Light Bulb | Breakdown Document | Yanni He

Houdini Version: 19.5.605

Render Statistics

Renderer: Redshift

Frame count: 120

Avg. render time: 5 min/frame(render farm)

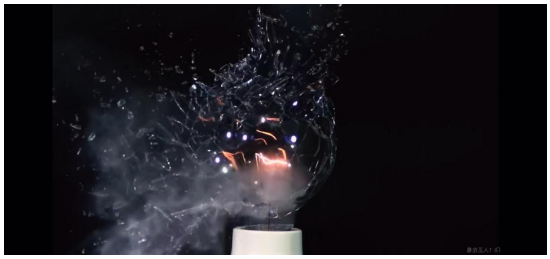
Image Resolution: 1280 x 720

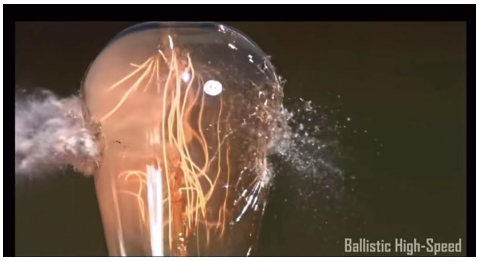
Lights: area light(1), spots light(2), skydome light(1), sun light(1)

Sampling: Reflection-4 Refraction-6 Volume-2 Transparency-16 GI-4

This project is based on two videos of shattered light bulbs.

Reference:

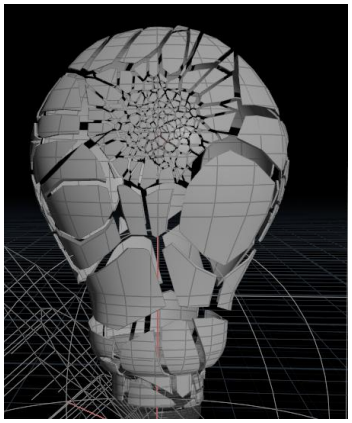
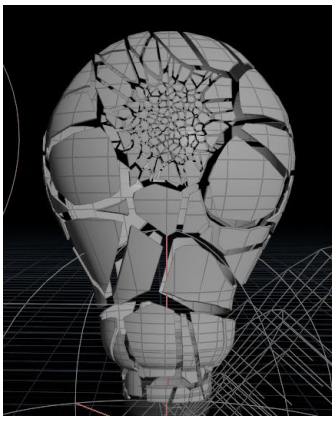
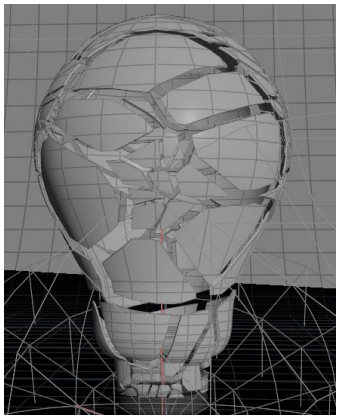


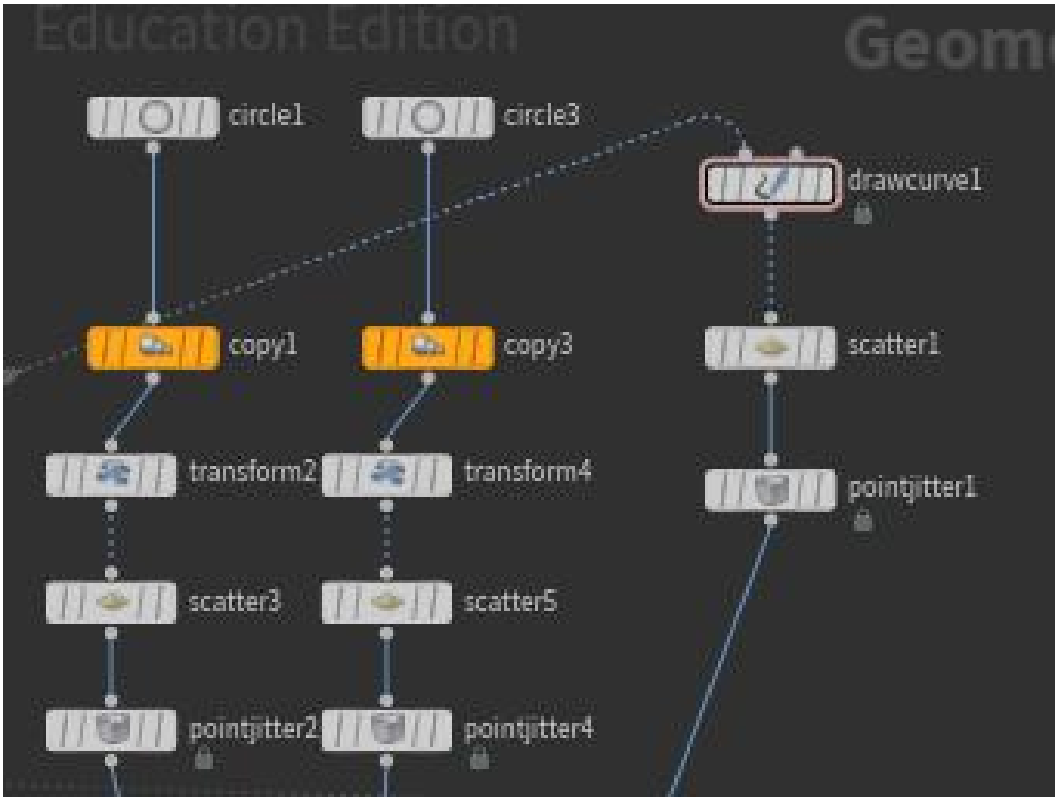


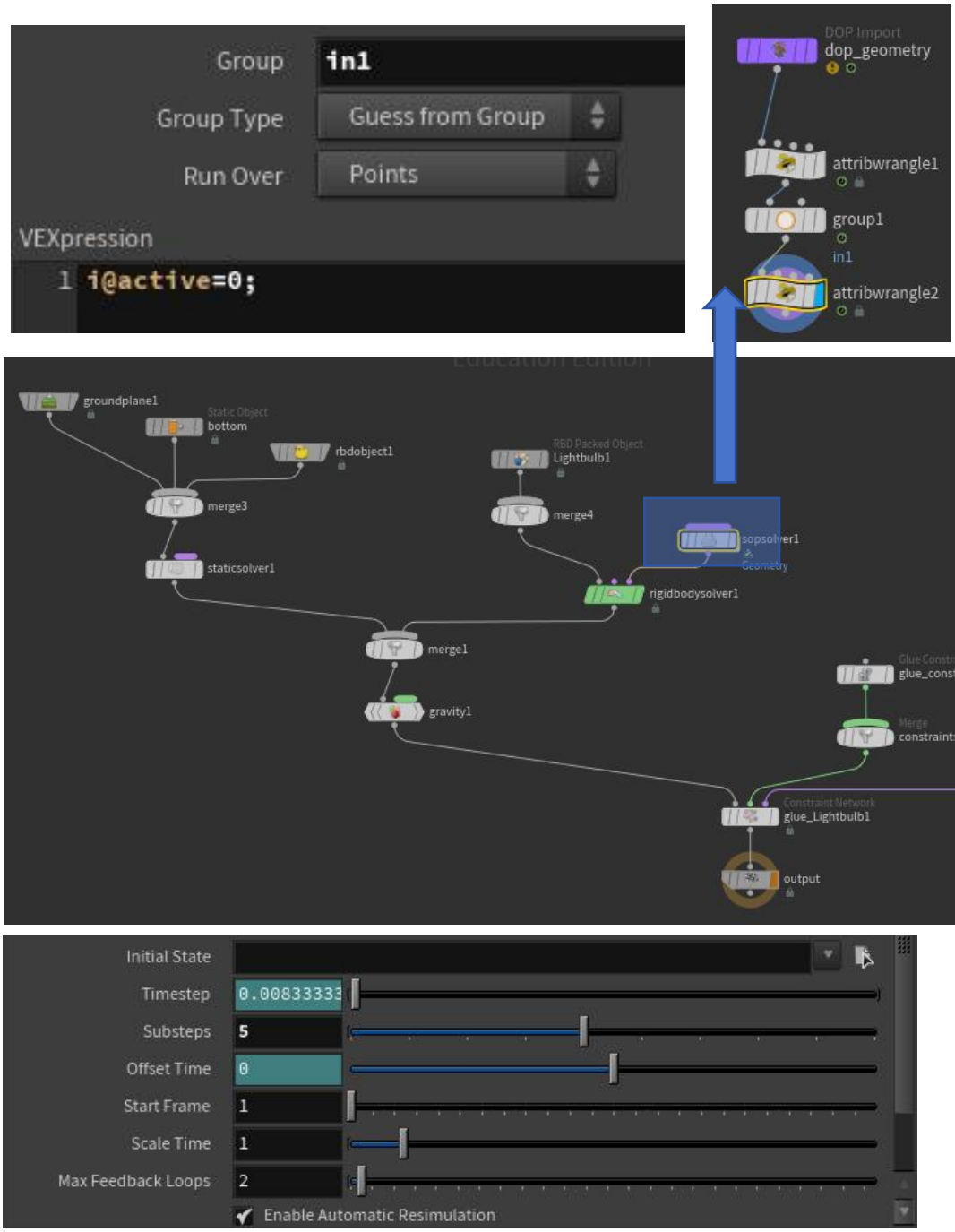
All the work was done in Houdini, utilizing simulation modules such as RBD, Pyro, and Vellum. The purpose of creating this project is to proficiently master the fracture/destruction and Pyro modules in Houdini and recreate real-world references.

Part 1 - RBD - Set Up

o set up the destruction, I mainly used draw curve and circle node and scatter the points on the geometry. Draw curve node can customized the voronoifracture very well. Customizing the range and size of the fragments allows for an accurate recreation of the reference, where the bullet passes through the middle of the light bulb, initiating different sizes of shattered glass from the impact point.

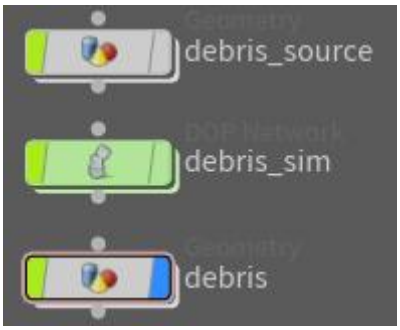




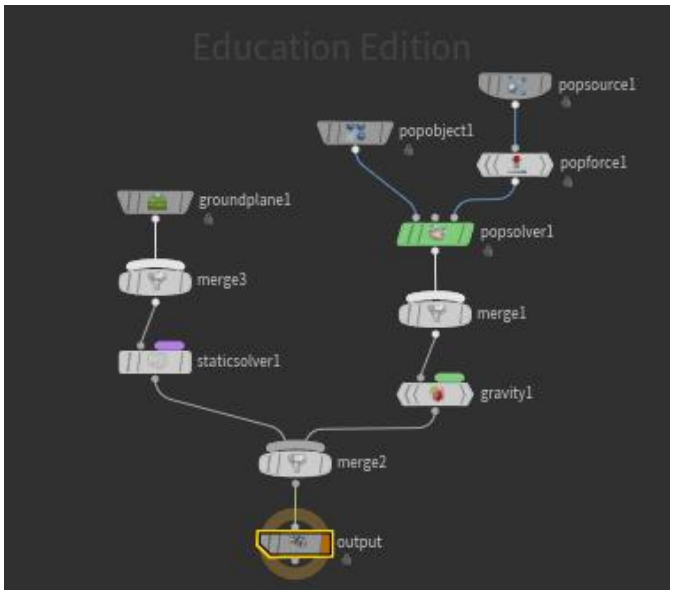


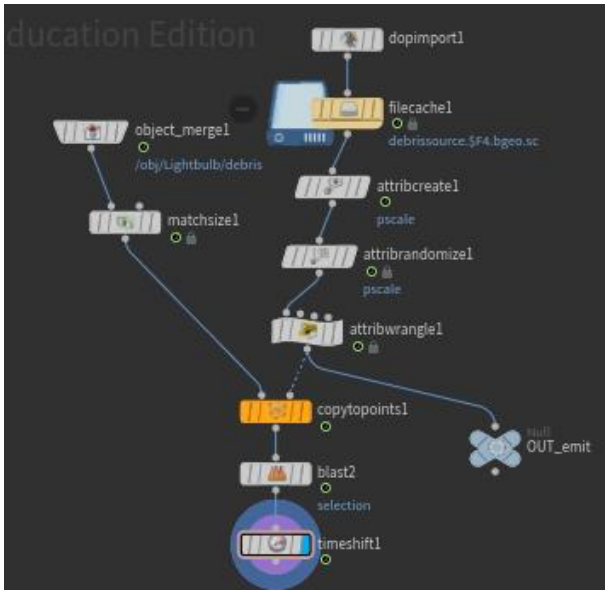


Due to the slow-motion style of the video reference, I set the substeps to 5 for using retime later. In the DOP network, I set the light bulb as one RBD packed geometry(also glued boject), and set bullet as rbdobject that connected with static solver.

In the sop solver, I grouped those points that would not be affected by destrcution and set their active attribute as 0.







I set only the pieces shattered in the middle of the light bulb as the debris source to avoid redundant computations.

After solving all debris points, I copied small shattered glass geometry to all points that already have various pscale and orient attribute.

Part 2 - Pyro

In this project, I primarily utilized the SOP-level Pyro solver.This workflow to be very intuitive and straightforward, making it beginner-friendly with the SOP solver.

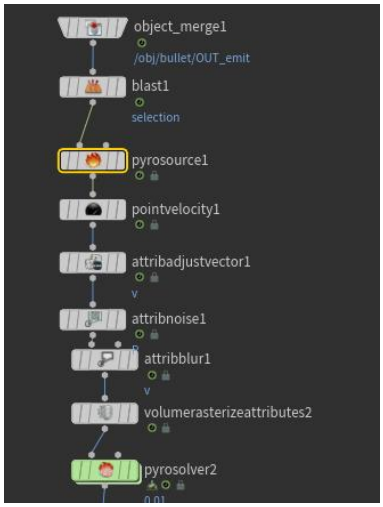
I mainly configured three different Pyro setups.

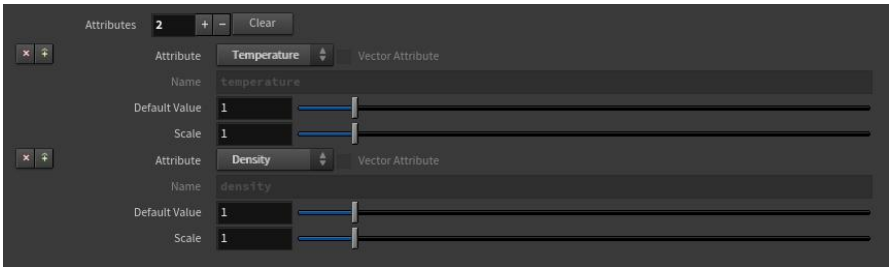
The first one involves smoke generated by the friction between the bullet and its surroundings upon impact.

The second one is smoke generated by shattered glass resulting from the collision between the glass and the bullet.

The third one involves smoke produced by the heating and burning of the filament at the end.

Pyro 1 - bullet smoke





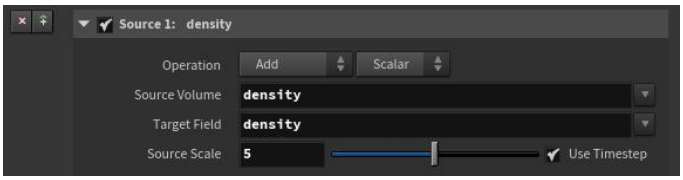
Pyro source node: only add temperature and density attribute.

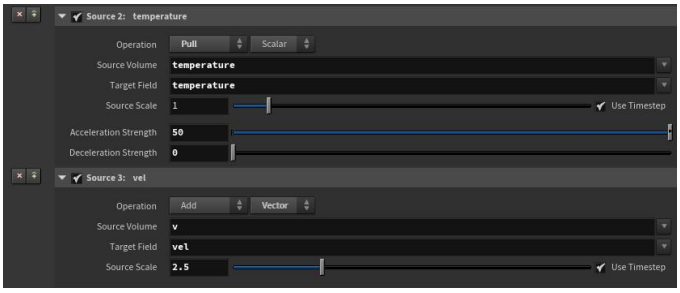
Pointvelocity & attribute adjust vector: add vel/v attribute

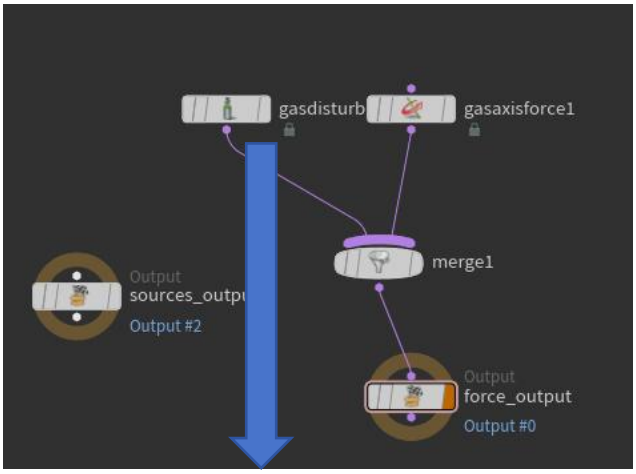
Attribute noise: add noise on the emit source

Pyro 2 - shattered glass smoke

This type of smoke generally has a higher density and, due to lighting and collision sources, it appears brighter and thicker. Therefore, I set the density to 5, and operation mode as Add. In addition, the time scale is set to 0.2 to slow down the movement of the smoke, matching the fractured speed. I also used two micro solvers: "gas axis force" for better control over the movement of the smoke, and "gas disturb" to add details to the smoke.

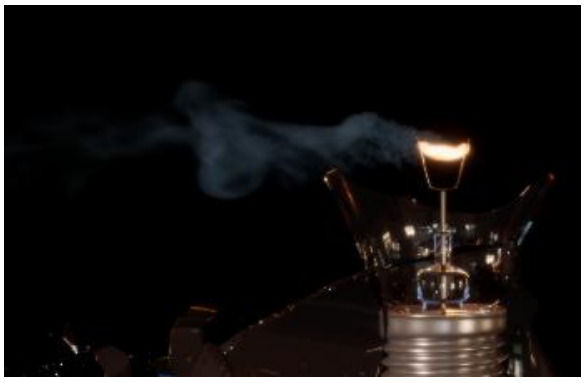




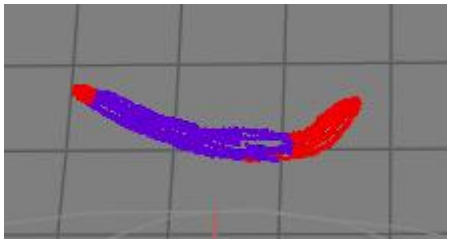


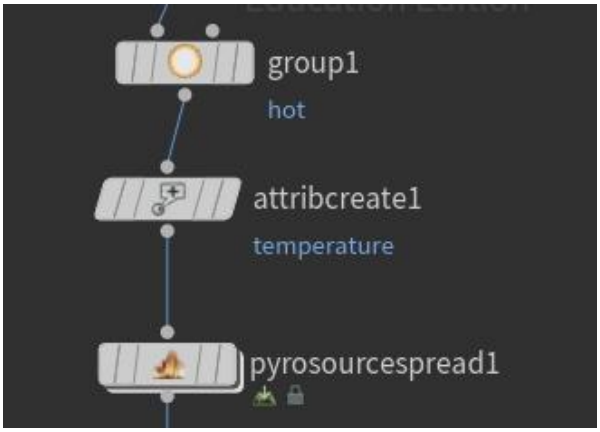
Pyro 3 - burning & flame

According to the reference, the filament will have a burning effect and fine smoke.



Burning Effect: After setting pryo source, choose points and create temperature attribute. Use Pyrosourcespread node to simulate the burn/temperature.

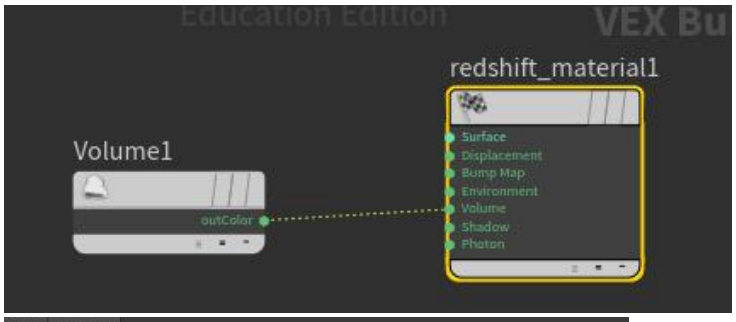


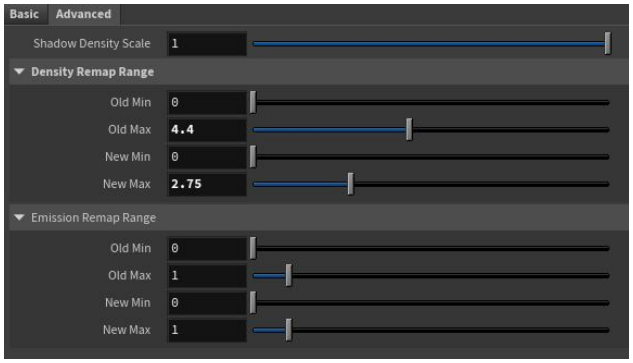


Pyro Volume shader: Flame to control the emission

Pyro - Output

The sop pyro solver have post-process module that can directly convert the pyro to VDB, I used cull volumes and resample volumes, set the voxel size scale to 2 to improve the effiency and save the space. I tested without resampling volumes and found there is no huge difference between them which means it only has little influence on the final output. If using RS volume shader - remeber to modify density remap range.





Vellum

The filament is influenced by the force of both the bullet and the smoke. Therefore, I chose to simulate the filament's motion using a Vellum Solver and RBD. The filament is divided into two parts: the upper part is directly fractured by the bullet, while the lower part remains unbroken but exhibits a soft motion.

The production methods for the two types of filaments are the same, with only parameter variations. I designated different points as "constant" and "cut" groups, set the bullet as the collision source, and imported it into the Vellum Solver for simulation.