Method S4. Dynamically Simulating Ligament Fibers in Maya

Begin with a Maya 2016 (Autodesk, San Rafael, CA, USA) scene including an animated anatomical joint. Create a polygonal plane with 10 width divisions and 50 height divisions and rename it "lig1." Add a float-type attribute to lig1 called "length." Translate, rotate, and scale lig1 to bring it to a ligament fiber's attachment sites on two mesh models; editing lattice points can also help to achieve the proper orientation. Although lig1 has width for convenience of visualization, length measurements will only be taken from a columnar strip along its center.

First, select lig1 and create an nCloth. Then, select both bone models and create Passive Colliders. Viewing the plane in vertex mode, select the first row of vertices and create a weld-type point-to-surface constraint to the bone mesh they are attached to. Repeat for the last row of vertices. Select the second row of vertices and create a spring-type point-to-surface constraint of strength 6. Repeat for the penultimate row of vertices.

The following nCloth collision and dynamic properties were used in the present study: bounce = 0, friction = 0.5, stickiness = 0, stretch resistance = 900, compression resistance = 200, bend resistance = 0, bend angle dropoff = 0, shear resistance = 0, restitution angle = 360, restitution tension = 1000, rigidity = 0, deform resistance = 0, input mesh attract = 0, rest length scale = 1, bend angle scale = 1, mass = 1, life = 0, drag = 0, tangential drag = 0, damp = 50, stretch damp = 0, scaling relation = link, local force = <<0,0,0>>, and local wind <<0,0,0>>. The nCloth was also set to ignore solver gravity and wind.

Once a ligament fiber is modeled as above, the following MEL code can be used to create a new expression in the Expression Editor:

```
{
float $start[];
float $end[];
float $lengthVal[];
float $length;
float $total;
int $i;
int $j;
for ($i=4; $i < 555; $i = $i+11) //the first vertex queried is vertex four, the fifth in the first row of the plane. By incrementing by 11 vertices, we move to the fifth vertex in the following row.
        {
        $j = $i+11; //the fifth vertex in the next row
        $start = `pointPosition ("lig1.vtx["+$i+"]")`; //find the position of the fifth vertex in the first row
        $end = `pointPosition ("lig1.vtx["+$j+"]")`; //find the position of the fifth vertex in the following row
        $lengthVal[0] = $start[0]-$end[0];
        $lengthVal[1] = $start[1]-$end[1];
        $lengthVal[2] = $start[2]-$end[2];
        $length = `mag <<$lengthVal[0],$lengthVal[1],$lengthVal[2]>>`; //find the distance between the vertices
        $total = $total + $length; //sum the whole column of vertex distances to find the total fiber length
        }
print $total;
setKeyframe -at length -v $total lig1;
}
```

When the animation is played, this expression will check for the length of an internal column of vertices in the plane lig1, and will keyframe that length at each key in the keyset.