# Computer Animation2: Motion Capture Interpolation Name: Po-Ting Kuo Email: [potingku@usc.edu](mailto:potingku@usc.edu) Coding Environment: VS 2010

# <Description of what you have accomplished> Complete functions in Interpolator.cpp and Calculate Time code in each interpolation methods. The followings are the interpolation methods that I implement. 1. Bezier interpolation for Euler angles 2.Linear interpolation for quaternions 3.Bezier SLERP interpolation for quaternions

# <Abstract> In this project, we want to implement three interpolation method: Bezier interpolation for Euler angles, Linear interpolation for quaternions, and Bezier SLERP interpolation for quaternions. Then add linear interpolation for Euler angles to compare each scheme’s efficiency and drawbacks with each other scheme.

# <Folder Index> I have five folders stored under csci520\_HW which shows the difference in four interpolation schemes. 1. Plot\_Graph 2. scheme\_excel 3. Video1\_BezierEuler\_Input: the comparison between Input && BezierEuler 4. Video2\_ SLERPquaternion\_Input: the comparison between Input && SLERPquaternion 5. Video3\_ BezierSLERPquaternion\_Input: the comparison between Input && BezierSLERPquaternion

# <Plot\_Graph> Graph1. Linear Euler versus Bezier Euler

# graph1.jpg Graph2. SLERP quaternion versus Bezier SLERP

# graph2.jpg Graph3. Linear Euler versus Linear SLERP quaternion graph3.jpg Graph4. Bezier Euler versus Bezier SLERP quaternion graph4.jpg <A report comparing the four interpolation methods: linear Euler, Bezier Euler, SLERP quaternion, Bezier SLERP quaternion.>

# LinearEuler.pngBezierEuler.png Linear Quaternion.pngBezierQuaternion.pngTime Comparison: By analyzing the four techniques consuming time, I found the time each technique take: Linear Euler < Bezier Euler< Linear Quaternion< Bezier. That means Bezier interpolation mechanism takes more time than Linear interpolation and computing in Euler is faster than in Quaternion. Efficiency Comparison: When watching the video between Bezier Euler and Linear Euler, I found there is only little difference between them. The Bezier Euler has less strange rotation then Linear Euler, which means better result. However, it still has strange rotation. And by analyzing the graph, I found the quaternion interpolation gives the better result than Euler angle. The curve produced by quaternion looks smoother than by Euler. Also, the robot will not produce strange rotation when in quaternion interpolation. Summary: Linear Euler’s curve is very sharp, which means the motion produced will change dramatically during frames. Bezier Euler has smoother curve than Linear Euler; however, still might cause strange rotation. Linear Quaternion works great. The result curve is closer to the actual motion curve. However, it still produce sharp motion. Bezier Quaternion has the best result, no sharp curve, no strange rotation and the closest to the actual curve. However, it’s time consuming is the highest. After analyze this, I kind know that why the game engine in industry often use quaternion angle for rotation. <Findings and Observations, What are the strengths and weaknesses of each technique?> I think Linear Interpolation is quickly but gives a rough interpolation and angular velocity is not smoothly. However, Bezier interpolation gives a smoother angular velocity but more computing time. 1. Euler angle Disadvantage>

# Euler angles is lack of intuition, it is easier to think the rotation about x,y,z axes. Euler angles has Gimbal lock problem. Euler angles representation is redundant since the matrices will still have zeroes for indices. 2. Euler angle Advantage Redundancy is less than rotation representation Homogeneous matrix has the ability to represent all the other basic transformations. 3. Quaternion Disadvantage Quaternion computation is more complex and time-consuming 4. Quaternion Advantage 1. Non-singular expression 2. Can represent in 4D space. <Include any additional findings, interesting observations or insights gained during this homework.> It is worth noted that when doing SLERP, my function calculate the CosTheta between two quaternion and if the degree is more than 90 degree, then I make q=-q since –q is closer and this is interesting to me since this phenomenon is the same as the two distance on earth. What’s more, when computing the an and bn+1 for Bezier interpolation, I looked up the Rick Parent's "Computer Animation" book for detailed implement description, and I am very excited that I fulfilled his paper and it actually work!

# <Extra Credit> I have used std::clock(); function to calculate the time that each interpolation scheme spend and when finished, the clock time minus the start time will give me the calculating time.

<Reference Material>  
1. Csci520 course slides  
2. Rick Parent's "Computer Animation" book 