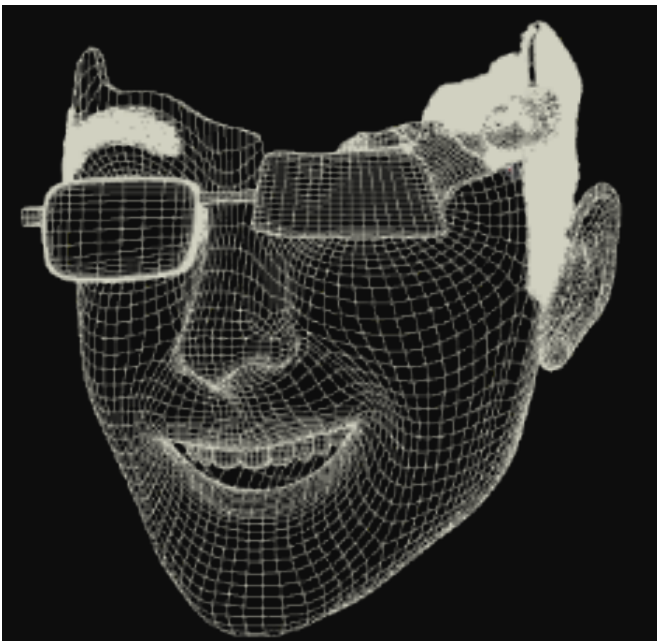


Jared Kengla / VFX 428 / Spring 2018 / May 29th

SEANMEN / Houdini 16.5.268



This project was inspired by a portrait I drew of my brother Sean. The main focus was on look development and FLIP simulations. Not responsible for boy figure or teeth model.

Particle Separation: 0.03

Sim time:

FLIP 5:25 hr

Whitewater: 5:45 hr

SEANMEN / Houdini 16.5.268

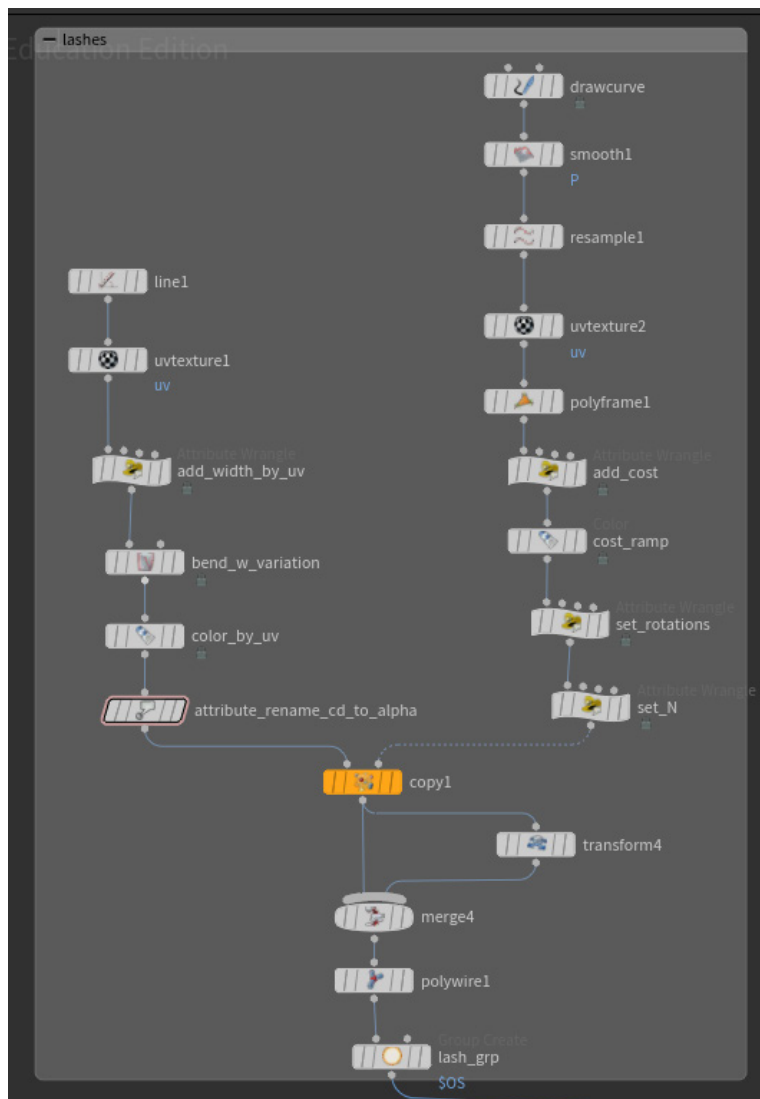
Modeling:

The face and glasses were sculpted in Maya and imported to Houdini as .obj. The boy model (Jasper) is a free download from Mixamo.

The face geometry was referenced as a VDB collision object in the FLIP sim.

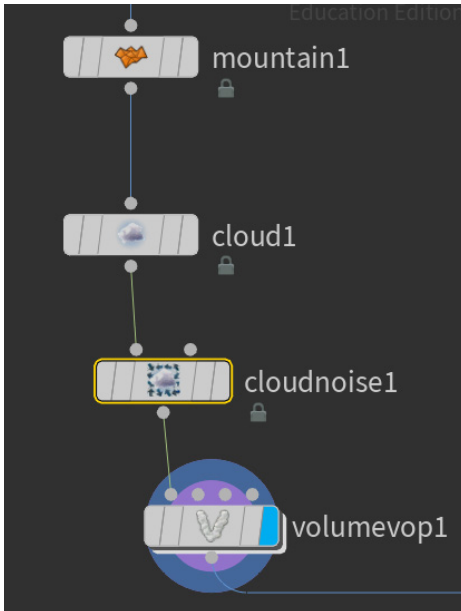
The fluid emitter was created as a simple sphere SOP.

Eyelashes were added in Houdini using polyframed curves and given variation using copystamping for bend, rotation, and scale.

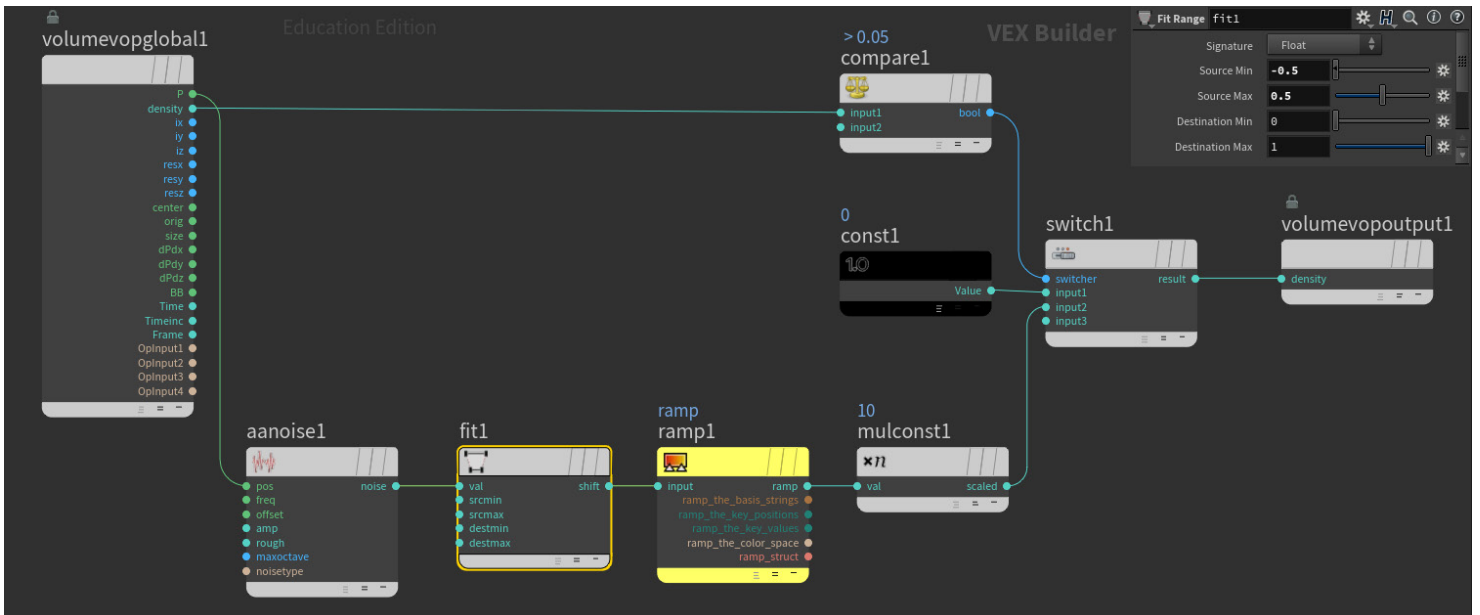


SEANMEN / Houdini 16.5.268

Cloud:



The cloud was roughly modeled and then converted to a cloud volume using the cloud node, and adding noise with cloudnoise and a volume VOP. In the VOPnet, noise is fitted between -0.5 and 0.5, referenced by a ramp, and multiplied. This resulting value is switched between a constant 0 based on a density compare node.



Texturing:

Substance Painter was used to texture the geometry, and shaders were built for Mantra.

SEANMEN / Houdini 16.5.268

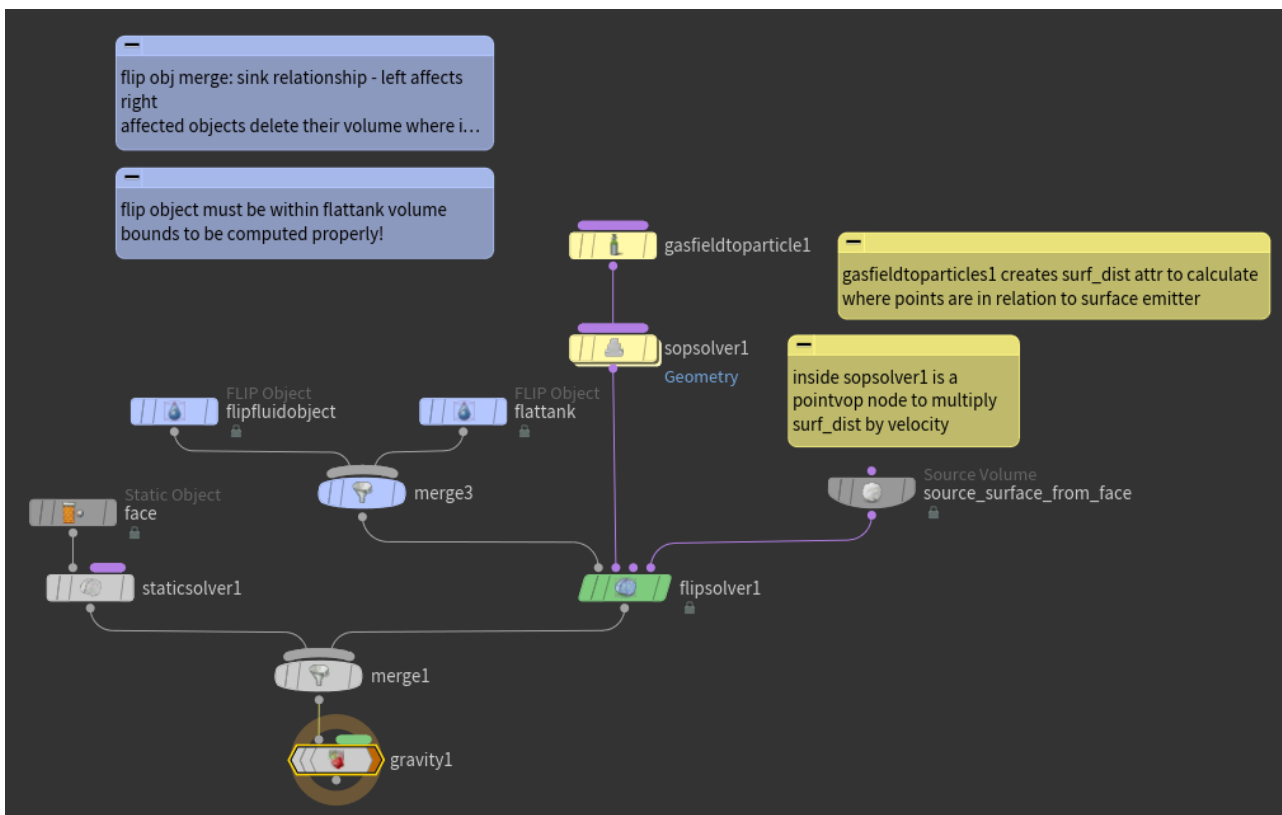
Adding velocity noise:

To add noise to the FLIP emitter, at first I animated a sphere object with the mountain node. I removed that and followed a short video called “Houdini Tip01: Super Simple Fake Airfield” uploaded by “topos” and I have pictures of that process below.

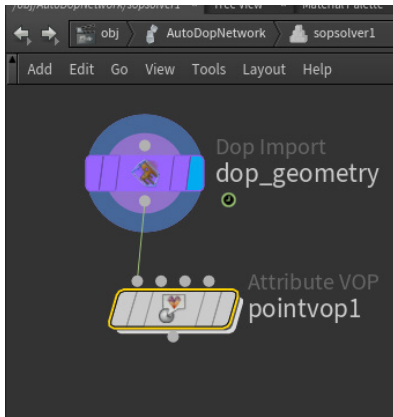
I reset the normals of the emitter geo with a wrangle node to the desired flow direction, and referenced that N attribute in the Fluid Source SOP node for the velocity, adding Curl Noise to it.

In the DOP Net, a gasfieldtoparticle node creates a field called “surf_dist” based on the surface attribute of the emitter, which gives us the particles’ distance away from the surface. Then a pointvop in a sopsolver is used to normalize that variable and multiply it with the velocity to lower the velocity as particles move away from the surface.

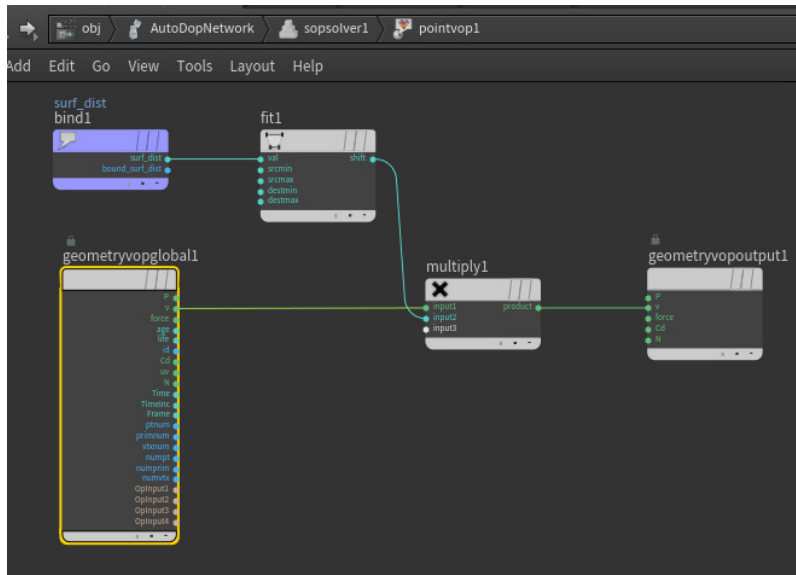
DopNet:



sopsolver1:



pointvop1:



Integrating FLIP with Oceans:

To get tips on integrating with oceans I followed the H16 Masterclass on Ocean Tools (1:30:00 to 1:45:00).

BASICS: This workflow uses the particle surfacing node to crush the simulation values at the bounds to zero and then extrude those bounds outwards. Ocean spectra maps are then applied to the surface, masking out the areas of interesting simulation events. The ocean spectra is baked out once, and the ocean masks are baked out per frame, and these are linked up to the ocean surface shader.

I used the Ocean Flat Tank shelf tool to start out and put it in with my existing DopNet. This took me some triage to figure out the wiring, since I knew that to have my FLIP interact with my flat tank, they needed to be on the same solver. I found it was best to use the solver created with the Ocean Flat Tank shelf tool since it was referencing many other parameters already. Also, for ocean flat tanks, turn off visibility of the `_fluid` node, and turn on `_extended` (`_interior` establishes the deeper volume effect and `_initial` is important for the setup).

Integrating FLIP with Oceans cont'd:

The flattank and emitter fluid objects were merged with a left affects right sink relationship. I needed to make sure the bounding volume (at first referencing my flat tank) included my emitter.

SURFACING: Once I cached the compressed fluid sim, I played with the operations in the flat tank extended object to surface it. However, I realized I did not give the simulation enough breathing room and values were crushed a bit at the edges. To get around this quick and dirty I just subtracted a VDB in that region. Also for some reason the "average position" surfacing method was dropping and glitching certain frames, so I switched to "spherical" and used a bit of a gaussian filter to smooth it out.

Whitewater:

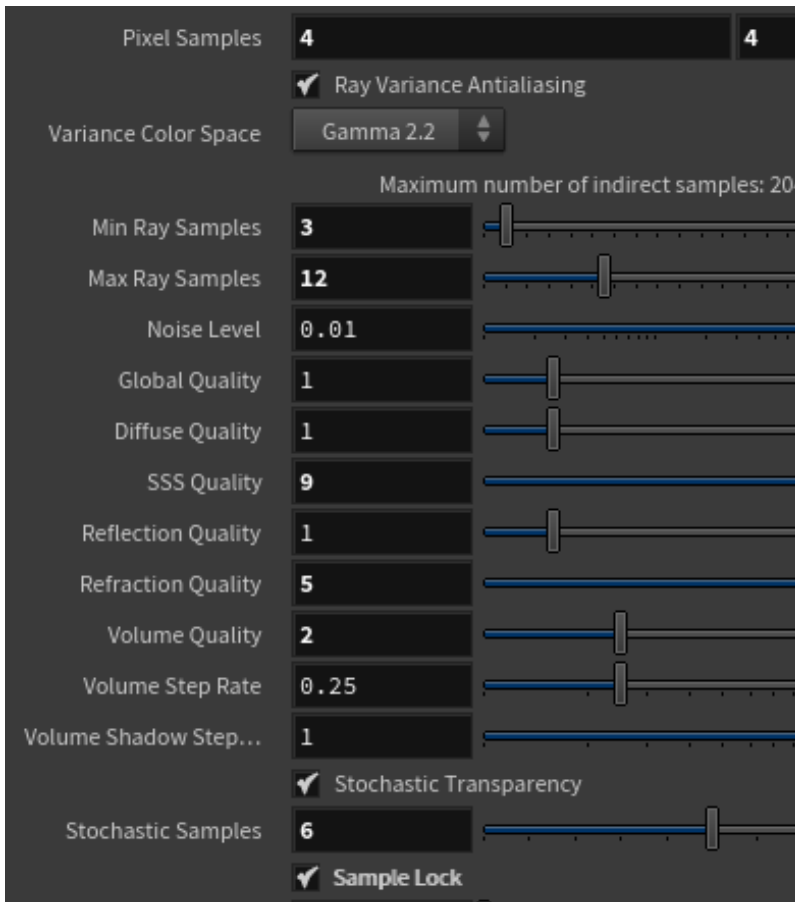
Whitewater simulation was added with the shelf tool atop of the FLIP Sim. Afterwards the pscale was tweaked to 0.2 times it's original size, and its density in the spray shader was amped up a bit.

SEANMEN / Houdini 16.5.268

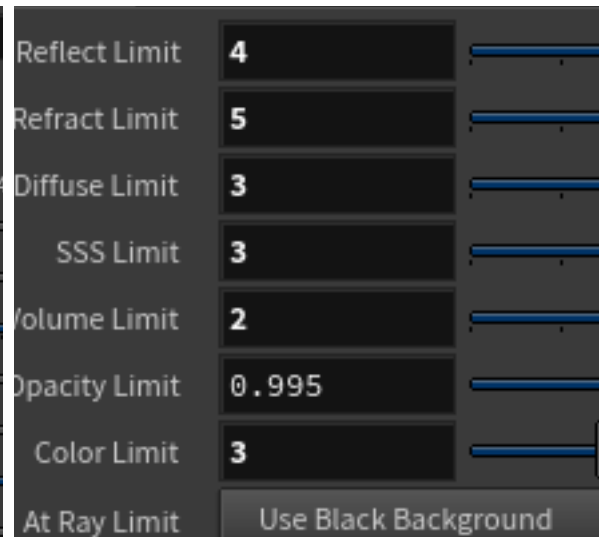
Rendering:

Video Settings:

Sampling:



Limits:



Average Render Time
(with Render Farm):

88 minutes per frame

For the High Resolution Still (1365 x 2048) Max Samples was pushed to 25, SSS Quality to 26, Refraction Quality to 22, and Stochastic Samples to 18.

To render this feasibly, using SCAD's Renderfarm, Tiled Render was enabled (under Images>Output) with a 20 by 10 division and the Index set to \$F - 1. The most taxing of those tiles took about 45 minutes to render, and still have a bit of noise.