Jared Kengla / VSFX 350 / Fall 2017 / 7 November

Project 1 - Jacko Complex Scene Resubmit / Houdini 16.0.671

Average render time: 17 hrs/frame local Resolution: 1280x720
Samples: 3:3 Min/Max Rays: 2/10
Noise Level: 0.01 Lights: 1 point x 13 instances, 2 grid, 1 sphere
Global Quality: 2 Diffuse Quality/Limit: 1.5/3
Refraction Quality/Limit: 2/4 Reflection Quality/Limit: 1.5/4
SSS Quality/Limit: 6/3

Complexity of Geometry: 4,300 points; 16,600 vertices; 4,100 primitives & polygons; 39 packed geos

Description:

This project involved creating a porch-full of jack-O'lanterns with variation by instancing to points.

Challenges I faced in the process included certain quirks that I worked around but do not quite understand. The materials on the instanced pumpkins only rendered when the original pumpkin was visible, so I left it visible and turned off its render visibility. The point() function was not properly referencing the foreach_begin node within an instance context for varying properties on the instanced lights, so I created another foreach loop within that context. Subsurface scattering had me scratching my head, until I realized all I needed was a normal node on my pumpkin geometry (after the Boolean had its way), and it is definitely expensive to render. The UVs around the mouths also gave me trouble.

Jared Kengla / VSFX 350 / Fall 2017 / 7 November

Project 1 - Jacko Complex Scene Resubmit / Houdini 16.0.671

The Method I used to construct this jackolantern scene involved primarily a copytopoints node within a foreach loop as shown in Figure 1. I varied certain attributes by creating attributes on points and referencing those in other geometry contexts using the point() function.

For example, the facial features were created in one geometry context and referenced point attributes to vary per pumpkin before performing a boolean operation to subtract from the main pumpkin body. This reference is displayed in Figure 2 to randomly vary the shape of the teeth between triangular prisms and rectangular prisms. Other varied attributes included scale, squash and stretch, the bend of the mouth and stem, stem rotation, and eye scale all achieved using the same method.

Lights were instanced to points using an instance container referencing the points in the instanced_pumpkins context. Here the point() function was not properly referencing outside of the instance container, so I recreated a foreachloop and used the point() function to reference the local loop (Figure 3). The lights are referencing an asad light shader in the SHOP context, although I could not get this to properly dim the lights according to the point order like I wanted.

Other architectural features like the wall slats and the porch involved copystamping with an expression based on the height and desired spacing in between the copies.

Figure 2:

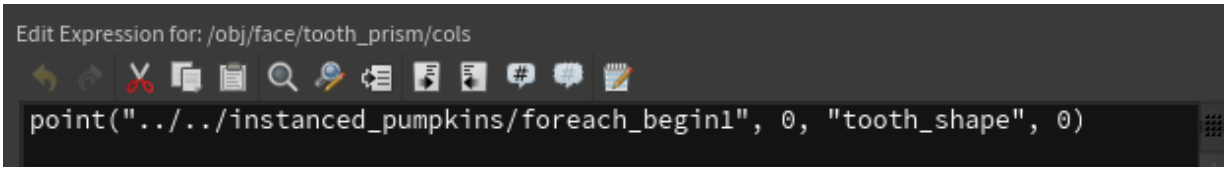

Figure 1:

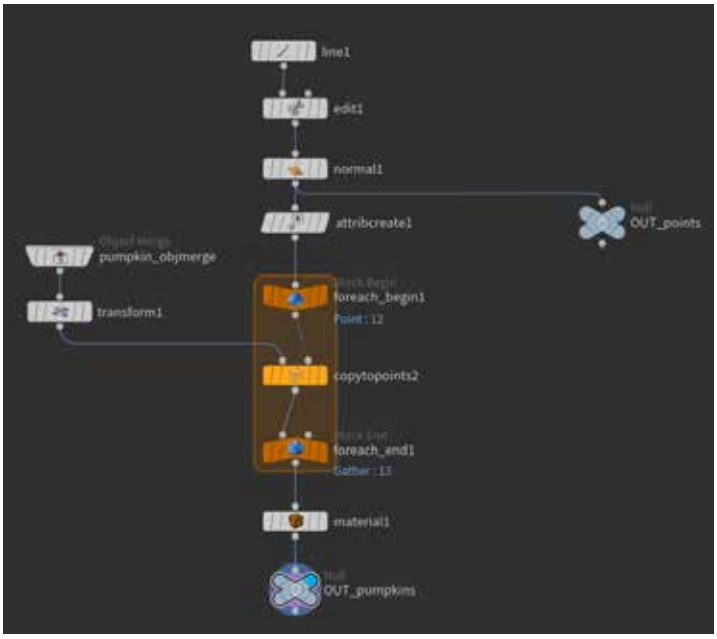

Figure 3: