CSCI 520 Computer Animation

Assignment #2

Wanyu Zhang
6773445719

2023-03-10

University of Southern California

Section 1 Interpolation Results and Findings

1.1 Interpolation Results

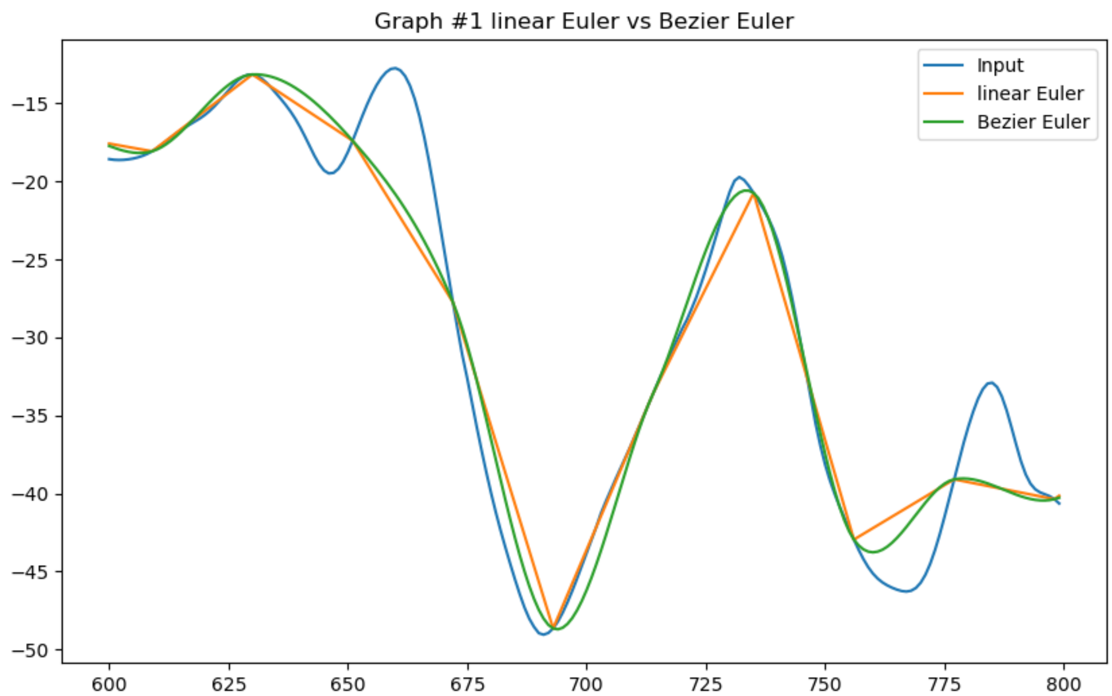

Figure 1: Linear Euler Interpolation versus Bezier Euler Interpolation

Figure 1 compares linear Euler to Bezier Euler interpolation, where we can directly see the line of linear Euler is more straight than Bezier Euler because without control points, linear Euler exactly interpolates point along the segment. For Bezier Euler, it looks more similar than linear Euler, however, at some peak points (a short sequence of points which has big variation), for instance 660 - 670, the Bezier Euler doesn't perform well because of the sample rate. We sample on a basis of every 20 samples, which means if we cannot sample at a higher rate such as twice as input signal, we will not be able to recover it.

Problem 1 continued on next page. . .

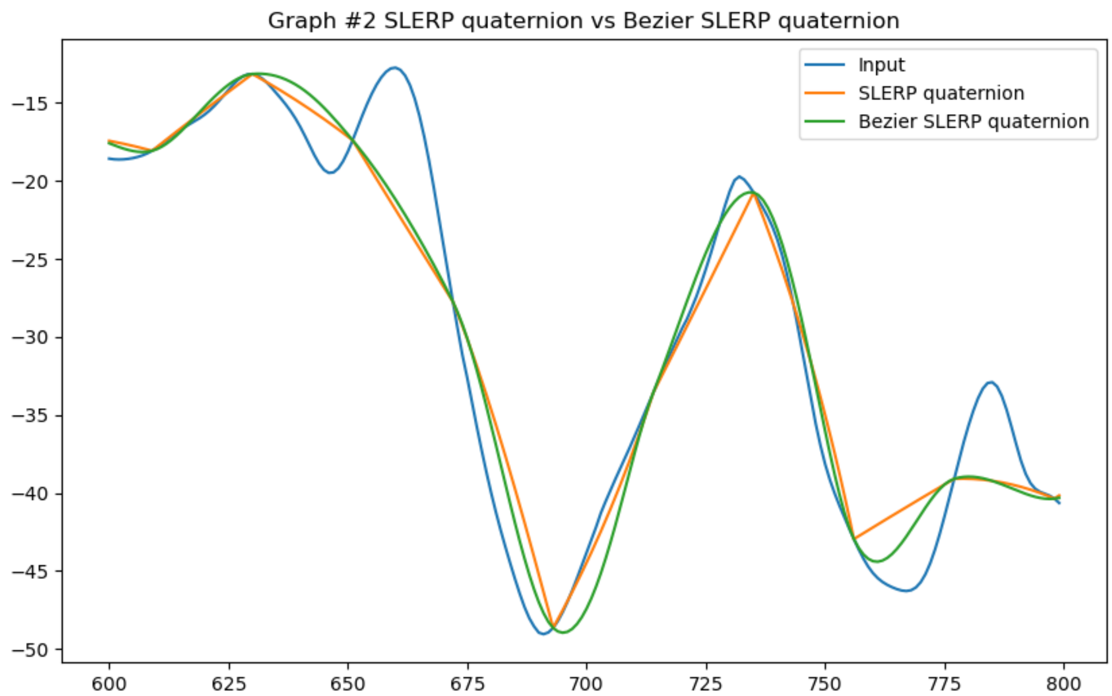

Figure 2: SLERP quaternion Interpolation versus Bezier SLERP quaternion Interpolation

Figure 2 compares SLERP quaternion to Bezier SLERP quaternion interpolation, and the figure looks alike figure 1, it is because we simply use another pose measurement called quaternion instead of euler degrees. Here we can see SLERP is like a ”linear” method on sphere and it counts for why the orange line is more straight. Still, Bezier interpolation is more similar but still cannot recover perfectly at some peak points.

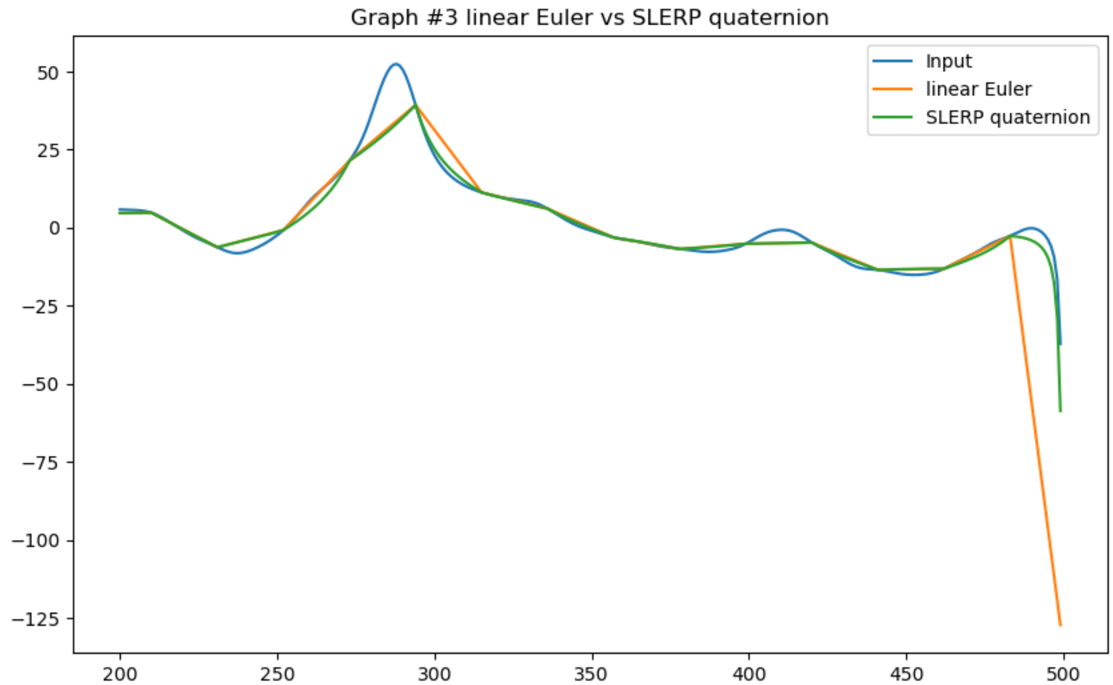

Figure 3: linear Euler Interpolation versus SLERP quaternion Interpolation

Figure 3 compares linear Euler to SLERP quaternion, and they both use ”linear” interpolation methods under euler degrees or quaternion. Both methods present a straight-alike line but quaternion performs more smoothly. At the meantime, the character is rotating in 3D scene, therefore, quaternion interpolation is a better method when angles change quickly. We can clearly see at the last 100 frames there are a lot of aliasing issues of Euler degrees because of the degrees variation.

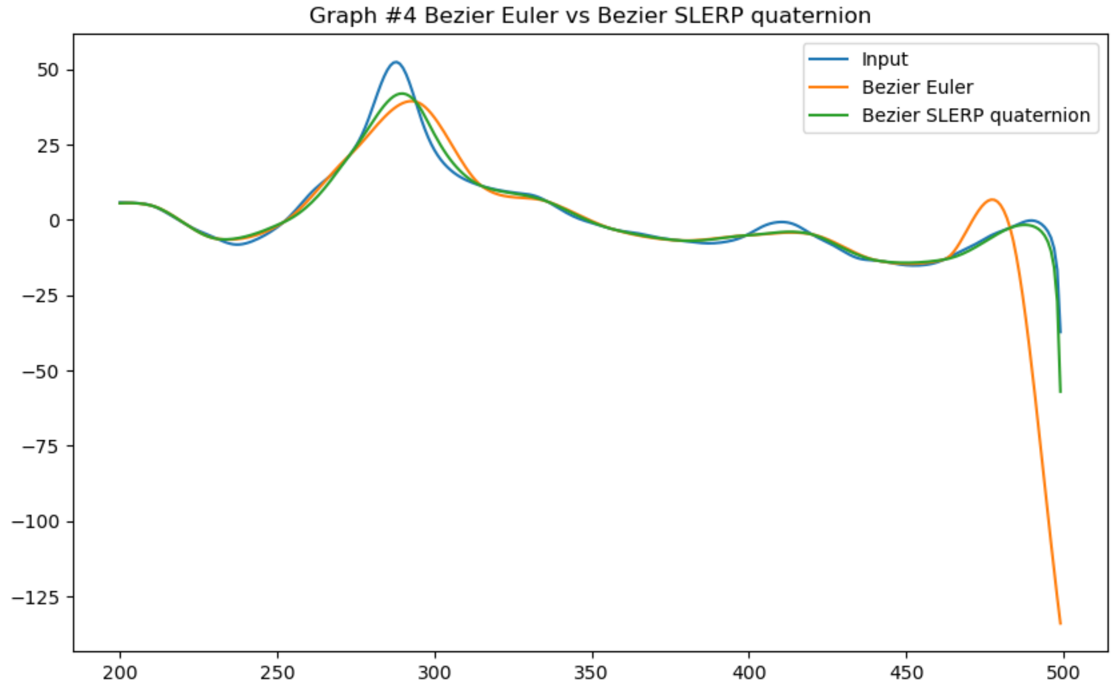

Figure 4: Bezier Euler versus Bezier SLERP quaternion Interpolation

Figure 4 compares Bezier Euler to Bezier SLERP quaternion and both lines are more smooth and quaternion is closer to original input. We can still observe that Euler still doesn't perform well in the last 100 frames, and it again proves that quaternion interpolation is a stable method when angles change quickly.

Here is the conclusion of four different combinations:

	Linear/SLERP	Bezier
Euler	Very basic, only perform well at linear segement	smooth but unstable
Quaternion	stable but not smooth enough	stable and smooth

Obviously, Bezier Quaternion performs best.

Section 2 Computation Time Analysis and Improvement

2.1 Computation Time Analysis

Result shows below(Under Debug mode):

Method	dance.amc	martialArts.amc
Linear Euler	64ms	198ms
Bezier Euler	292ms	882ms
Linear Quaternion	234ms	706ms
Bezier Quaternion	491ms	1455ms

We can read from the table that Bezier Quaternion takes most time because the method requires routine from euler angles to Quaternion and Quaternion to euler angles, moreover, when we use SLERP method to interpolate, we do a lot of float operations. Also, we need to form two Bezier control points and it takes extra time. Therefore, although Bezier Quaternion performs best it is also very time-consuming.

2.2 Improvement

When we form Bezier control points, we can notice that for frame n, we need to know a_n and b_{n+1} which means we have already computed the control points for a future frame $n + 1$.
Here I come up with an idea that we can try swap buffer (like OpenGL), we have to compute b_{n+1} control points in advance, then for every frame $n(n > 1)$, we compute a_{n+1}, b_{n+1} control points only, and at the end of each iteration, we swap a_n and b_n with a_{n+1} and b_{n+1}. Therefore, for frame $n + 1$, we only need to compute a_{n+2}, b_{n+2}.

Method	dance.amc	martialArts.amc
Bezier Quaternion	491ms	1455ms
Bezier Quaternion without Swapping Buffer	529ms	1508ms
Improvements	38ms	53ms

When interpolation with more joints, it will be more efficient.

Section 3 Extra Credits

3.1 OpenGL Renderer

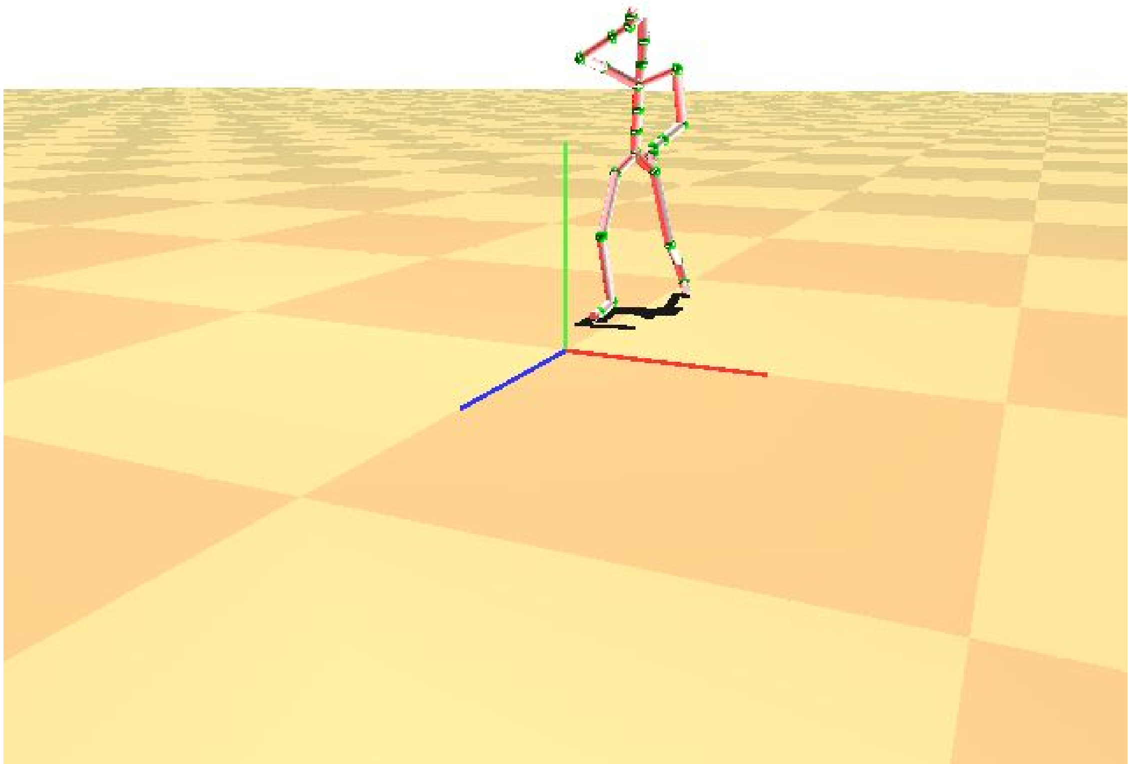

Figure 5: Motion Player

Added one more OpenGL light into the scene, effects could be checked under Bezier Quaternion folder.

3.2 Computation Time Analysis (as stated before)

3.3 Swap Buffer Improvements (as stated before)