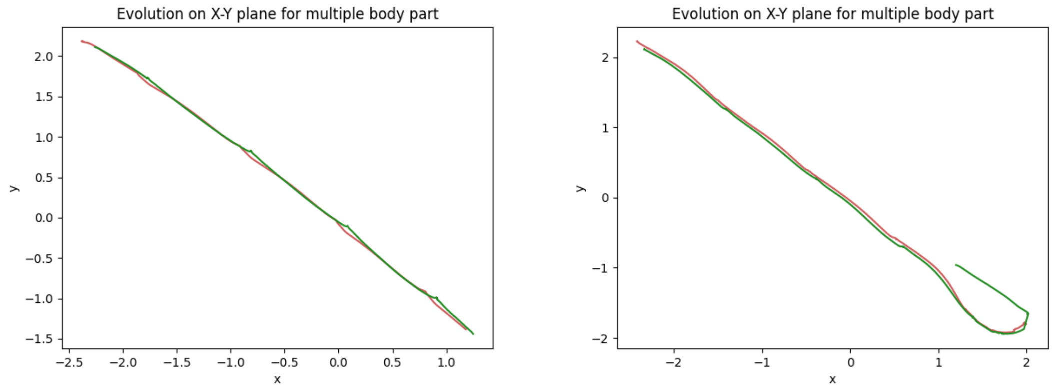


(b)

Figure 3: (a) Animation of two files. (b) Problem of truncation in the position vector.

As earlier mentioned, during the lab capturing session we noticed that an actor while walking pointed his toe towards the inside. This "anomalous" behaviour can also be seen thanks to the plots in Fig. 4. It is possible to look at the comparison between `right toe` and `right foot` of two different actor takes. It is noticeable that in the left figure the tiptoe points inwards, meanwhile in the right graph the toe is pointed towards the outside (as it should behave).

This confrontation can be done with all the other actors so we believe that our graphs, having enough domain expertise, may help in detect anomalies in the gait.



(a) (b)

Figure 4: (a) Toe inward. (b) Toe outward.

4 Conclusion

Overall, we are pleased with the results we were able to achieve and, this being our first experience with motion capture, we really enjoyed capturing takes in the lab and analyzing data through software.

A potential improvement to our work could be the integration of a graphical model of a human skeleton to actually see the movement of the anatomy of each actor. However, with the way our takes are treated, we fear we may encounter some potential obstacles.

As illustrated, missing joints information can cause the loss of registered data. In this project, these situations does not affects too much the analysis, especially during the animation of the skeleton. However, if we were to blend a graphical model onto these takes, we would have certain frames where the model behaviour is unspecified, potentially generating some artifacts in the model appearance.

With the expertise we now have, maybe it would be better to capture movements that are more confined to the center of the capturing volume, or using some external products such as tapis-roulant, in order to do extensive walking/running takes without moving to the borders of the volume. Another solution could be to expand the volume of the capturing system, but this would mean using more cameras for tracking the markers, leading to an increase to the overall system cost. Finally, an interesting future development could be to acquire more takes of walks and then feed the joint's data to a Machine Learning model, in order to develop an anomaly detection algorithm for the posture during the walk of a patient.

5 Code

The source code can be found at the following `github repository`.

We suggest reading the `README.md` for guidance on how to run the software.