CSCI520 Assignment 2 Report: Motion Capture Interpolation

Leonardo Zhu

Introduction

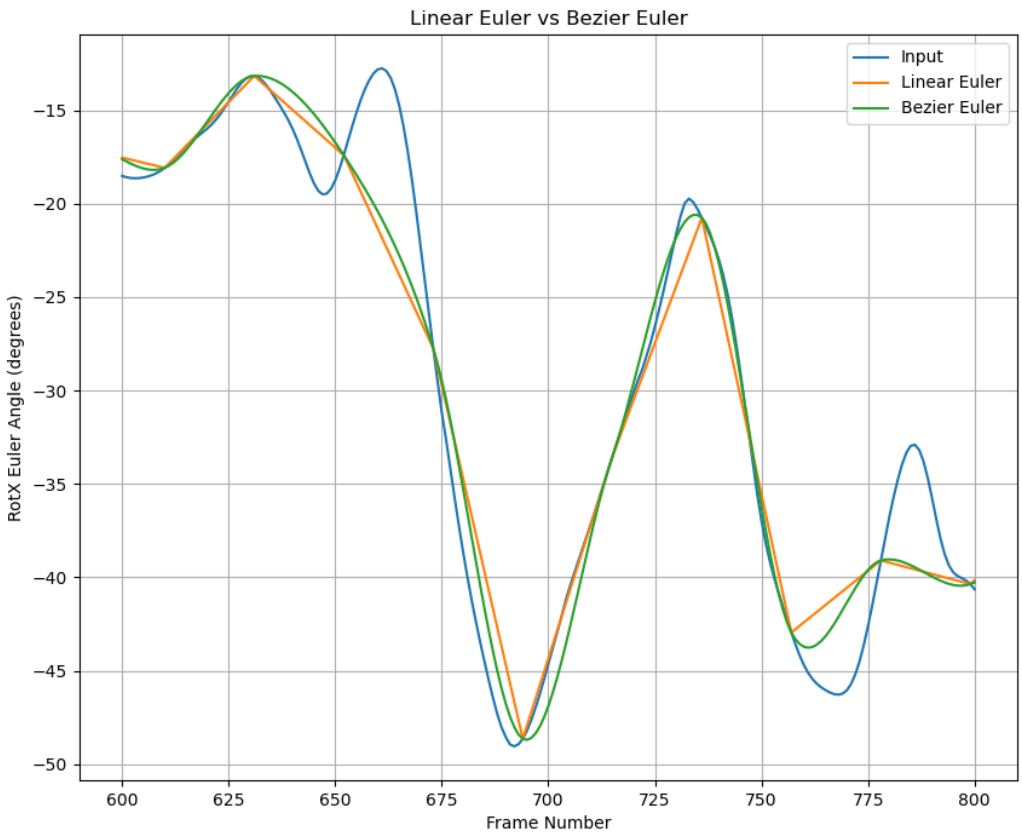
In this assignment, I implement Bezier interpolation for Euler angles, as well as SLERP and Bezier SLERP interpolations for quaternions to interpolate human motion data obtained from an optical mocap system. The human model (skeleton, saved in .asf files) is structured hierarchically, with a root node and several child nodes representing different joints. The recorded motions are stored in .amc files, where joint rotations are encoded using Euler angles. To create keyframes for interpolation, N consecutive frames are removed from the original motion, with N being a specified input parameter. The program generates a motion sequence of the same length as the original, based on the value of N and the chosen interpolation method.
The source code along with all the AMC and ASF files used in this project are located at the following path:
`‘unzipped_folder’/mocapPlayer-starter/`

Locations of the two executable files (`interpolate.exe` and `mocapPlayer.exe`):
`‘unzipped_folder’/IDE-starter/VS2017/Debug/`

Graphs

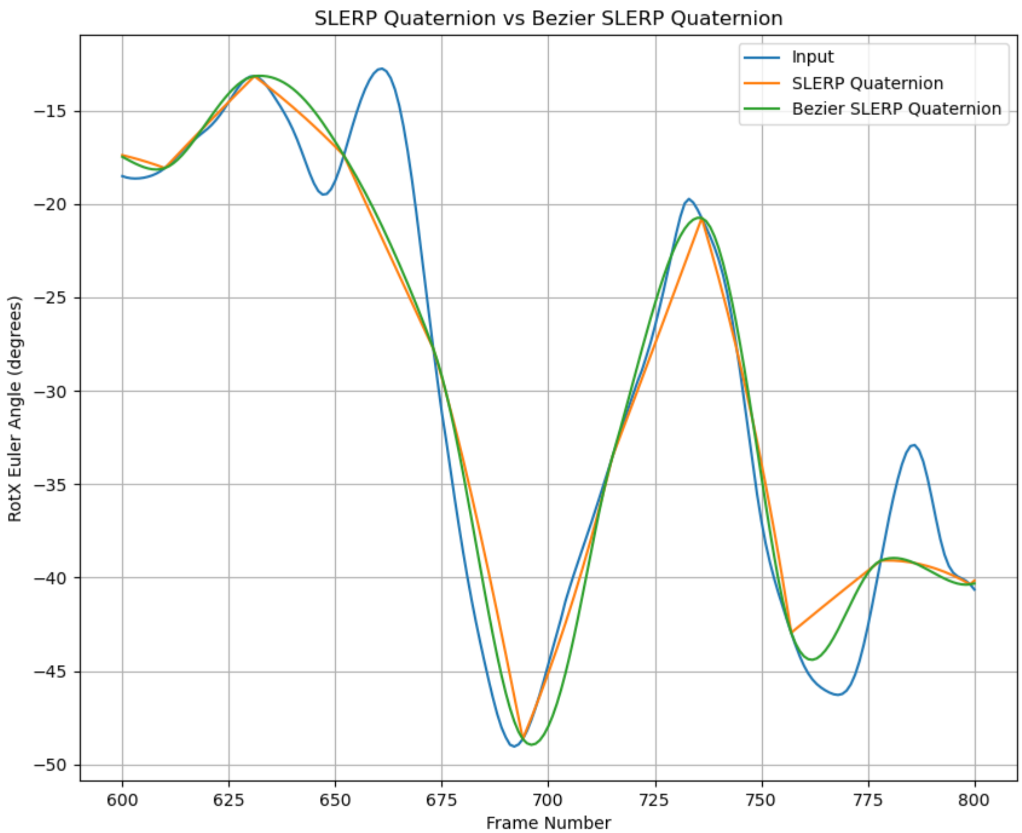
Path to the four graphs, the corresponding .amc files, and the code to plot the graphs:
`‘unzipped_folder’/graphs/`

Graphs 1 and 2 below are plotted for lfemur joint, rotation around X axis, frames 600-800, with N=20, for 131_04-dance.amc input file.



Graph 1: Comparison between linear Euler and Bezier Euler interpolation (and input).

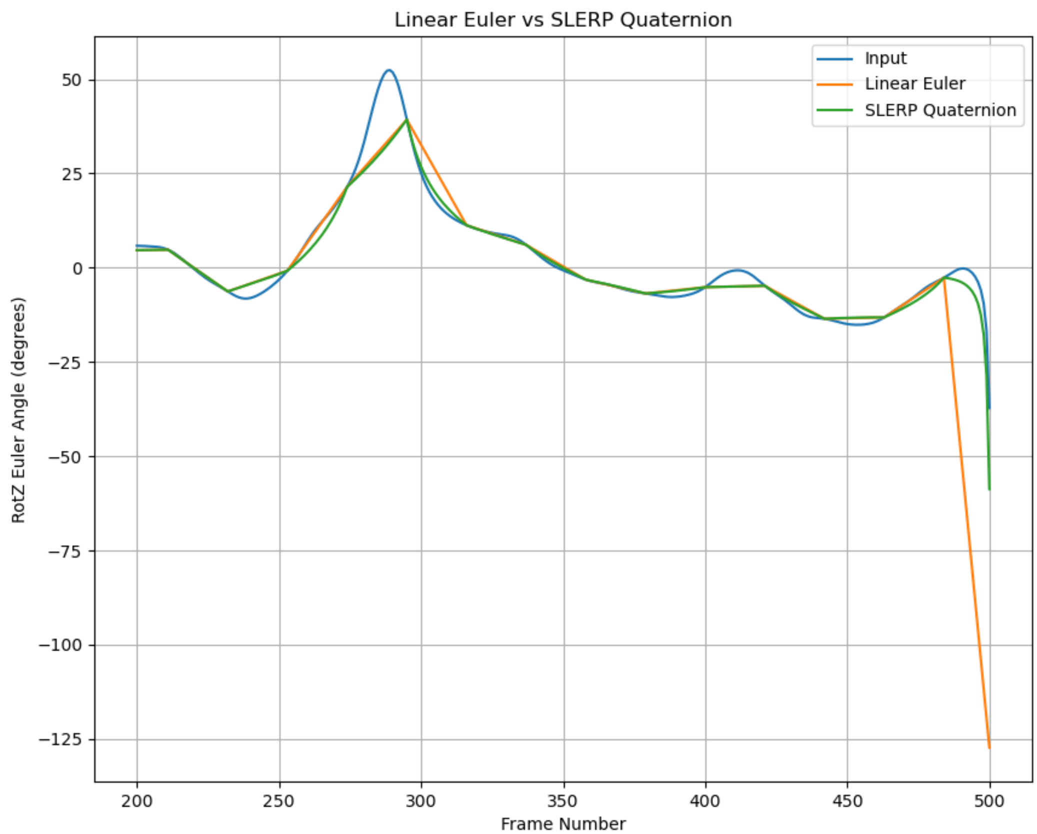
Graph 1 Observation: As expected, the curve of Bezier Euler interpolation is smoother than the curve of the linear Euler method around keyframes. However, when compared to the curve of the input motion, both interpolation techniques lose many details due to the limitation on interpolation methods. For example, the two turning points around Frame 650 in the input motion are missing in both interpolated curves because those turning points in Euler angles are included in the dropped frames instead of the keyframes. In other words, Bezier Euler seems to perform slightly better than linear Euler but cannot claim a complete victory.



Graph 2: Comparison between SLERP quaternion and Bezier SLERP quaternion interpolation (and input).

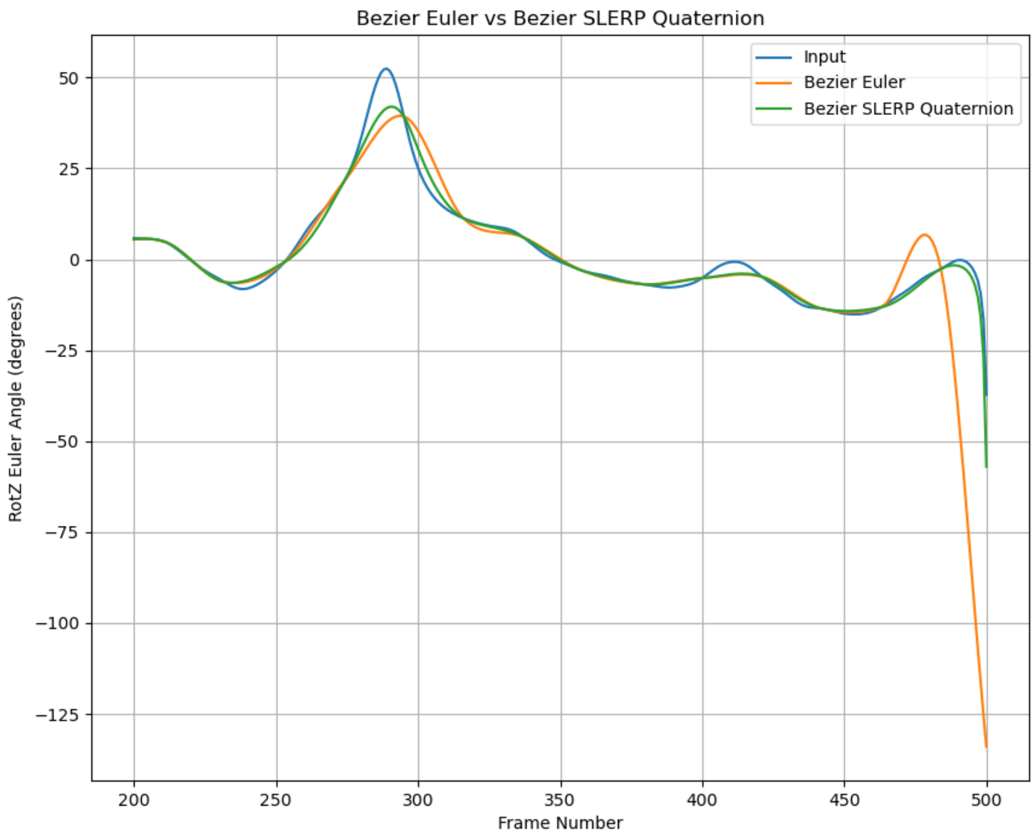
Graph 2 Observation: As expected, the curve of Bezier SLERP quaternion interpolation is smoother than the curve of SLERP quaternion interpolation around keyframes. However, similar to the Euler methods, when compared to the curve of the input motion, both quaternion interpolation techniques lose many details in this case. The overall performance of Bezier SLERP quaternion is slightly better than that of SLERP quaternion. **The difference between Graph 1 and Graph 2 is basically negligible, which means that Bézier interpolation performs better overall than linear methods, but only to a limited extent.**

Graphs 3 and 4 below are plotted for root joint, rotation around Z axis, frames 200-500, with N=20, for 131_04-dance.amc input file.



Graph 3: Comparison between linear Euler and SLERP quaternion (and input).

Graph 3 Observation: As both linear Euler and SLERP quaternion are linear interpolation methods, their curves overlap with each other in most parts. However, the curve of SLERP quaternion is smoother than the curve of linear Euler between Frames 250 and 320, and after Frame 460, which means SLERP quaternion interpolation performs slightly better than linear Euler interpolation.



Graph 4: Comparison between Bezier Euler and Bezier SLERP quaternion (and input).

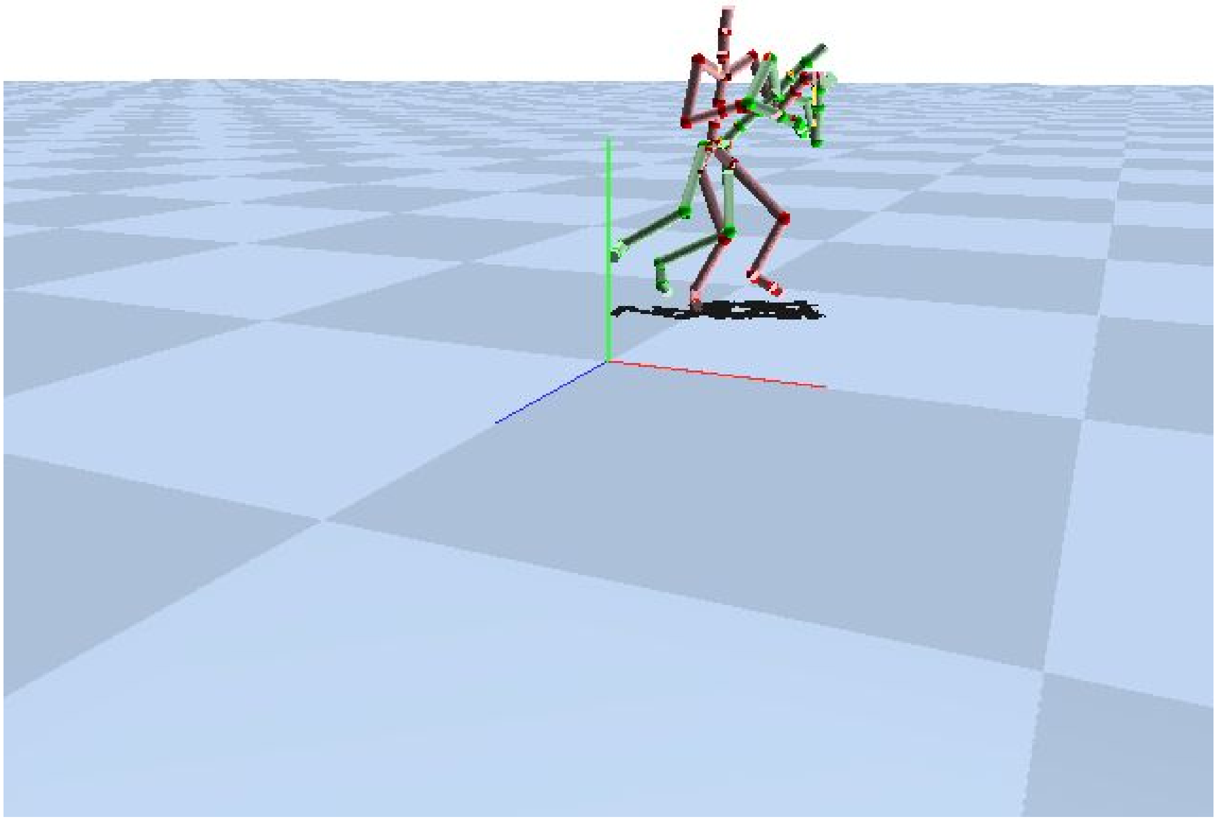
Graph 4 Observation: The curves of Bezier Euler and Bezier SLERP quaternion interpolation are both smooth around keyframes, as expected for Bezier splines. However, Bezier SLERP quaternion interpolation produces motions that most closely resemble the input motion, particularly between Frames 460 and 500. Thus, by comparing Graph 3 and Graph 4, we can conclude that **quaternion interpolation performs better overall than Euler angle interpolation, with Bezier SLERP quaternion interpolation performing the best.**

Videos

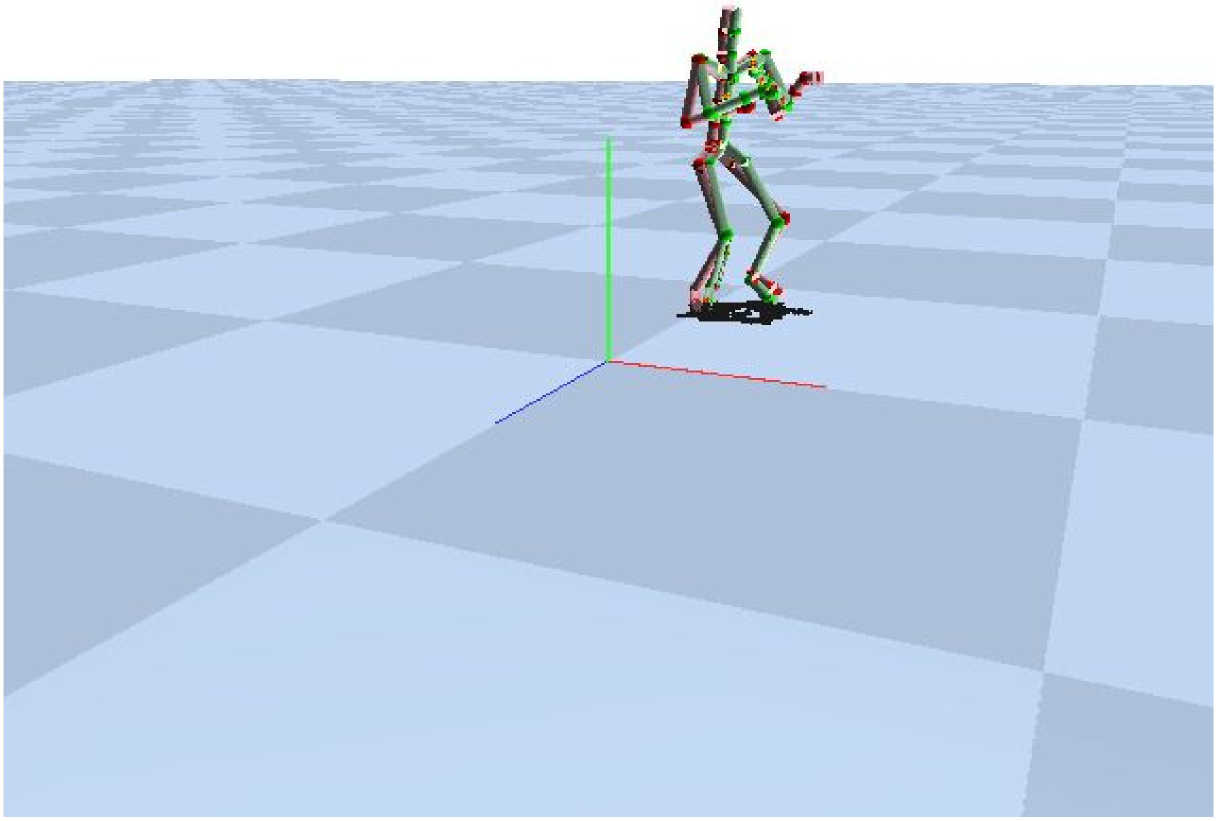
Path to the three videos: `'unzipped_folder'/videos/`

These three videos show comparisons between the input motion and the interpolated results using different interpolation techniques for 135_06-martialArts.amc with N=40, where the input motion is displayed in red, and the interpolated motion is displayed in green.

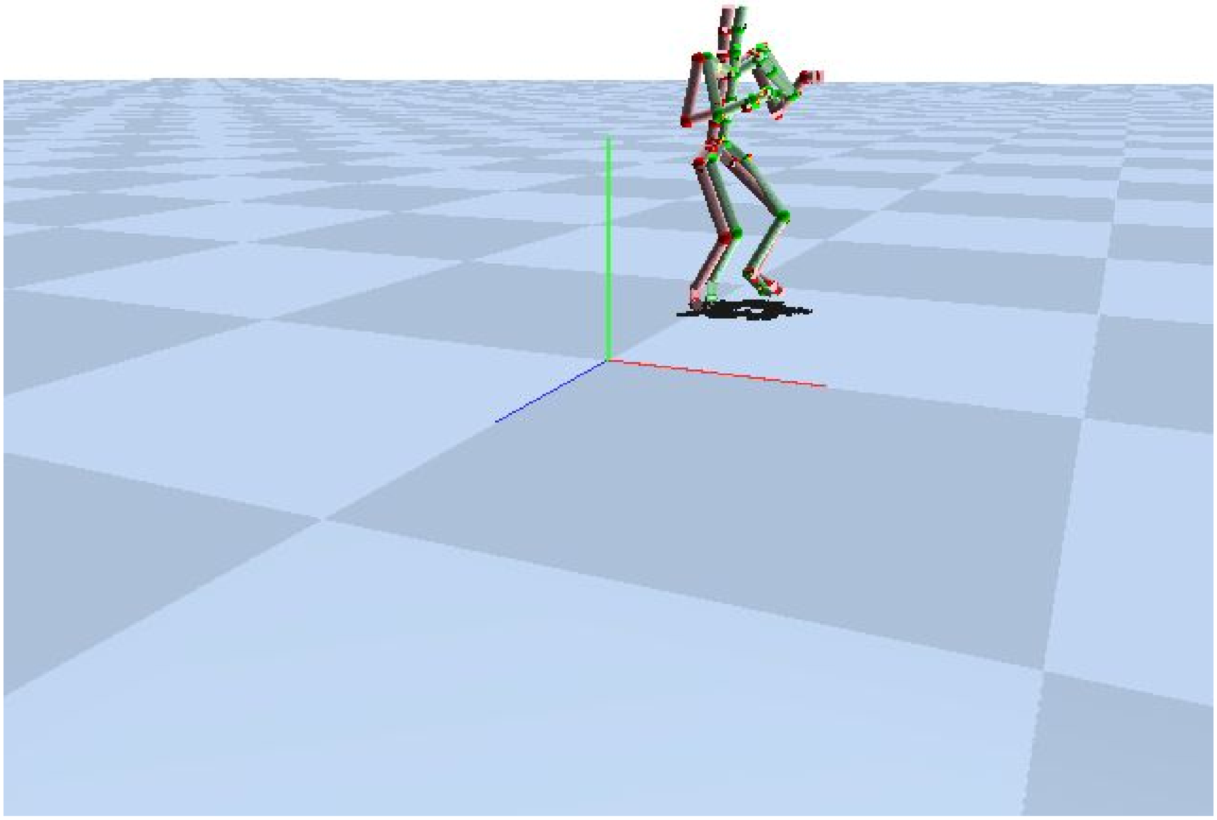
By observing the three comparison videos, we can see that some frames interpolated by the Bezier Euler method exhibit unrealistic or exaggerated motion, while the motion sequences generated by both SLERP quaternion and Bezier SLERP quaternion appear more reasonable and more closely resemble the original motion. For instance, in Frame 2392, as shown below, the green skeleton representing the interpolated motion is leaning forward with its feet floating above the ground, appearing to stumble over something and deviate from the input motion's trajectory. In contrast, in spite of small difference in joint positions, the motions interpolated by both SLERP quaternion and Bezier SLERP quaternion closely resemble the original motion. Therefore, we can conclude from the videos that **quaternion interpolation performs significantly better than Euler angle interpolation.**



Frame 2392 interpolated by Bezier Euler.



Frame 2392 interpolated by SLERP quaternion.



Frame 2392 interpolated by Bezier SLERP quaternion.

Extra Credit

- Changed the ground color to fog blue in the mocapPlayer.
- To further distinguish the skeletons when two or more motions are displayed simultaneously, I created a list of bone colors in `displaySkeleton.cpp` to ensure consistency with the joint colors. Specifically, the bone colors cycle through light green, light red, and light blue.
- With the utility class `PerformanceCounter` defined in `performanceCounter.h`, I modified the output of the interpolator to print the computation time of different interpolation techniques. For instance, I applied the four interpolation approaches to interpolate frames for "131_04-dance.amc" input file with N = 20 to get their corresponding interpolation times, as shown in the screenshots below.
  Computation time comparison: Linear Euler (0.0584096 sec) < Bezier Euler (0.609423 sec) < SLERP quaternion (0.747242 sec) < Bezier SLERP quaternion (1.93384 sec). The difference in computation time between the least time-consuming Linear Euler

interpolation and the most time-consuming Bezier quaternion interpolation is almost two orders of magnitude. Given that Bezier Euler is even slightly faster than SLERP quaternion, when compared to the tiny time spent in Linear Euler, it is reasonable to say that SLERPing quaternions costs more time than calculating Bezier spline control points and implementing the De Casteljau algorithm for interpolation of root positions and bones' rotations. The reason is probably that converting the rotations of bones between Euler angles and quaternions back and forth requires a lot of computing power, as this kind of conversion involves matrix manipulation.

```
Loading input motion from ../../../mocapPlayer-starter/131_04-dance.amc...
1086 samples in '../../../mocapPlayer-starter/131_04-dance.amc' are read.
Interpolation type is: LINEAR
Angle representation for interpolation is: EULER
Interpolating...
Computation Time for Linear Euler Interpolation: 0.0584096 sec
Interpolation completed.
Writing output motion capture file to 131-le.amc...
Write 1086 samples to '131-le.amc'
```

Screenshot for computation time of linear Euler interpolation.

```
Loading input motion from ../../../mocapPlayer-starter/131_04-dance.amc...
1086 samples in '../../../mocapPlayer-starter/131_04-dance.amc' are read.
Interpolation type is: BEZIER
Angle representation for interpolation is: EULER
Interpolating...
Computation Time for Bezier Euler Interpolation: 0.609423 sec
Interpolation completed.
Writing output motion capture file to 131-be.amc...
Write 1086 samples to '131-be.amc'
```

Screenshot for computation time of Bezier Euler interpolation.

```
Loading input motion from ../../../mocapPlayer-starter/131_04-dance.amc...
1086 samples in '../../../mocapPlayer-starter/131_04-dance.amc' are read.
Interpolation type is: LINEAR
Angle representation for interpolation is: QUATERNION
Interpolating...
Computation Time for SLERP Quaternion Interpolation: 0.747242 sec
Interpolation completed.
Writing output motion capture file to 131-lq.amc...
Write 1086 samples to '131-lq.amc'
```

Screenshot for computation time of SLERP quaternion interpolation.

```
Loading input motion from ../../../mocapPlayer-starter/131_04-dance.amc..
1086 samples in '../../../mocapPlayer-starter/131_04-dance.amc' are read.
Interpolation type is: BEZIER
Angle representation for interpolation is: QUATERNION
Interpolating...
Computation Time for Bezier SLERP Quaternion Interpolation: 1.93384 sec
Interpolation completed.
Writing output motion capture file to 131-bq.amc...
Write 1086 samples to '131-bq.amc'
```

Screenshot for computation time of Bezier SLERP quaternion interpolation.

Conclusion

Euler Angle Interpolation:

Advantages:

- Easy to implement
- Extremely rapid interpolation

Disadvantages:

- Gimbal lock problem
- Produces unrealistic motions

Quaternion Interpolation:

Advantages:

- Smooth rotations
- No singularities

Disadvantages:

- Computationally expensive
- Acceleration continuity is not preserved to produce motion at unnatural speed

Linear Interpolation (including SLERP):

Advantages:

- Easy to implement
- Computation time is least

Disadvantages:

- Abrupt change in motion around keyframes
- Cause artifacts frequently, especially for curved trajectories

Bezier Interpolation (including SLERP):
Advantages:

- Smooth and natural motion around keyframes

Disadvantages:

- Complex to implement
- Computationally expensive

In summary, according to the observation above, Bezier SLERP quaternion interpolation performs the best and most closely resemble the original motion among the four interpolation approaches but requires the most amount of computation time. Linear Euler interpolation performs worst and tends to generate unrealistic motions but features the least computation time. Based on the analysis of computation time, SLERP quaternion interpolation is the best choice when balancing computation time and interpolation accuracy.