CSCI 6608 Advanced Computer Animation
Assignment 1
October 3, 2018

Johna Latouf

How the curves are interpolated:

1. The getFileInfo() function reads the specified .dat file and arranges the time values into the q_times array and quaternion values into q_points. These points are rearranged by box into q_points_by_box. The box quads are also drawn based on the file information.
    a. The number of frames are calculated based on the time values in the file and the FRAME_RATE constant.
    b. From here, slerpKeyframes(), curveKeyframesBO(), and curveKeyframesCR() are called

2. **slerpKeyframes()**
   For each box b, for each "point" or keyframe p in the animation (each entry in the .dat file), quaternions q and q+1 and t, the location of each frame in the time between points, are used to determine that frame's quaternion with the q_slerp() function using this formula, taken from the class slides:
   phi = arccos(q1•q2)
   Q1 = (sin((1-t)*phi)) / (sin(phi)) * q1
   Q2 = (sin(t*phi) / (sin(phi)) * q2
   Return Q1 + Q2

3. **curveKeyframesCR()**
    a. For each box b, for each keyframe p, quaternions q0 =q[p-1], q1 = q[p], q2 = q[p+1], and q3 = q[p+2] are used to interpolate a Catmull Rom curve.
    b. The control points for each segment are q1, q1plus, q2minus, and q2 are put into an array all_qs
       q1plus = q1 * $(q0^{-1} * q2)^{1/6}$
       q2minus = q2 * $(q1^{-1} * q3)^{-1/6}$
       The blending function is taken from slide 82

$$\mathbf{q}(u) = \mathbf{q}_0 \prod_{i=1}^{n} \exp\left(\boldsymbol{\omega}_i b_{i,n}^{+}(u)\right)$$

   w_i = log(all_qs[i-1]$^{-1}$ * all_qs[i])

   The blending $b_{n,k}$ values are summed from i to n, multiplied with w_i and the product of each is multiplied with q0 for each frame

4. **curveKeyframesBO()**

The blending for each Bessel Overhauser frame uses the same formula as Catmull Rom (3. c.), but q1plus and q2minus are calculated using the quaternion version of the formulas from slides 96-97

Control points:
$q1plus = q1 * t_i1^{(u2-u1)/3}$
$q2minus = q2 * t_i2^{-(u2-u1)/3}$

t_i1, t_i2, and their plus and minus half values are quaternions:
$t_i1 = (t1plushalf^{(u1-u0)} * t1minushalf^{(u2-u1)})^{1/(u2-u1)}$
$t_i2 = (t2plushalf^{(u2-u1)} * t2minushalf^{(u3-u2)})^{1/(u2-u1)}$
$t1plushalf = (q1^{-1} * q2)^{1/(u2 - u1)}$
$t1minushalf = (q0^{-1} * q1)^{1/(u1 - u0)}$
$t2plushalf = (q2^{-1} * q3)^{1/(u3 - u2)}$
$t2minushalf = (q1^{-1} * q2)^{1/(u2 - u1)}$

$(u_{i+1} - u_i)$ = distance between q_time[i+1] and q_time[i]

b. Because the first and last points cannot be included in the Catmull Rom and Bessel Overhauser curves, the first few frames are not animated

5. Each group of interpolated quaternions is stored in an array. Each time the timer loops to a new frame, a cur_t variable holds the current frame number. Slerp, Catmull Rom, or Bessel Overhauser (depending which state the user has chosen using keys s,c,b) quaternions are converted [1] to axis rotations:

With a normalized quaternion q = [s:**v**]:
Rotation Angle = 2.0 * arccos(s) * 180/PI
Rotation x = v.x / sqrt($1 - s^2$)
Rotation y = v.y / sqrt($1 - s^2$)
Rotation z = v.z / sqrt($1 - s^2$)

and applied to the boxes using GLrotate. Cur_t will restart when it reaches the end of the animation.

Uniform 4-10 Dimension 2 Plots:

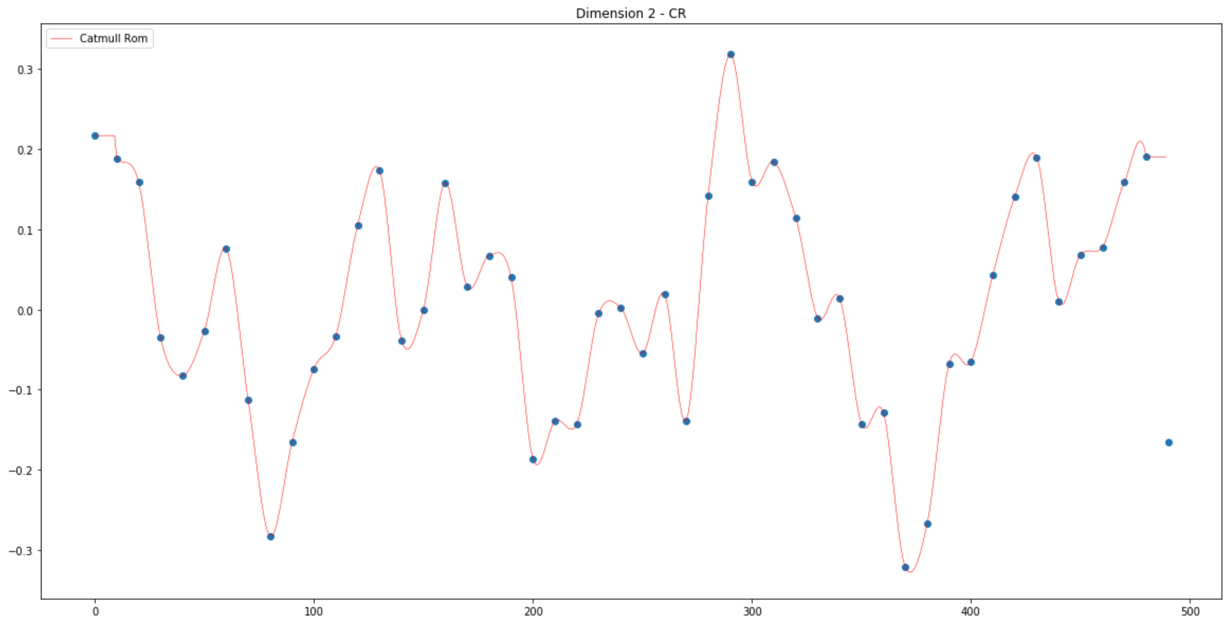

Figure 1 Uniform catmull rom interpolation

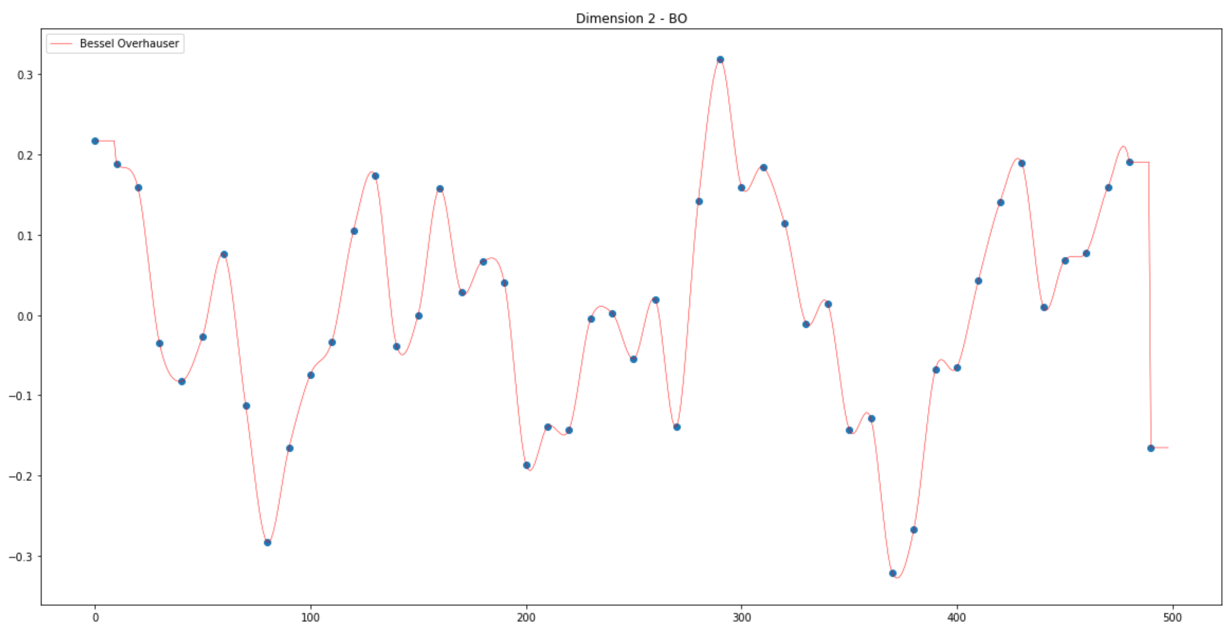

Figure 2 Uniform Bessel-Overhauser

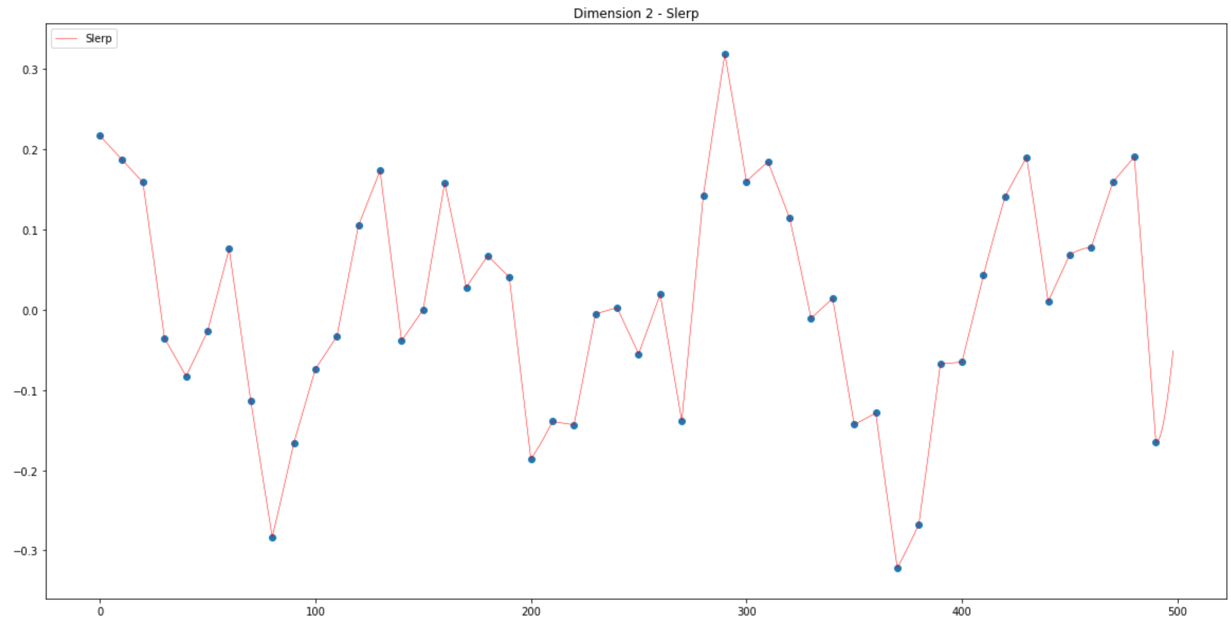

Figure 3 Uniform Slerp

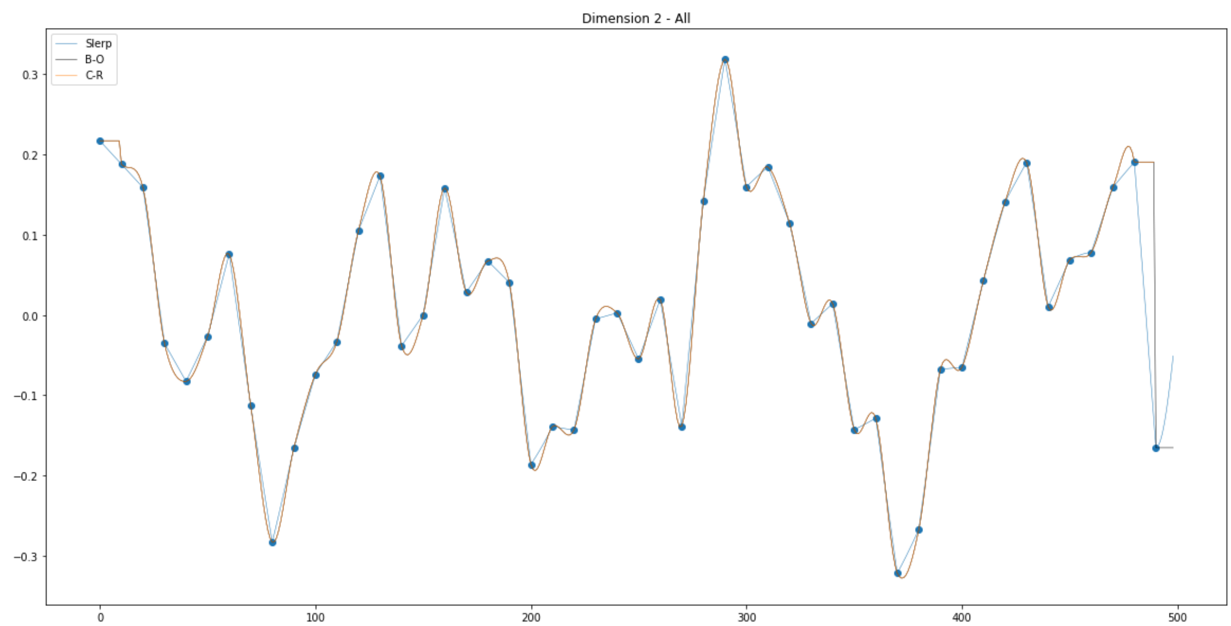

Figure 4 Uniform, all interpolations

Uniform 4-10 Dimension 16 Plots:

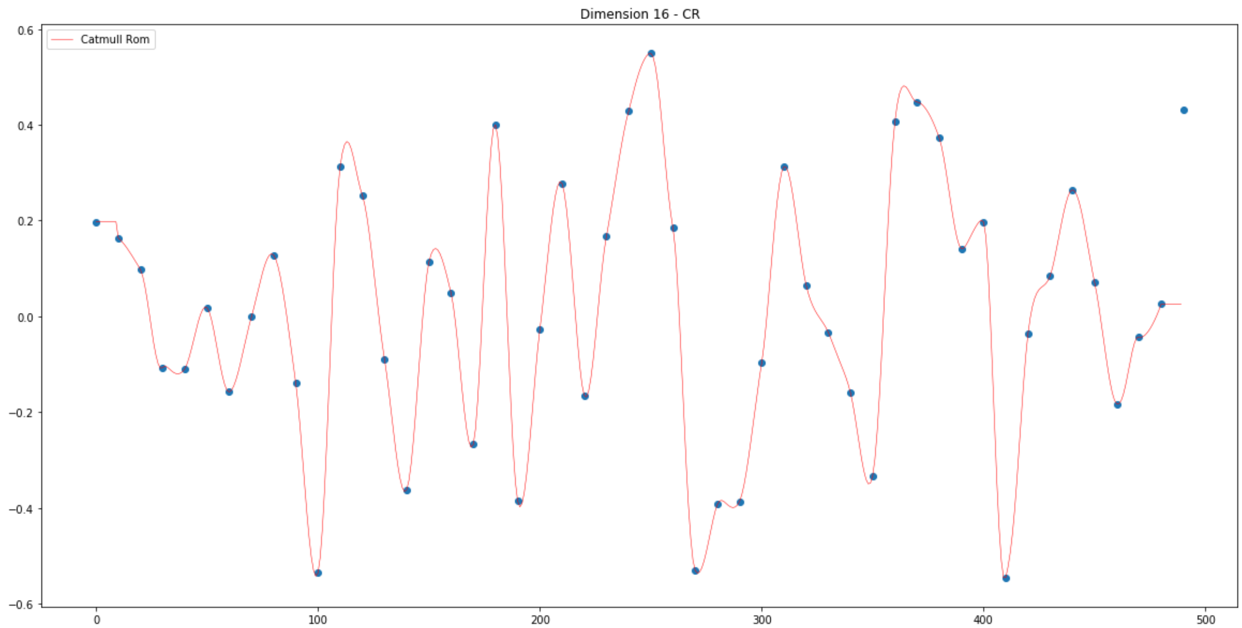

Figure 5 Uniform Catmull Rom

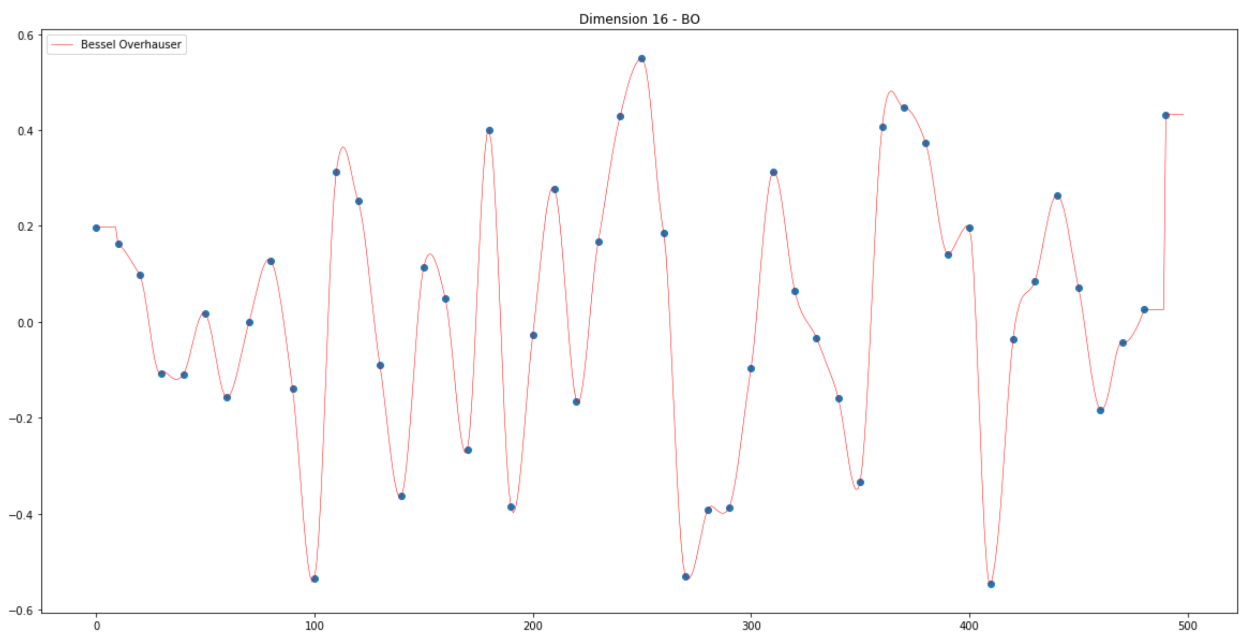

Figure 6 Uniform Bessel Overhauser

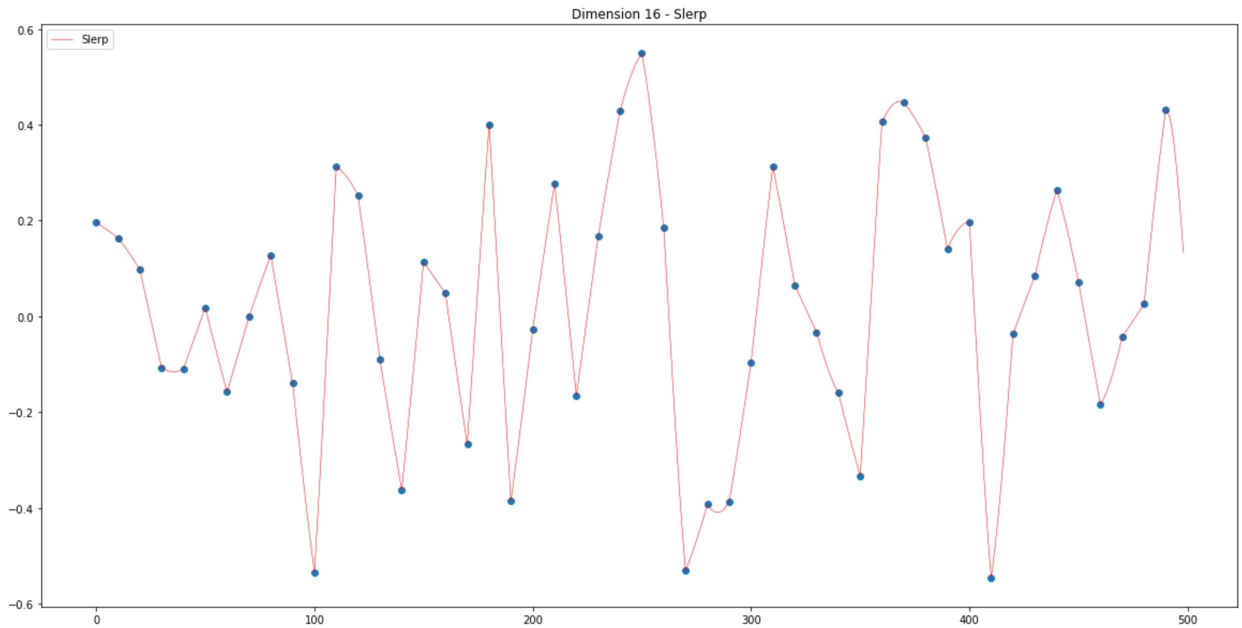

Figure 7 Uniform Slerp

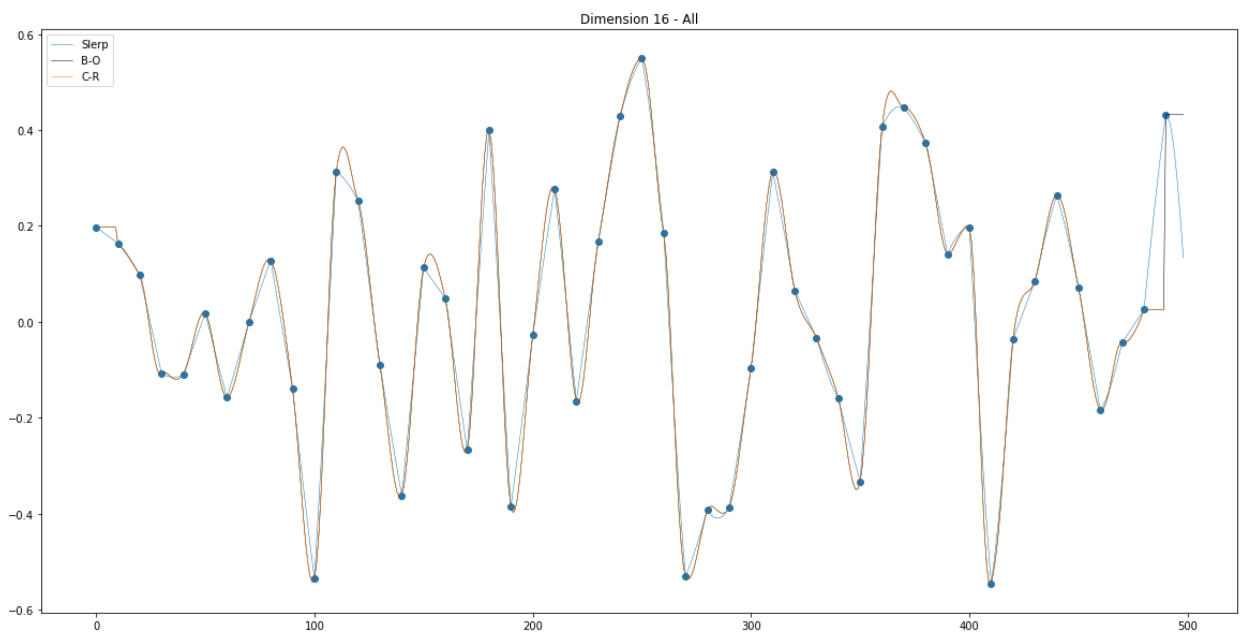

Figure 8 Uniform All

Non-Uniform 4-10 Dimension 2 Plots:

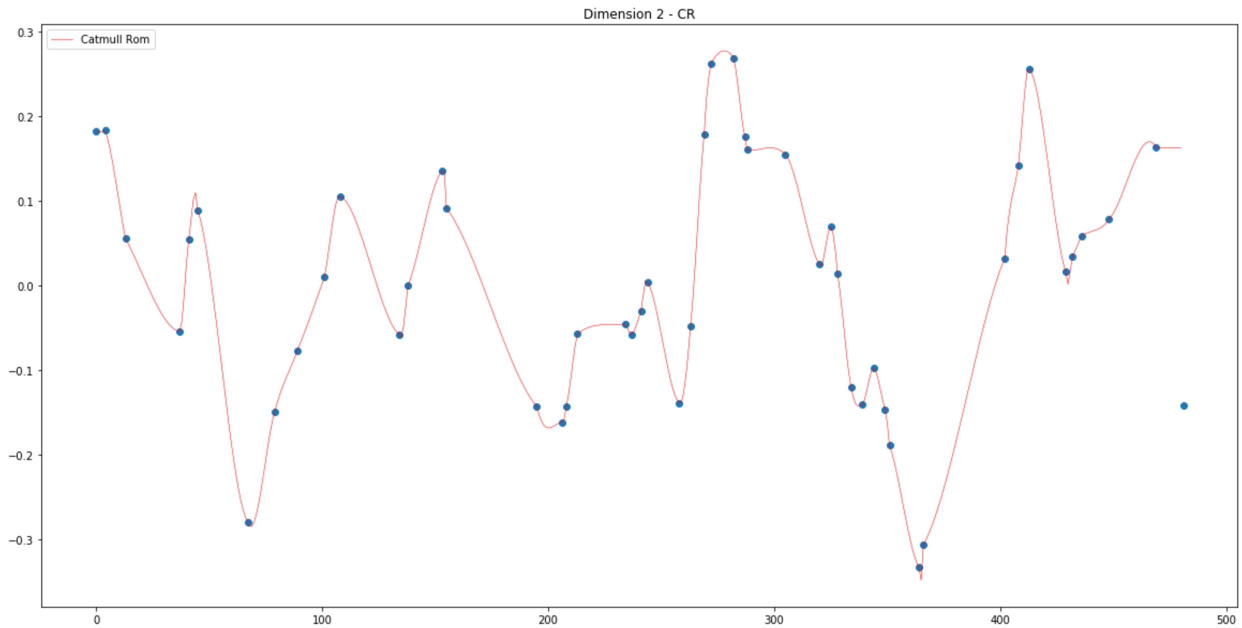

Figure 9 Non-Uniform Catmull Rom

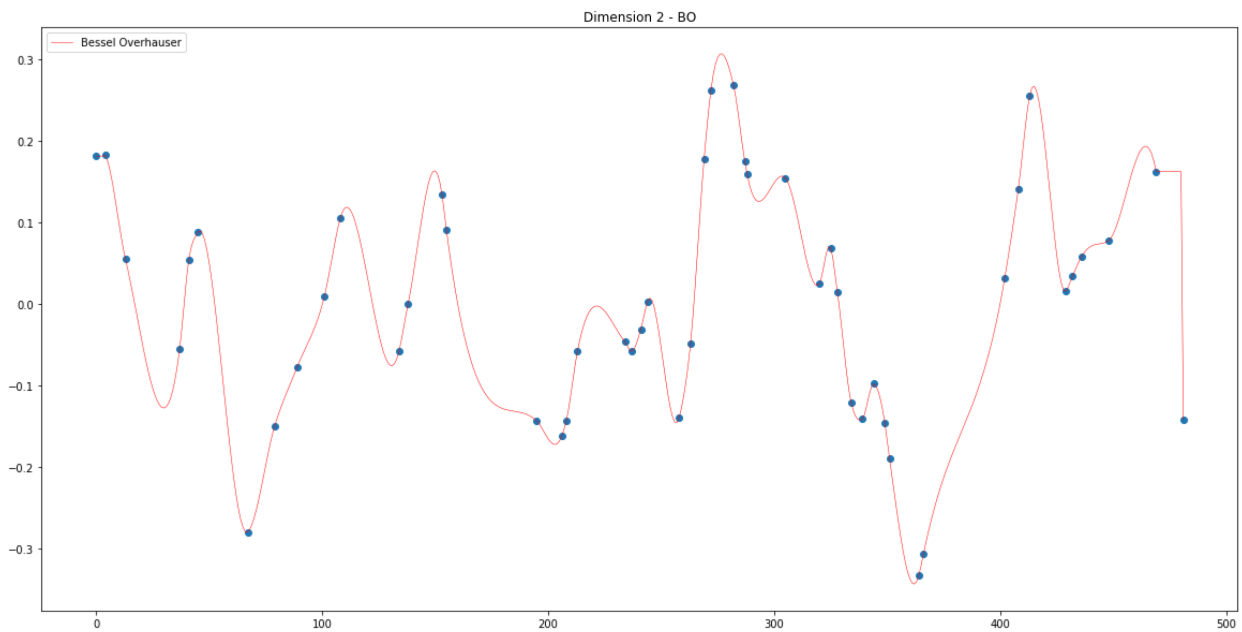

Figure 10 Non-uniform Bessel-Overhauser

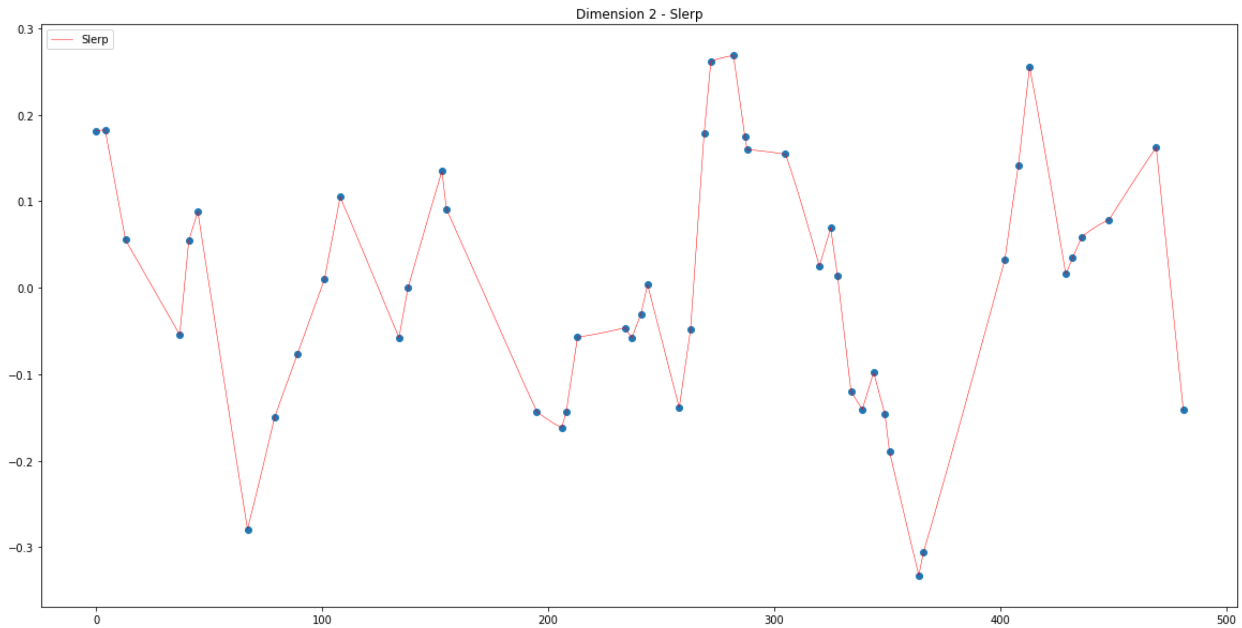

Figure 11 Non-Uniform Slerp

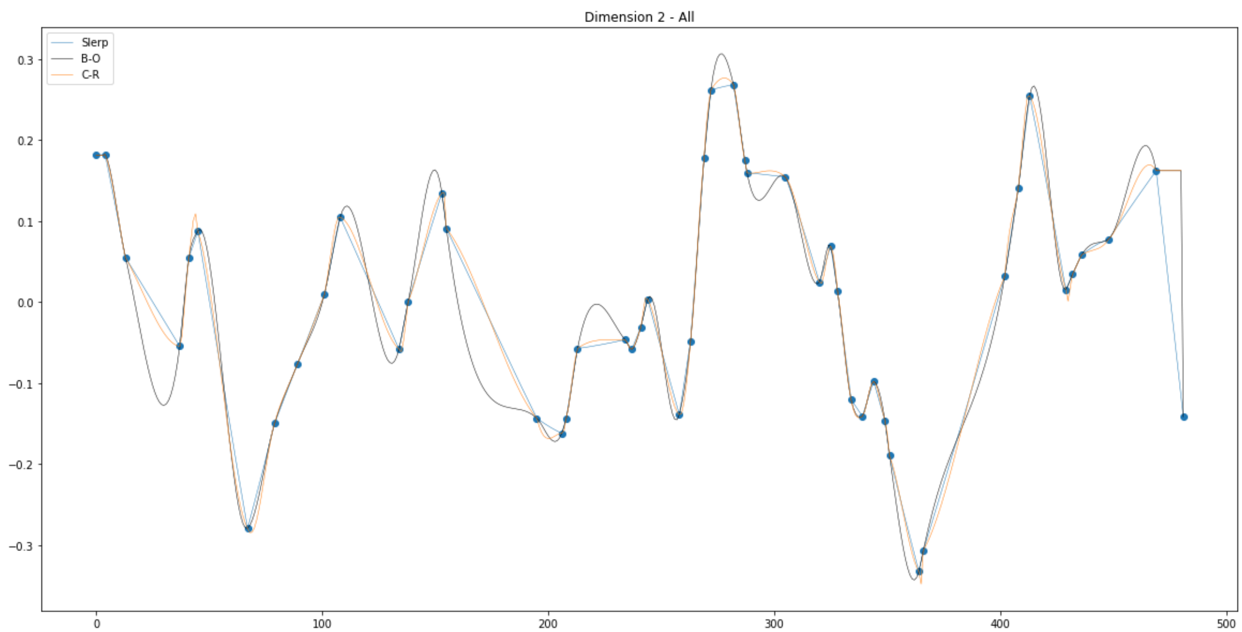

Figure 12 Non-uniform all interpolations

Non-Uniform 4-10 Dimension 16 Plots:

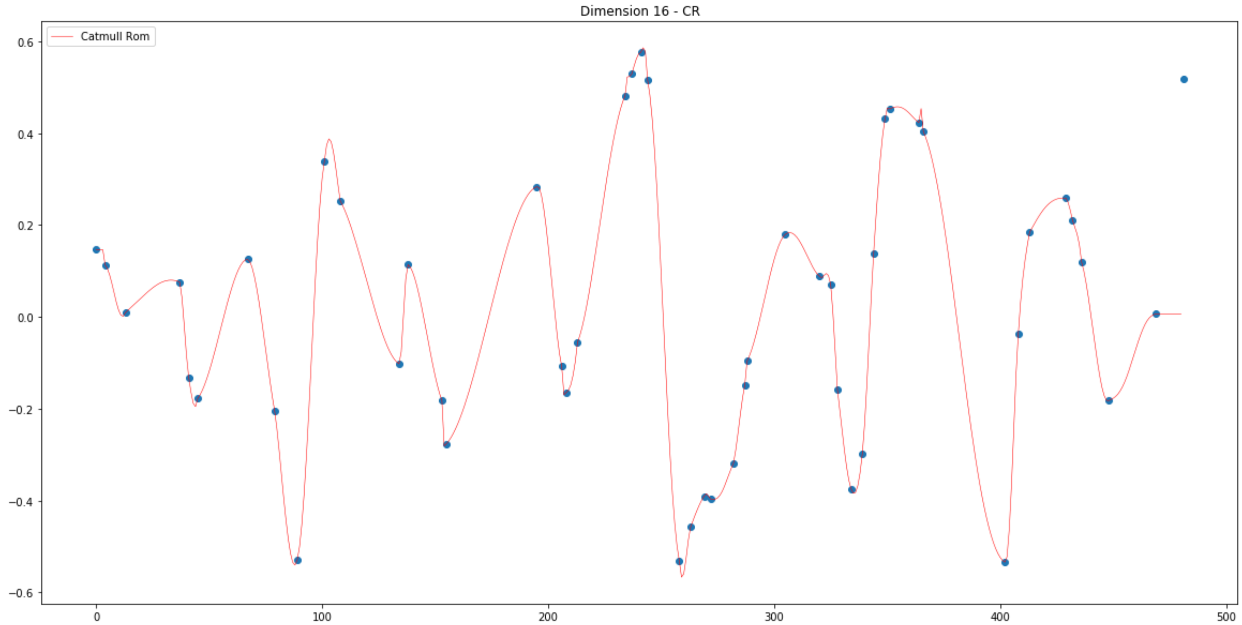

Figure 13 Non-uniform catmull rom

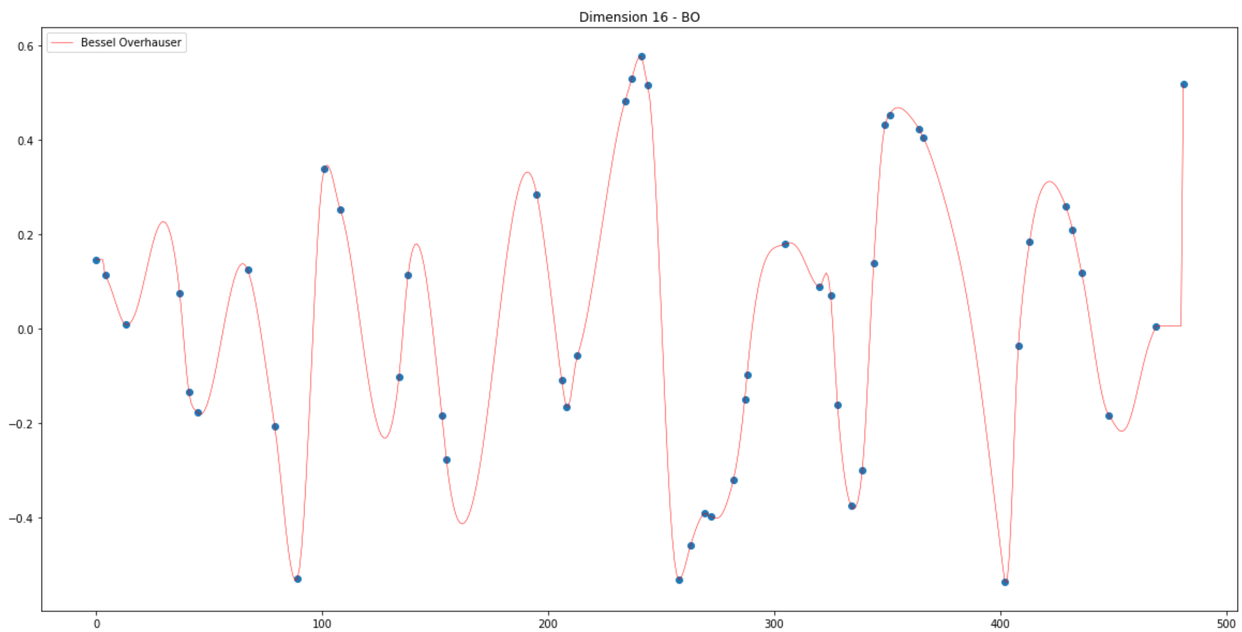

Figure 14 Non-uniform Bessel Overhauser

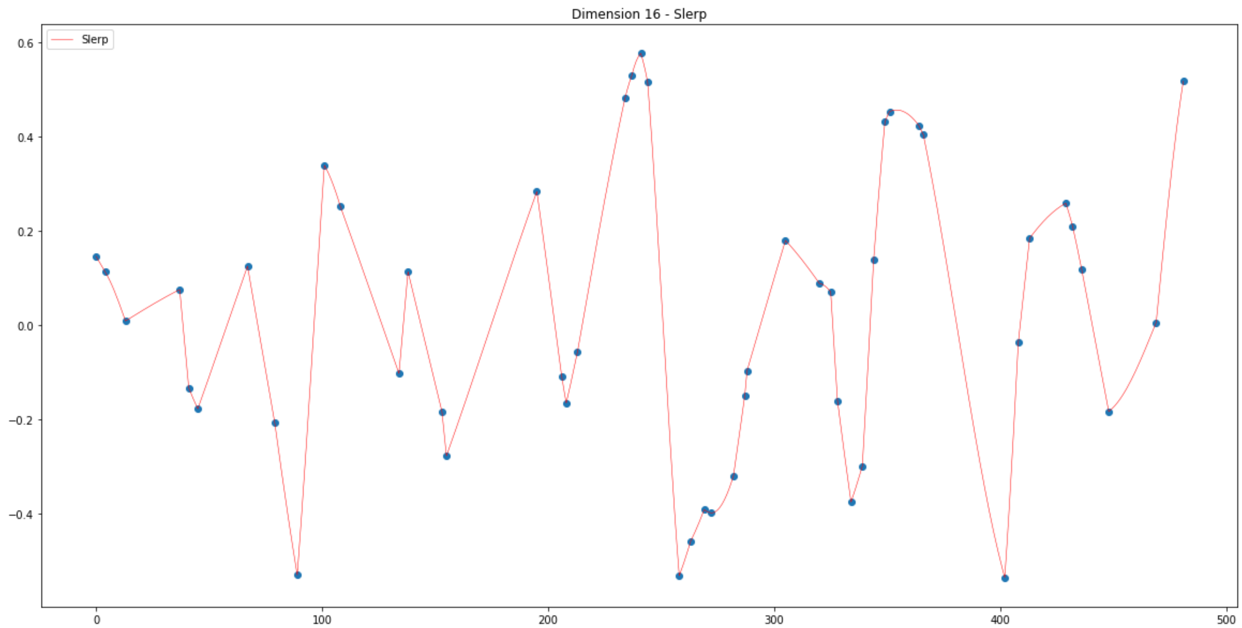

Figure 15 Non-uniform Bessel Overhauser

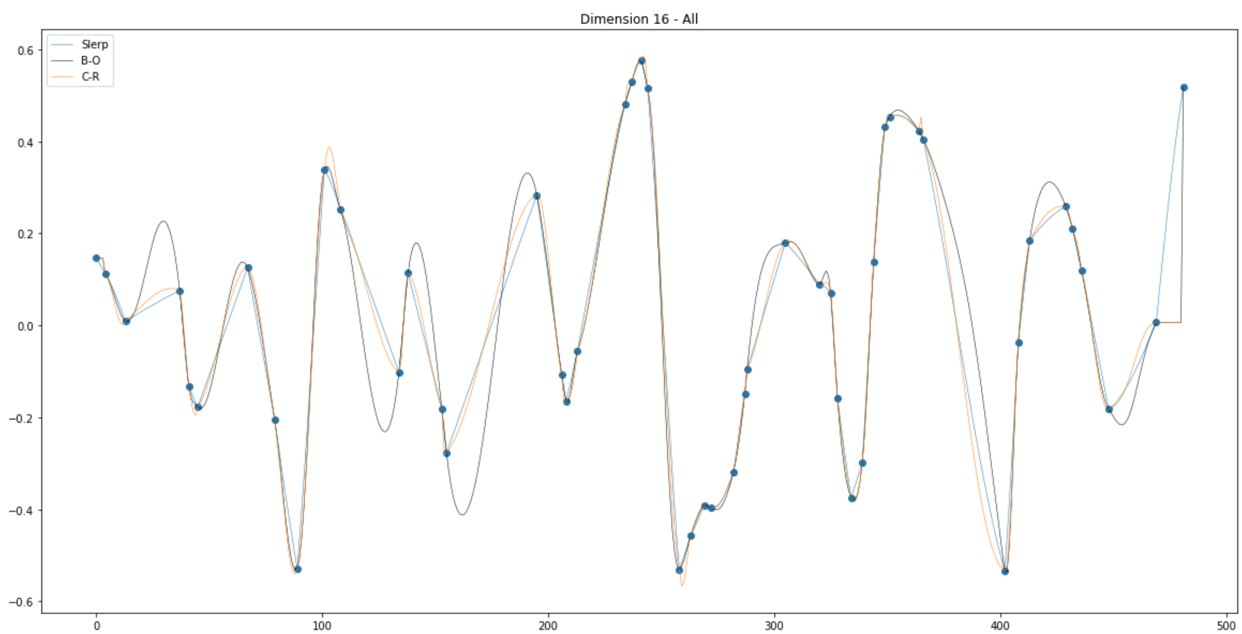

Figure 16 Non-uniform all interpolations

References

1 M. J. Baker. 2017. "Maths - Quaternion to AxisAngle". *Euclidian Space*. Retrieved September 30, 2018. Available: http://www.euclideanspace.com/maths/geometry/rotations/conversions/quaternionToAngle/