H13 Particles for Motion Graphics

Particle and Packed Primitive Workflow

Ari Danesh
ari@sidefx.com

SIDE EFFECTS SOFTWARE

Agenda

H13 Causes the Artist to Use New Workflows

Demonstration of a Common Shot using H13 Methodologies

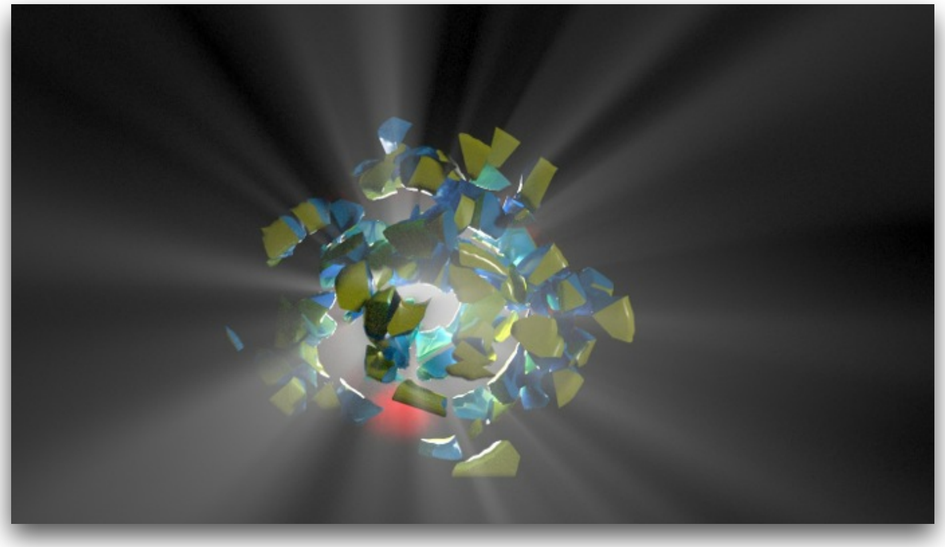
Taking a Logo and Destroying it

Having the Chunks Fly Toward Camera

Adding Particle Streams

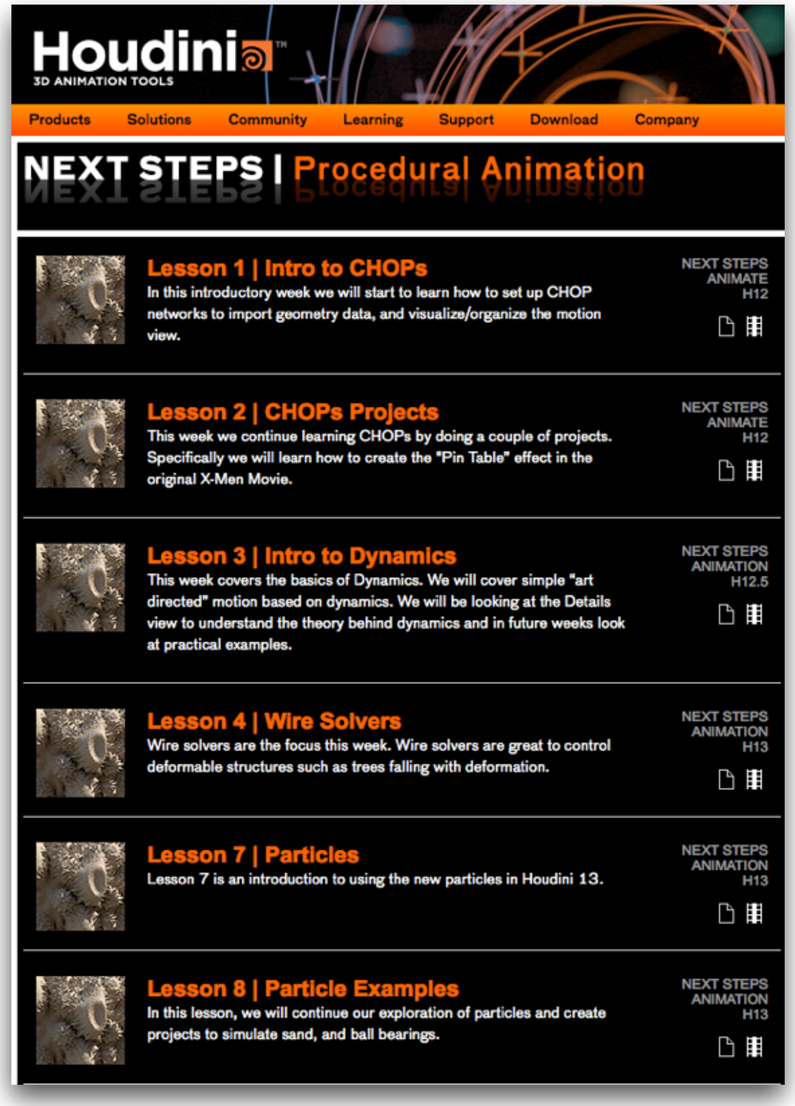
Materials

Volumetric Lighting



SIDE EFFECTS SOFTWARE

Prerequisites



Basic Requirements - Some Familiarity with H13 Particles, Wrangle Nodes, RBDs

This Workshop is not a repeat of H13 Particles I did for Next Steps: Procedural Animation M07 & M08

If you have not watched these videos please do so:

http://www.sidefx.com/index.php?option=com_content&task=blogcategory&id=228&Itemid=389

Project 1 - Logo Explode

SIDE EFFECTS SOFTWARE

Flip Book

SIDE EFFECTS SOFTWARE

Part 01 - Creating the LOGO and RBD

SIDE EFFECTS SOFTWARE

Creating the Logo

Drop down a Geometry Object -Rename it “destroy_logo”

Dive inside. Delete File SOP

Drop Down a Trace SOP

Add a picture or logo to the parameter “Image Input”

Append an Extrude SOP (not a polyextrude”

Input into the the second context of the Extrude SOP a curve that defines the bevel of the extrusion

SIDE EFFECTS SOFTWARE

Trace SOP

Trace surface node

Traces curves from an image file.

This node reads an image file and automatically traces it, generating a set of faces around areas exceeding a certain brightness threshold. You can control this threshold and the resolution of the resulting faces.

Uses / Works in Relation With

Commonly followed by an Extrude SOP to generate beveled objects from logos, scans of geographical maps, etc. It can be animated by reading in a series of images.

Parameters

Trace

Threshold	Minimum brightness level that should be traced.
Add Point Texture	Toggles the creation of point texture UV coordinates.

File Switch

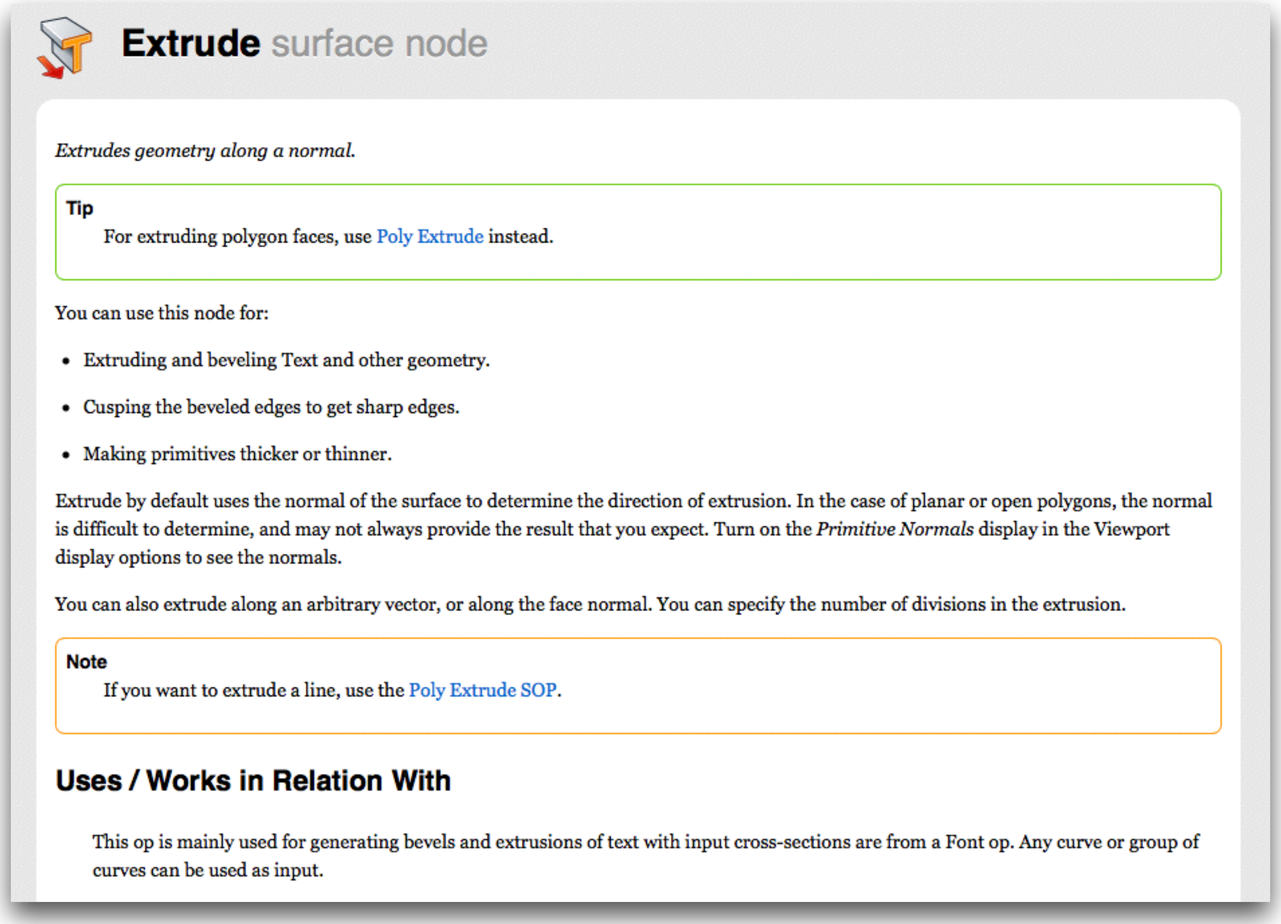
Image Input	Name of the file to use.
Scale to Size	Loads the file at the specified image size rather than its actual size.

COP Switch

COP Path	The path to the COP to use.
COP Frame	Which frame(s) to use from a cop composed of a sequence of images.
Planes	Which planes to use as the source.

SIDE EFFECTS SOFTWARE

Extrude SOP



Extrude surface node

Extrudes geometry along a normal.

Tip
For extruding polygon faces, use Poly Extrude instead.

You can use this node for:

- Extruding and beveling Text and other geometry.
- Cusping the beveled edges to get sharp edges.
- Making primitives thicker or thinner.

Extrude by default uses the normal of the surface to determine the direction of extrusion. In the case of planar or open polygons, the normal is difficult to determine, and may not always provide the result that you expect. Turn on the *Primitive Normals* display in the Viewport display options to see the normals.

You can also extrude along an arbitrary vector, or along the face normal. You can specify the number of divisions in the extrusion.

Note
If you want to extrude a line, use the Poly Extrude SOP.

Uses / Works in Relation With

This op is mainly used for generating bevels and extrusions of text with input cross-sections are from a Font op. Any curve or group of curves can be used as input.

SIDE EFFECTS SOFTWARE

Fracturing the Logo

We need to scatter “seed” points for the Voronoi Fracture to Work.

We do not just want surface factoring but we want to have a true 3D fracture. Therefore we have to scatter points in the interior of the logo too.

Converting to a Volume will allow us to accomplish this.

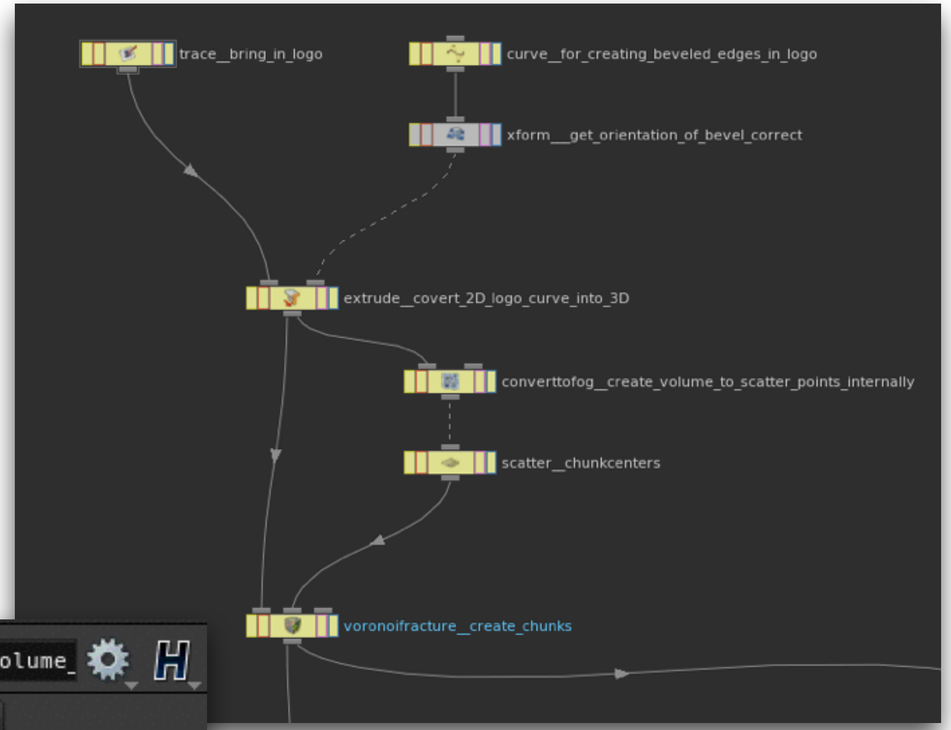
Append a isoOffset SOP to the Extrusion

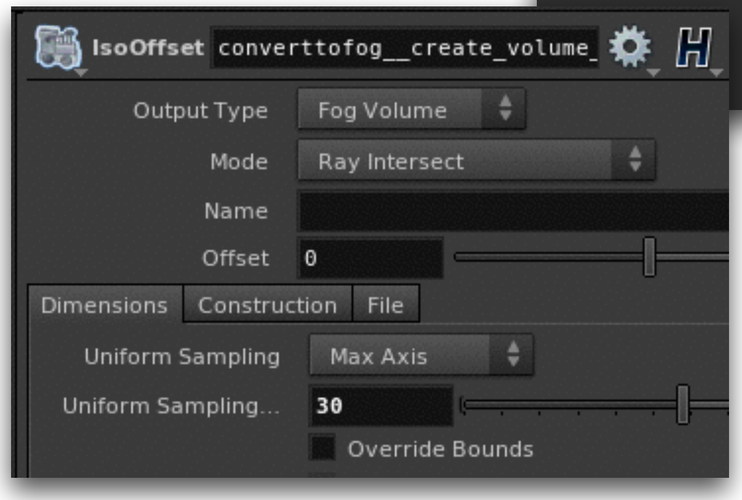
The Output Type is going to be a Fog Volume

Mode - Ray Intersect

Since we just need to break some crude pieces from the original logo we only need a low sampling rate.

Sampling rate - 30





SIDE EFFECTS SOFTWARE

Fracturing the Logo (cont.)

Now we have a 3D Volume we can “scatter” points throughout the logo and not just the surface

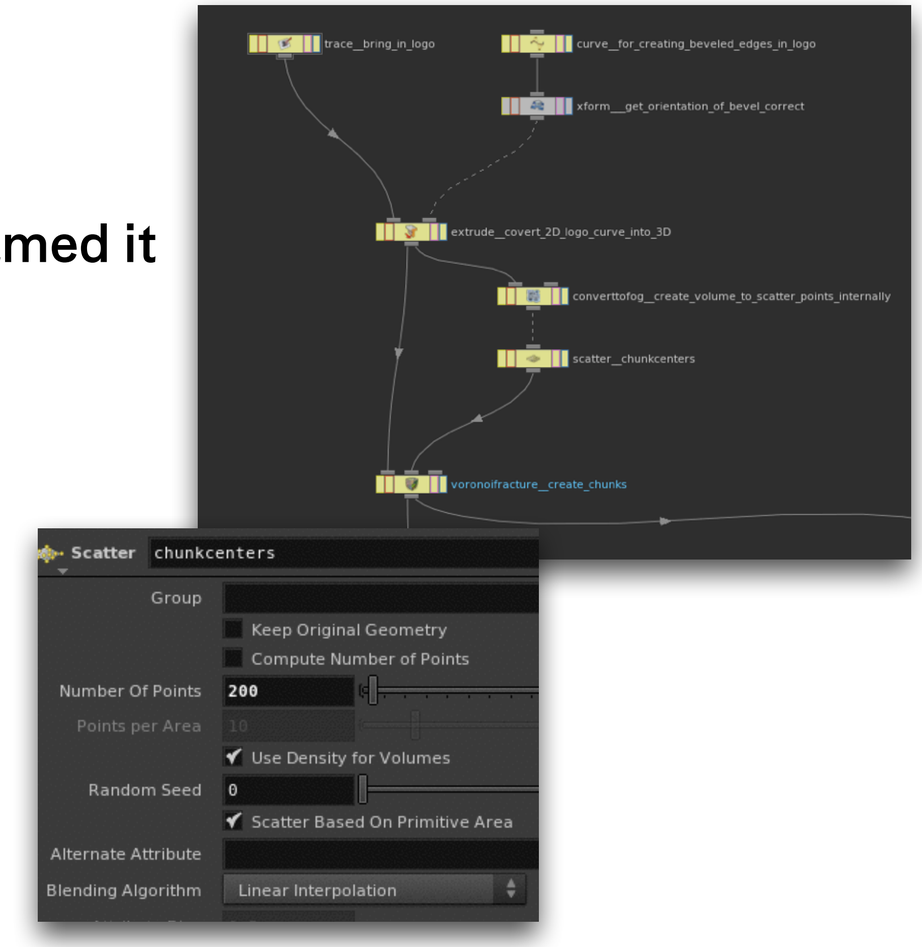
Append a Scatter SOP to the “isoOffset” SOP. I named it “chunk centers”

Scatter 200 points

Now drop down a VoronoiFracture SOP

In the first context wire the Extrude SOP

In the second context wire the Scatter SOP



SIDE EFFECTS SOFTWARE

Voronoi Fracture SOP

Voronoi Fracture surface node

Fractures the input geometry by performing a Voronoi decomposition of space around the input cell points

The Voronoi fracture SOP takes two primary inputs: the mesh to fracture, and the points around which to build each Voronoi cell. Often these points will be generated by either the Scatter SOP, or the Points From Volume SOP, but any source of points will work. For a solid fracture (where the interior surface of the object is built for each piece) it is very useful to have all the points reside within the volume, in which case one fractured piece will be generated for for each cell point. Pieces cut by this SOP can be further clustered together based on attribute values on the input points. An optional third input can provide an SDF Volume to be used in depth sampling when generating detail on the interior surfaces.

Tips

- Using points scattered in a density volume generated from the geometry using IsoOffset is a flexible way to generate cell points, as you can modulate the density in the volume before scattering to get more points within particular regions of the object. Where there is higher point density, the Fracture SOP will generate more, smaller pieces.
- Mesh complexity slows down fracturing more than anything. PolyReduce before fracturing if you can.
- When testing fractures, turning off **Create Inside Surface** will speed things up and should look the same on the surface of the object.
- If you have problems with insufficient point density in long appendages or something in the input object, try scattering points after using IsoOffset and scaling the density up in those areas.
- You can object merge the `CONSTRAINT_NETWORK SOP` from inside of this node to recover the connectivity graph of the pieces. This is useful for feeding into a glue constraint network.

SIDE EFFECTS SOFTWARE

Results So Far..

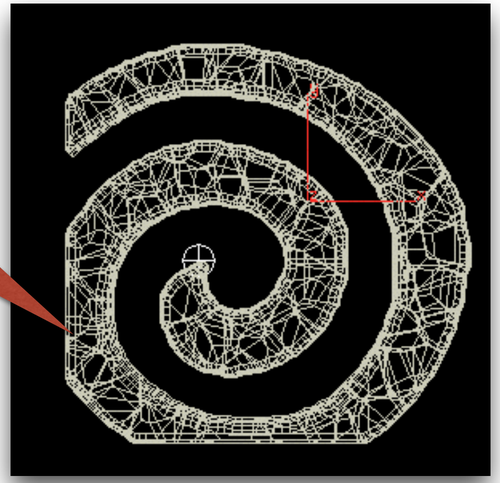
Right now we have a static object that is fractured

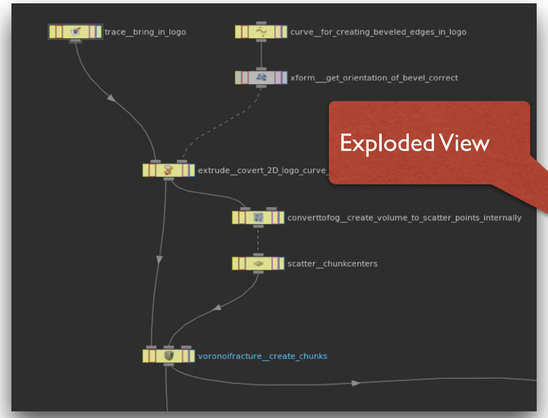
To visualize if we have inside “chunks” we can temporarily append a “Exploded View” SOP

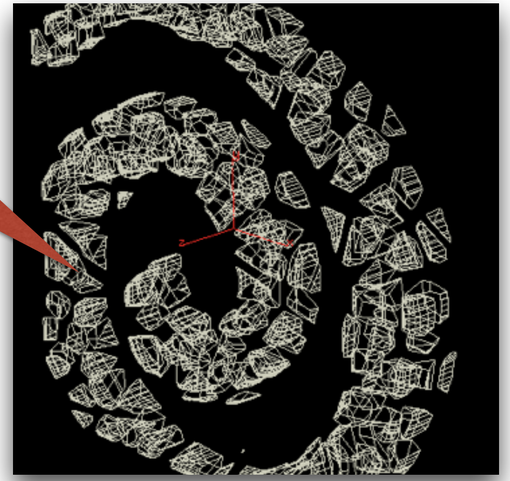
Scatter 200 points

Now drop down a VoronoiFracture SOP

- In the first context wire the Extrude SOP
- In the second context wire the Scatter SOP







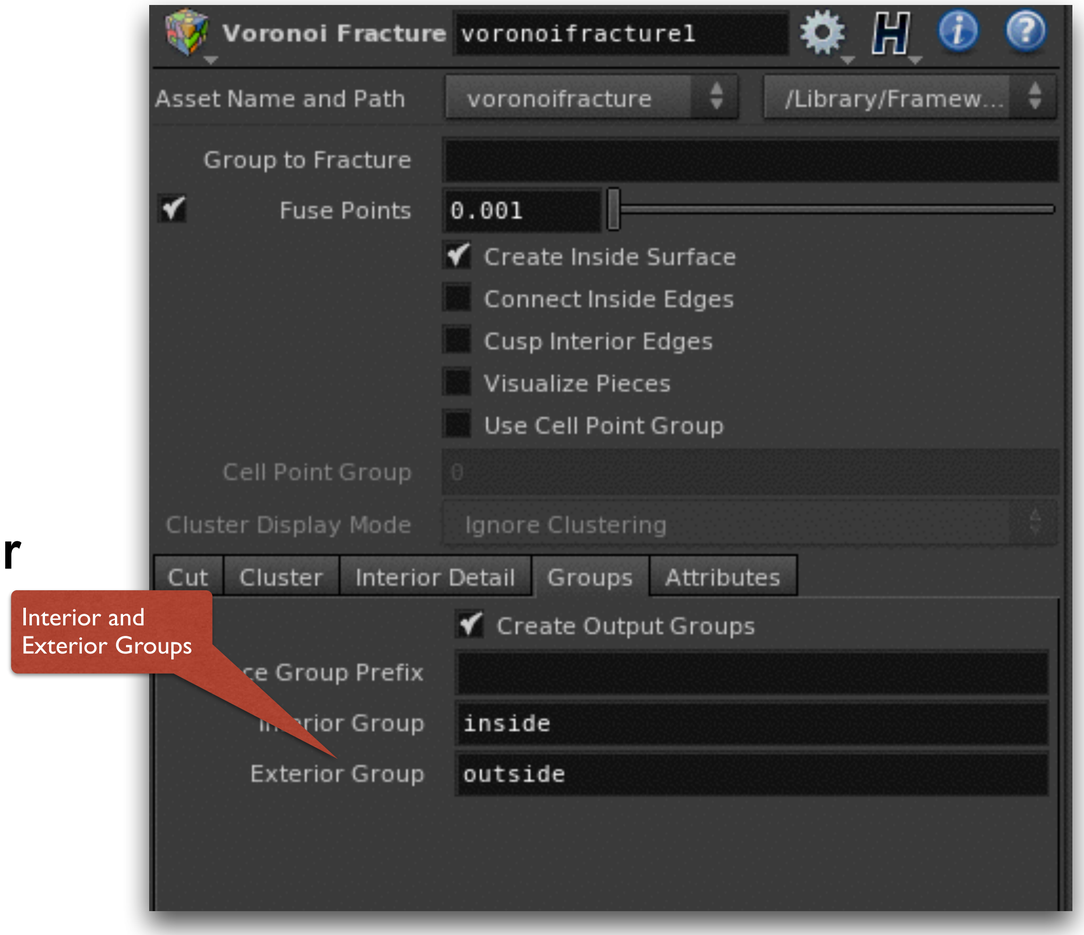
SIDE EFFECTS SOFTWARE

Voronoi Fracture , Creating Inside & Outside Groups

Go to the “Groups” Tab of the Voronoi Fracture

Enable “Create Output Groups”

Now and interior and exterior group will be created for surfacing



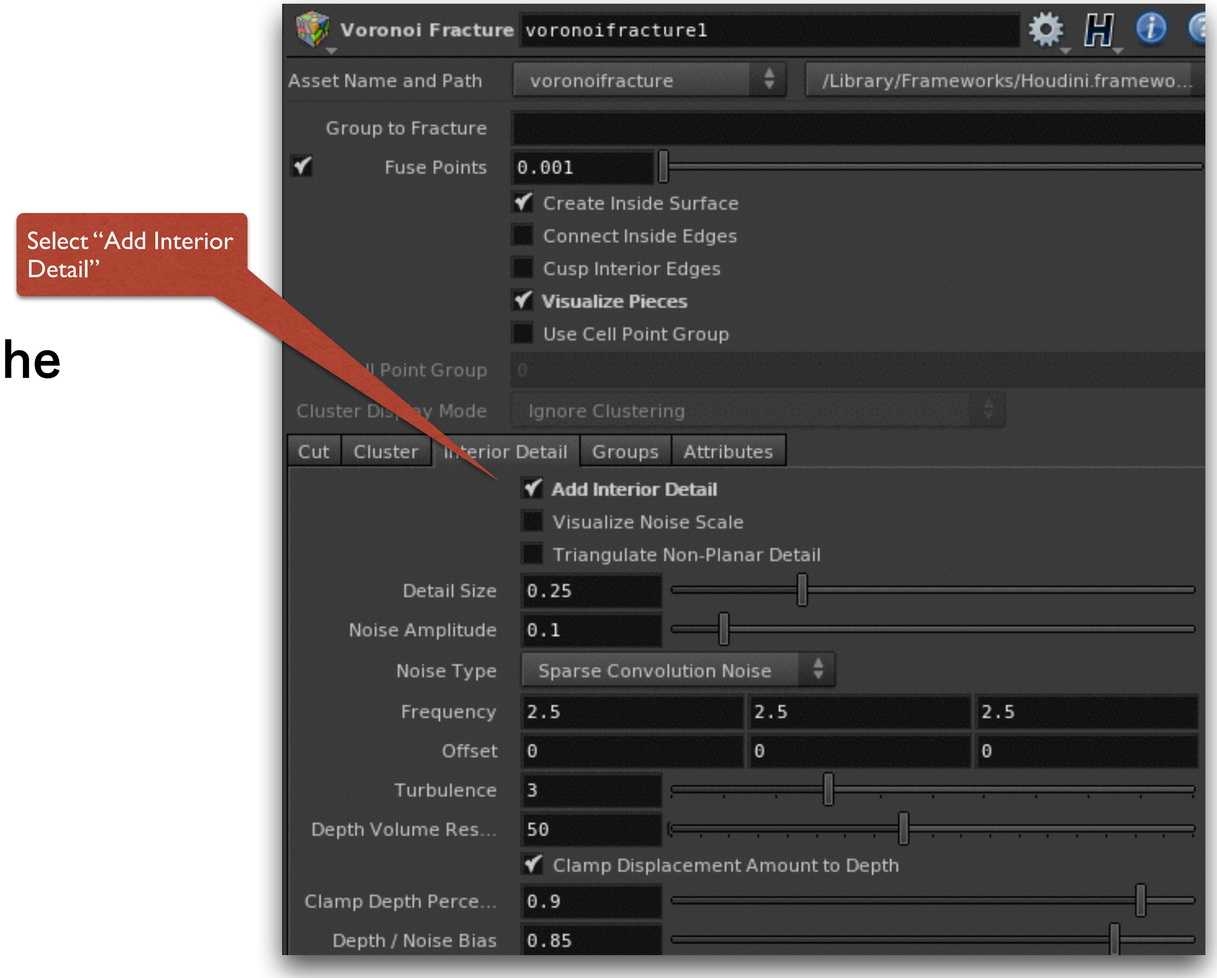
SIDE EFFECTS SOFTWARE

Voronoi Fracture , Adding Interior Details

If you want to add detail to the interior sides of the fractured geometry go to the Interior Detail Tab

Select “Add Interior Detail”

Create the roughness using Detail Size, Noise Amplitude, and the usual noise controls



SIDE EFFECTS SOFTWARE

Interior Detail

Interior Detail

Add Interior Detail	Add additional polygons to the interior surfaces created when **Create Inside Surfaces** is enabled.
Visualize Noise Scale	Add color to the pieces to indicate the amount of noise applied at each point.
Triangulate Non-Planar Detail	The **Add Interior Detail** option can create non-planar quadrilaterals. This option will triangulate the interior detail.
Detail Size	The size of the polygons added to the interior surfaces.
Noise Type	The type of noise added to the interior points.
Frequency	The frequency of the noise added to the interior points.
Offset	The offset of the noise added to the interior points.
Turbulence	The turbulence of the noise added to the interior points.
Depth Volume Resolution	Resolution of the default depth sampling SDF. The amplitude of the noise added to the interior points is scaled by the depth of the point within the surface of the original geometry, calculated by sampling an SDF of the geometry. If an SDF is not connected to the third input of the SOP, a default one will be created at this resolution.
Clamp Displacement Amount to Depth	Limit the amount of displacement done at a point by its depth within the surface. When adding high amplitude noise to the interior points, it's possible that the interior surface could interpenetrate the exterior. This option will clamp the amount of displacement at a particular point by the depth within the surface.
Clamp Depth Percentage	The percentage of the depth within the surface to which the displacement should be clamped. Because an SDF is not a perfect representation of depth, this option allows you to clamp to a percentage of the depth. Lowering this can guarantee no interpenetration, at the expense of more clamping of the interior noise.
Depth / Noise Bias	The value for the bias curve that maps depth within the surface to the amplitude of the noise applied.
Use Depth / Noise Ramp	Use a ramp control for more precise control of the depth / noise mapping.
Depth / Noise Ramp	The ramp that maps from depth within the surface to the amplitude of the noise applied.

SIDE EFFECTS SOFTWARE

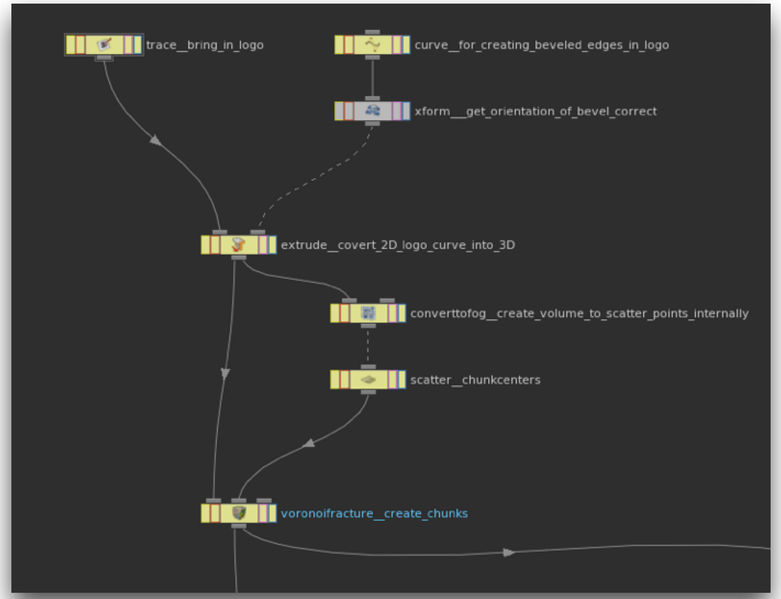
H12 & H13 Workflows Diverge

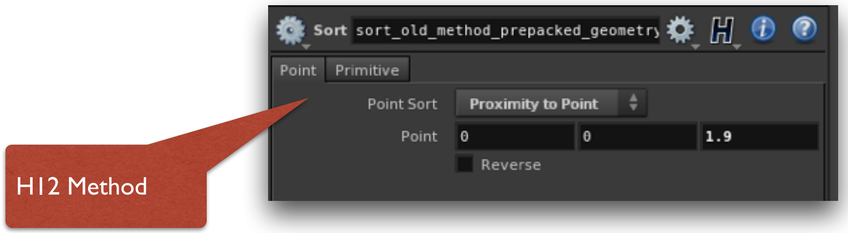
Up until now the workflows for setting up this shot in H12 and H13 have been the same.

In H12 you would now have used a “Sort SOP” to rearrange the points from closes to camera to farthest away.

The reason why you would have done this is so when we do the RBD simulation the closest “chunks” would start moving first. If the furthest away “chunks” moved first into the the non moving “chunks” the simulation would just stop.

In H13 we want to take advantage of Packed Primitives so we can not use the SORT SOP. Because Packing will do its own re-ordering





SIDE EFFECTS SOFTWARE

Creating Packed Primitives

After the Voronoi Fracture SOP append a “Assemble” SOP

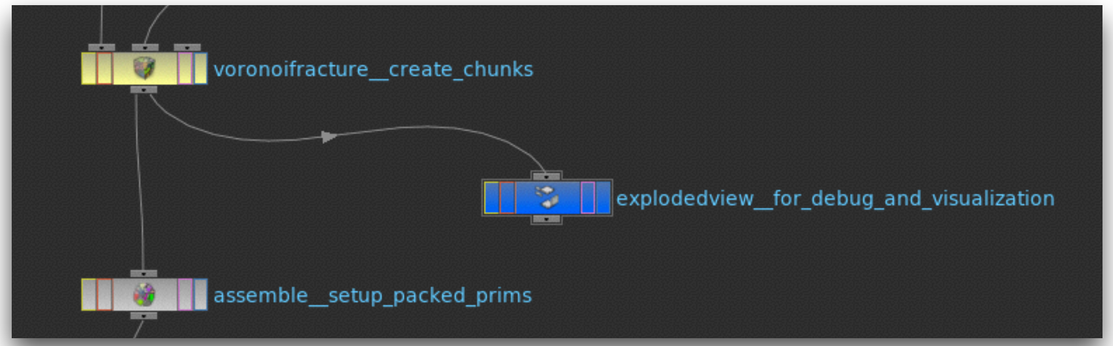
I named it “assemble__setup_packed_prims”

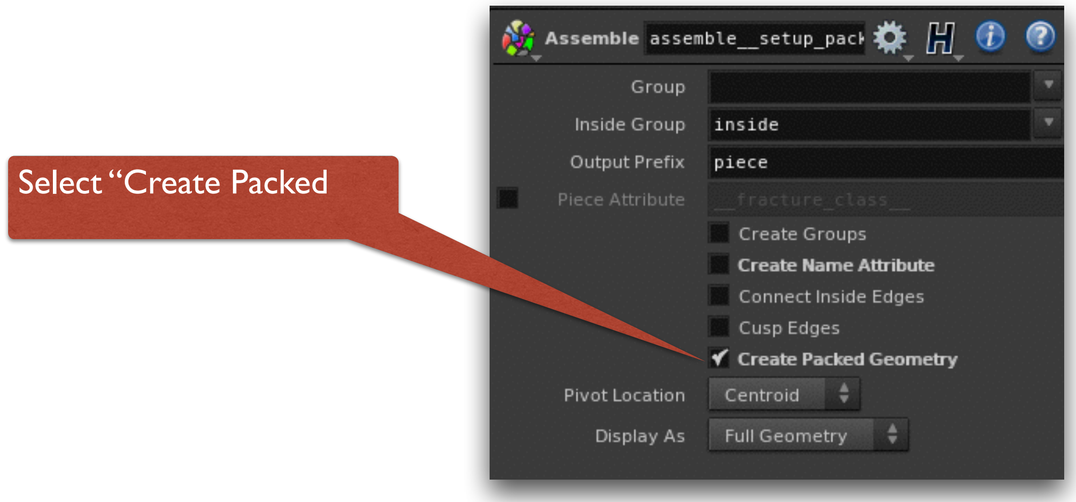
If you look at the parameters of the Assemble SOP you will notice that there is a check box for creating packed primitives

Select it

Name the Inside Group - “inside”

Name the Output Prefix - “piece”





SIDE EFFECTS SOFTWARE

Assemble SOP

Assemble surface node

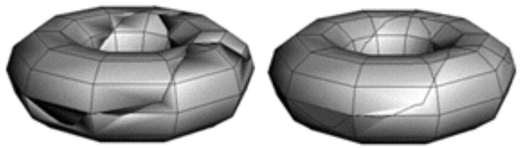
Cleans up a series of break operations and creates the resulting pieces.

This operator is used to finish the process of breaking a piece of geometry. It uses the groups and connected pieces created by the Break operator to output a set of disconnected pieces.

Note

If your broken object is made into a dynamic object, you do not need to append an assemble because one will be added automatically.

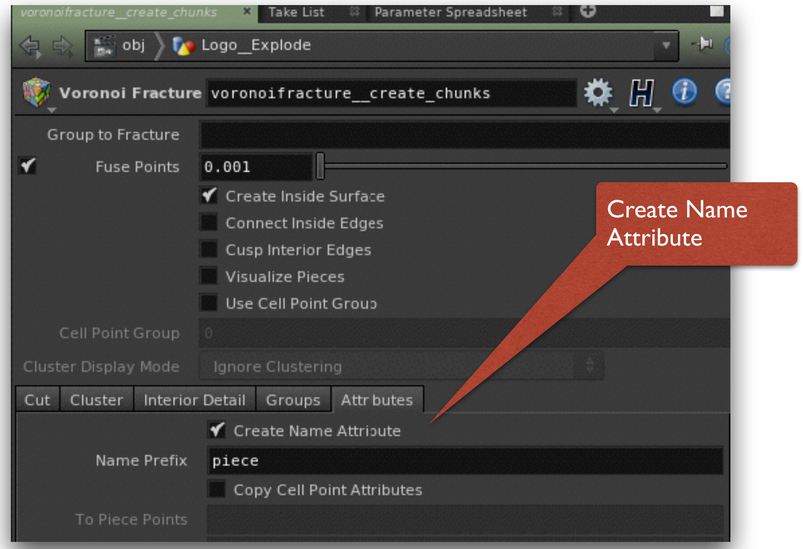
The assemble node cleans up the geometry. The following screen captures show what the geometry looks like before using the assemble tool, and after.

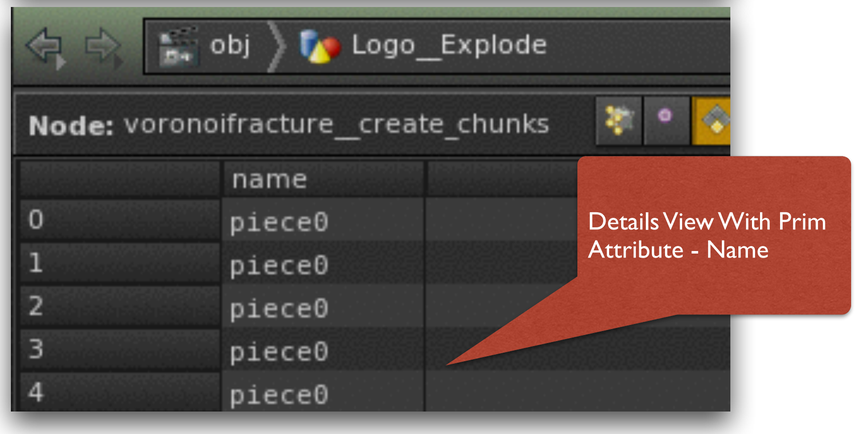


Create Packed Geometry	If the output geometry contains a primitive attribute called “name”, a packed fragment will be created for each unique value of the name attribute. Additionally, a point attribute called “name” will be created to identify the piece that packed fragment contains. Otherwise, the output geometry will be packed into an embedded packed primitive, and the **Output Prefix** will be used to create the “name” point attribute.

SIDE EFFECTS SOFTWARE

Looking at the Voronoi Fracture SOP Again





Click on the Attributes of the Voronoi Fracture SOP you just put down

Notice that by default “Create Name Attribute” is selected

A primitive attribute named “name” is created with “piece” as its prefix.

This attribute can be used in the Assemble SOP

SIDE EFFECTS SOFTWARE

Looking at the Voronoi Fracture SOP Again (cont.)

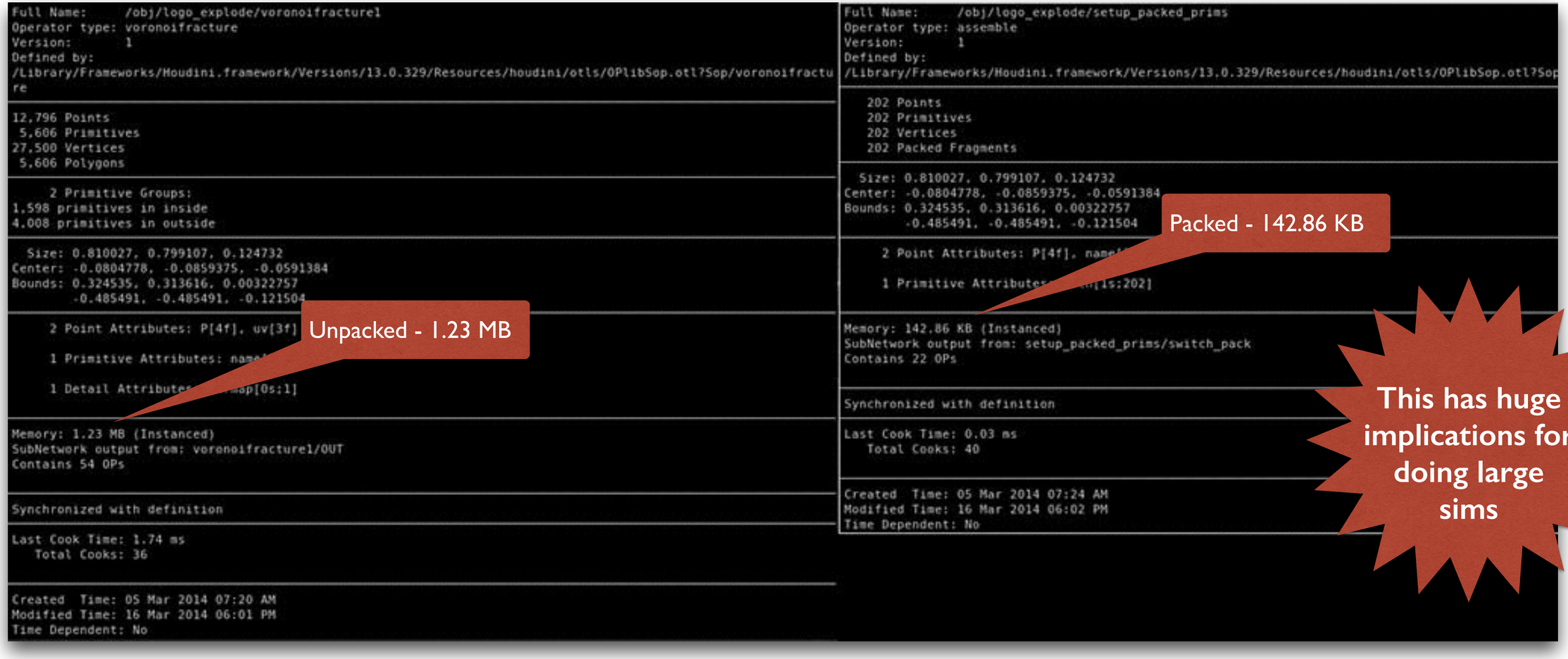
Now look at the Assemble SOP

We selected “Create Packed Geometry”

If the “name” attribute is present a packed fragment will be created for each unique value of the name attribute. Additionally, a point attribute called “name” will be created to identify the piece that packed fragment contains.

If it does not see the attribute “name” the output geometry will be packed into an embedded packed primitive, and the Output Prefix will be used to create the “name” point attribute.

Memory Footprint Packed vs Unpacked



SIDE EFFECTS SOFTWARE

Different Types of Packed Primitives

On-disk packed primitives - simply contain a file path, pointing to geometry in a paged disk file. Houdini can read subsets of the geometry as needed, and stream the geometry directly to the graphics card for display and transformation, so this is very fast and memory efficient.

In-memory packed primitives - reside in RAM. They contain a transform and a read-only reference to geometry elsewhere in memory. They can be shared and copied efficiently for instancing.

Packed fragments - are similar to in-memory packed primitives. They contain a transform and a read-only reference to a subset of another packed primitive. This lets you copy and share instances of pieces of a larger geometry efficiently.

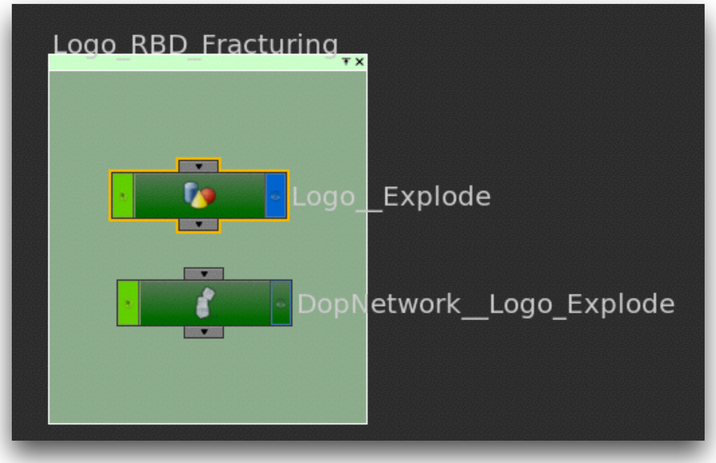
Alembic packed primitives - contain a file path pointing to an Alembic file, and a reference to an Alembic node inside the file.

Creating the RBD

The Simplest Setup

SIDE EFFECTS SOFTWARE

Using the Shelf Tool



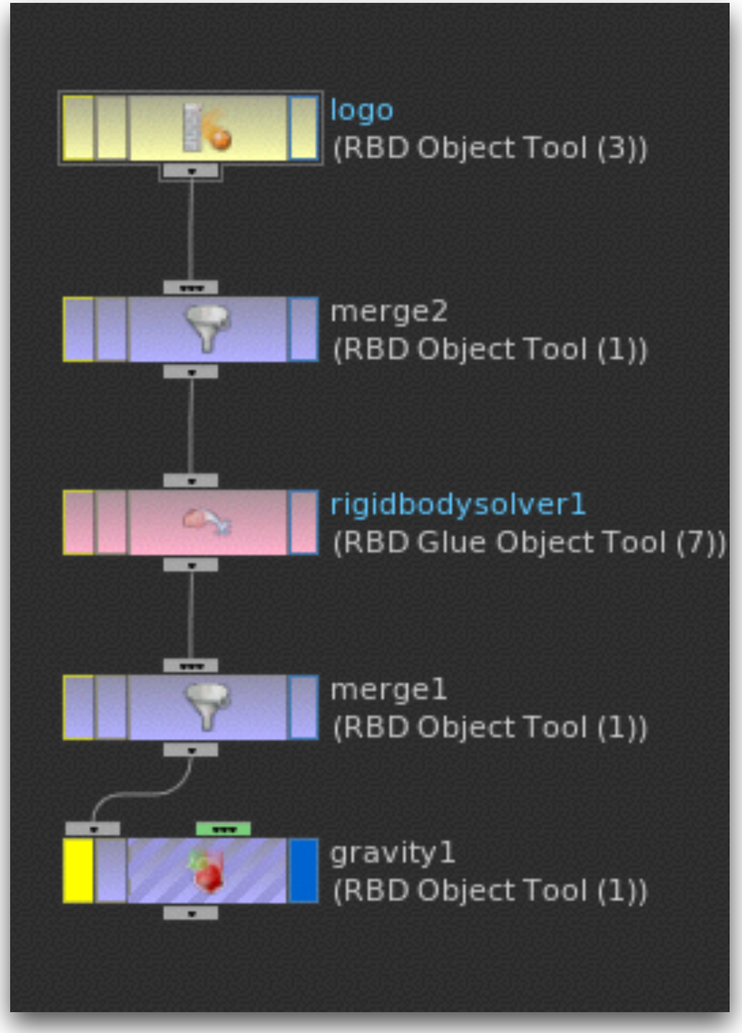
Go back up to the Object Level and select the logo_explode Object

In the Rigid Bodies Tab of the Shelf Tools Select RBD Object

A AutoDopNetwork is created for you

SIDE EFFECTS SOFTWARE

Modifying the AutoDopNetwork



We are going to advect the RBD chunks by using particle forces. This is very easy to do in H13 but difficult in H12

Before we do that for this simulation we will not need gravity so set the bypass flag on the last node of the network “gravity”

Now if we look at the network the rigid body solver is inline with two merges and the gravity

We need to add a multiple solver so we can add particle forces

Run the Simulation....

Oops!
It does not work :(

We Need a RBD Packed Geometry

SIDE EFFECTS SOFTWARE

Change RBD Object to RBD Packed Geometry

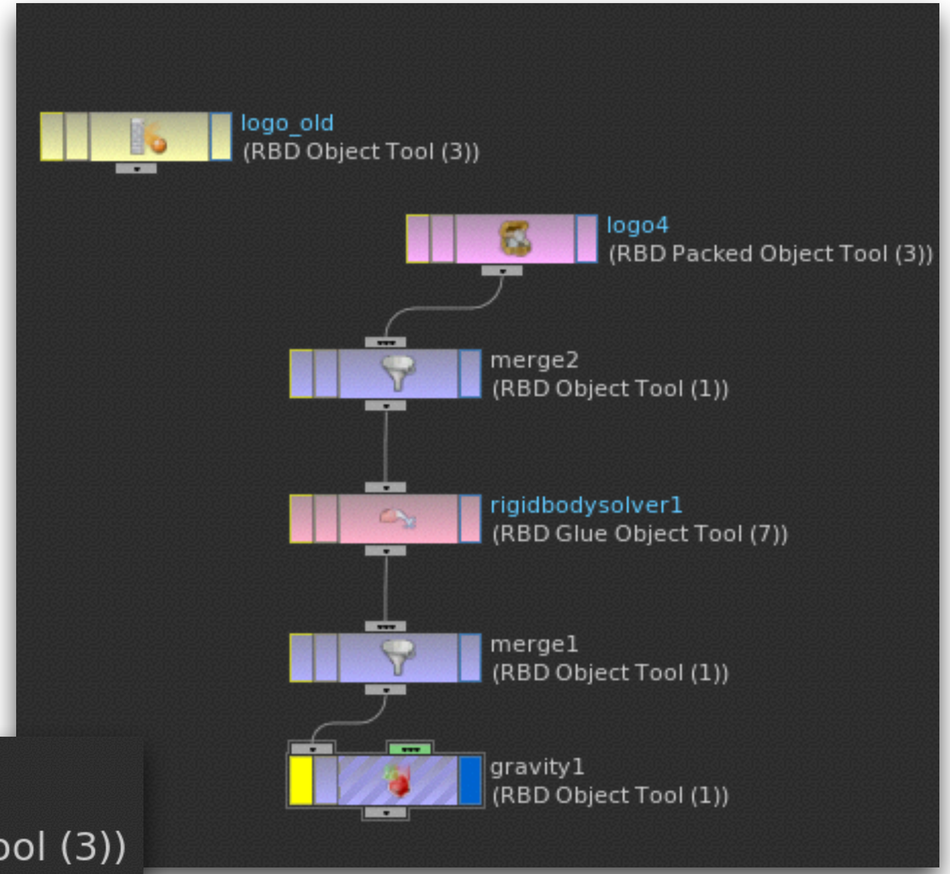
We set the “packed geometry” flag in the Assemble SOP but we made the the Object into an RBG Object

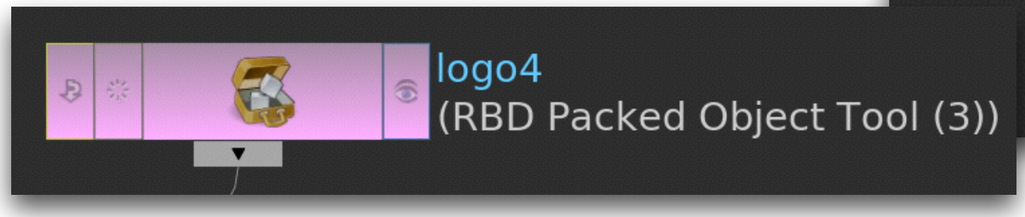
We need to swap out in the DOP Network the RBD Object to an RBD Packed Geometry Object

Drop down a RBD Object

Copy the SOPPath from the RBD object to the RBD Packed Object

Replace the RBD Object with the RBD Packed Object in the DOP Network





SIDE EFFECTS SOFTWARE

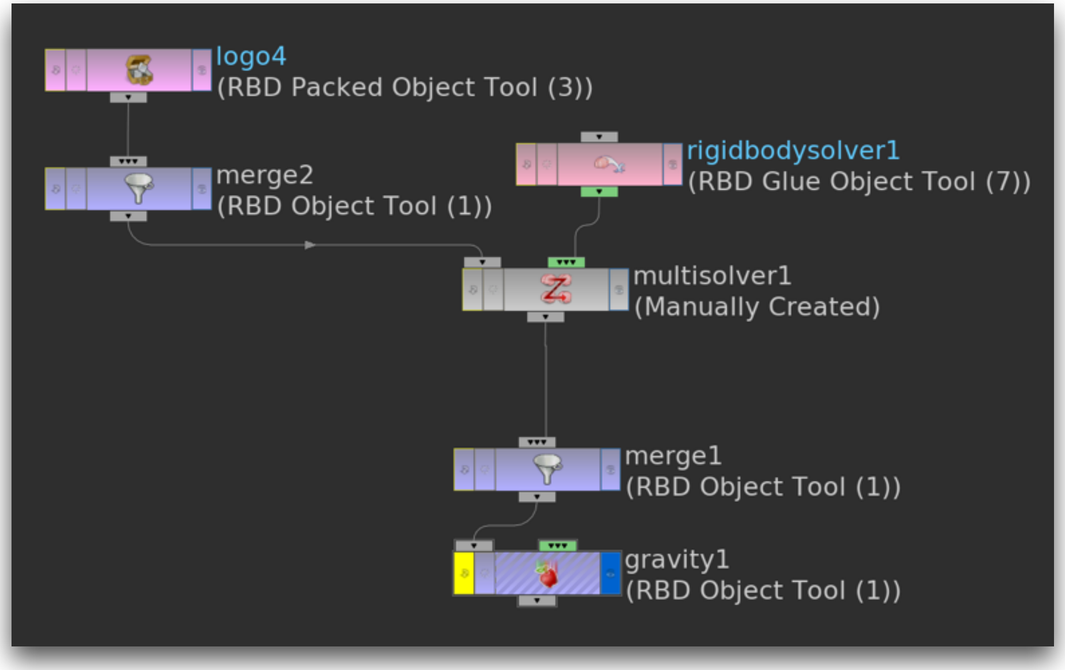
Modifying the AutoDopNetwork (cont.)

We want to advect the “chunks”

A good way to accomplish this is to use Particle Forces

We will have to re-arrange the network to use a Multiple Solver to use Particle Forces

Modifying the AutoDopNetwork (cont.)



Under the merge2 append a multiple solver

Remove the rigid body solver and rewire as show in the image on the left

If you middle click on the right context of the multiple solver you will see that you can add lots of different forces

We are going to add particle forces...

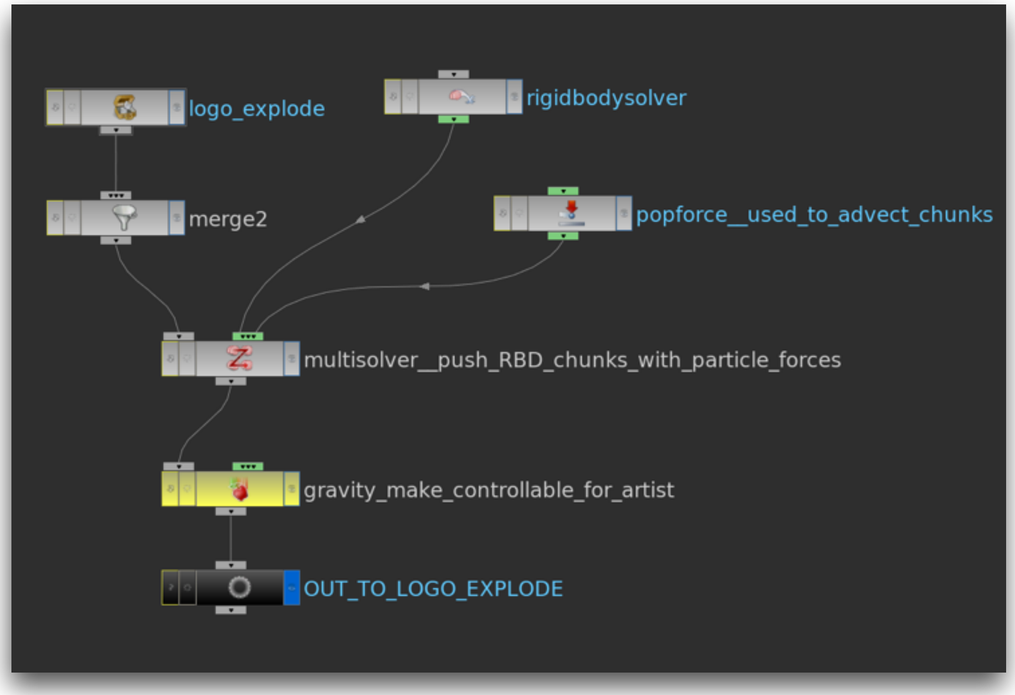
Modifying the AutoDopNetwork (cont.)

Now We are Ready to add Particle Forces

Drop down a POP Force (Caution: The is a Force DOP and a POP Force DOP. Use the POP Force)

Wire it into the right context of the Multiple Solver

I renamed some of the node to add descriptions



SIDE EFFECTS SOFTWARE

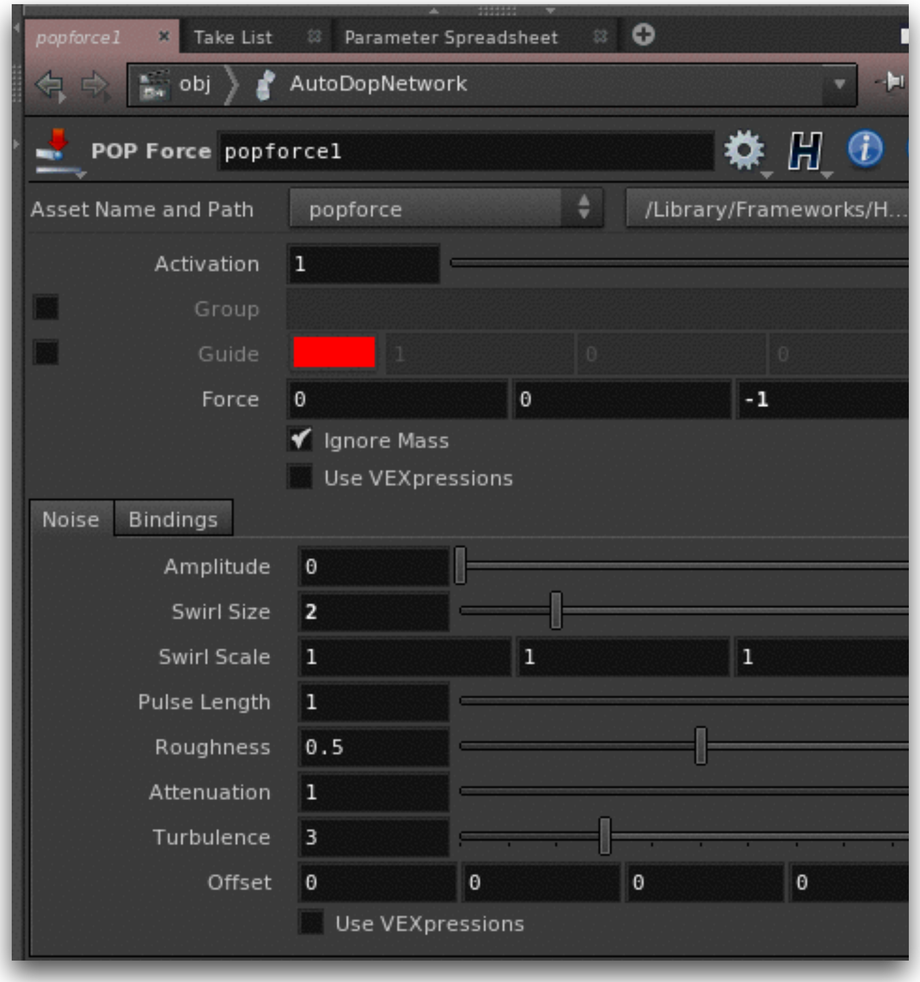
Adding Force in the Z-Axis

In the POP Force set the Force parameter to (0,0,-1)

Run the Sim....

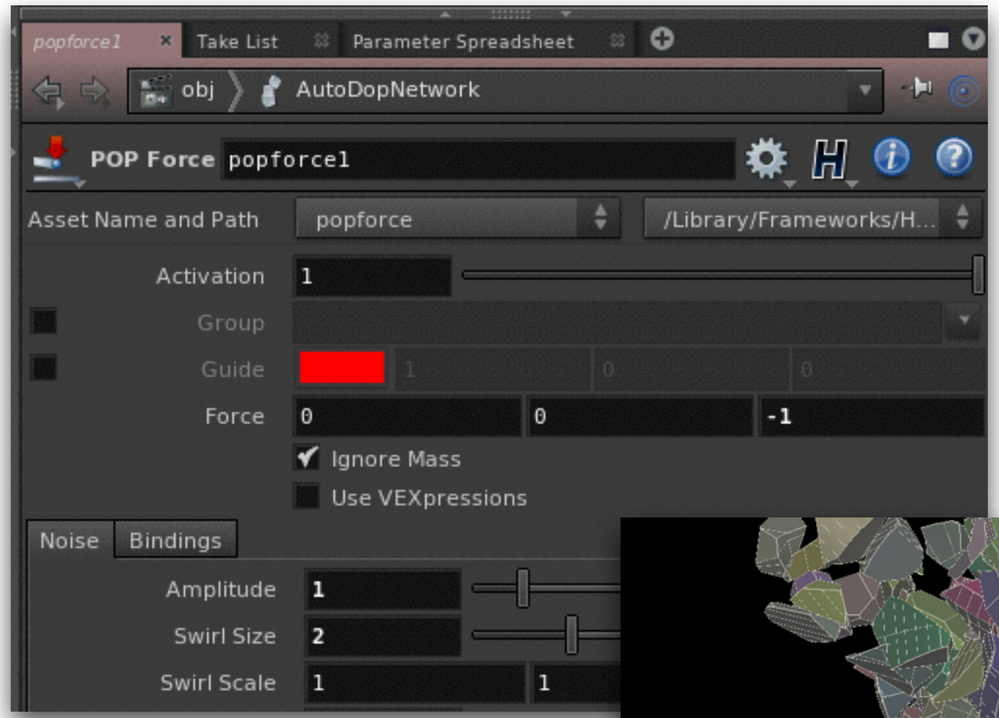
Hmm... Pretty Boring

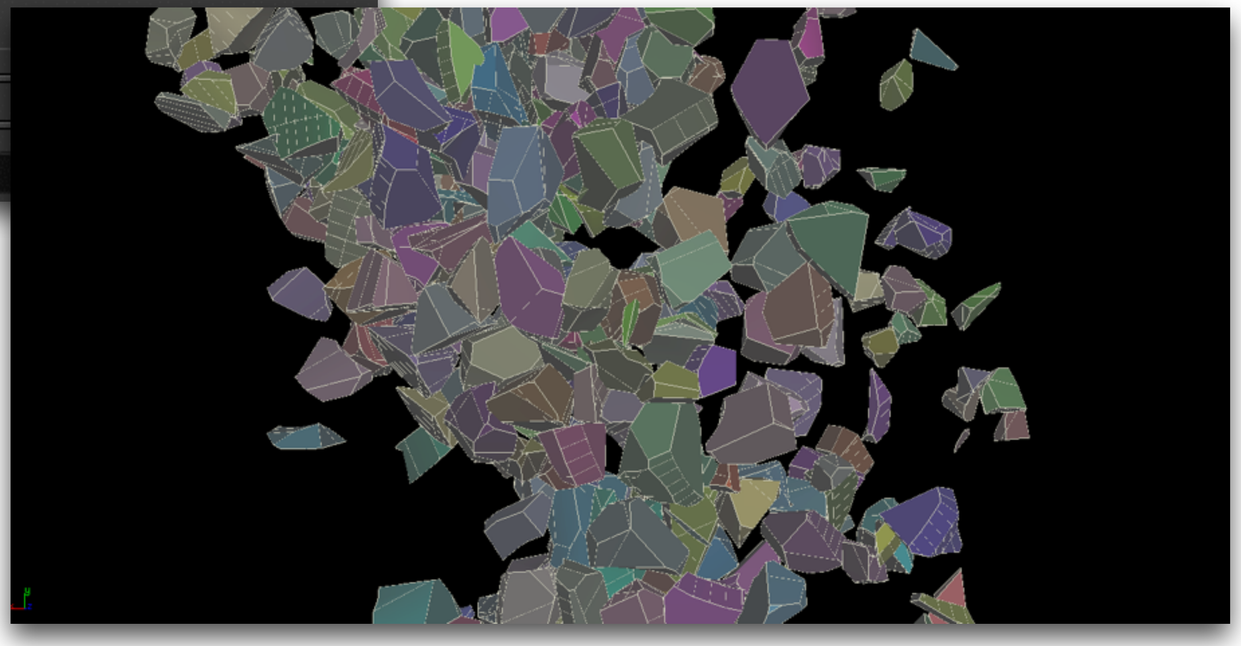
Right now every moves as one monolithic block



SIDE EFFECTS SOFTWARE

Adding Force in the Z-Axis





Let us add some curl noise

In the noise tab set Amplitude to 1

Swirl Size - 2

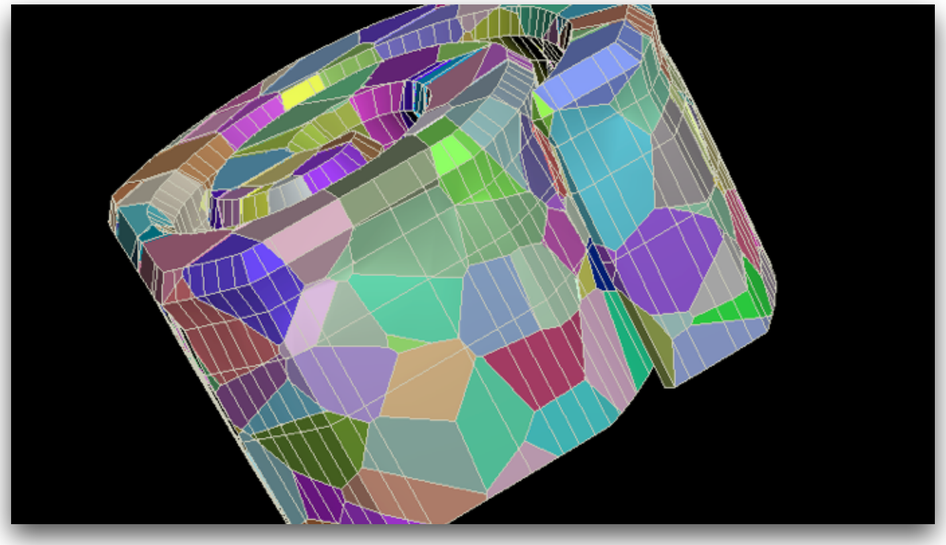
Now things are starting to Break Apart!

SIDE EFFECTS SOFTWARE

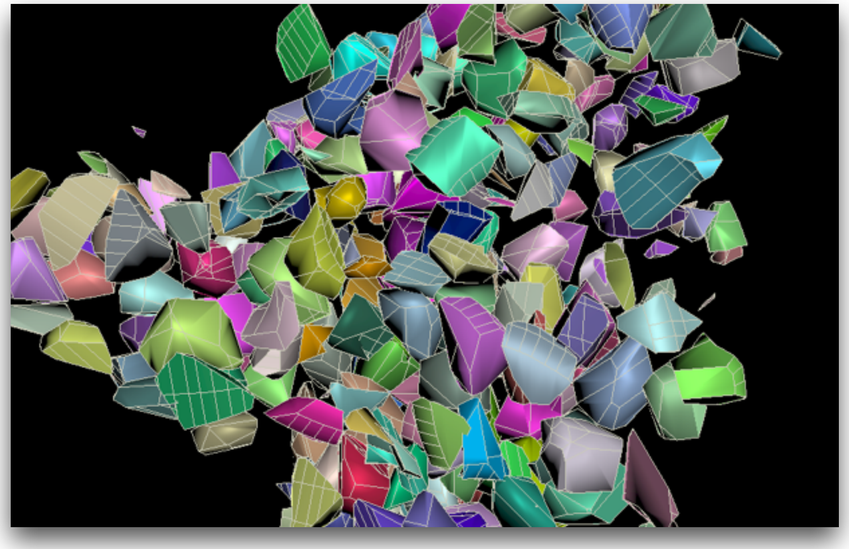
Back at the Object Level...

Go back up to the Object Level

What Happened?



Object Level



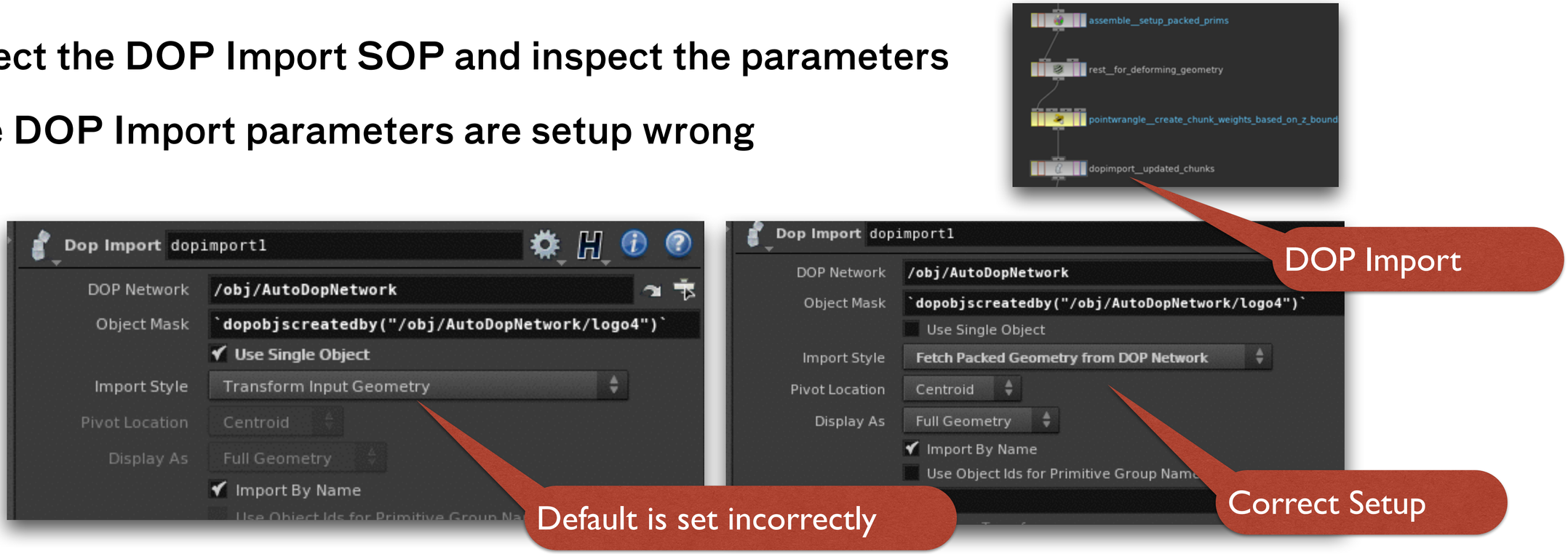
DOP Level

SIDE EFFECTS SOFTWARE

Dive into the Logo Explode Object

Select the DOP Import SOP and inspect the parameters

The DOP Import parameters are setup wrong



Set Import Style - Fetch Packed Geometry from DOP Network

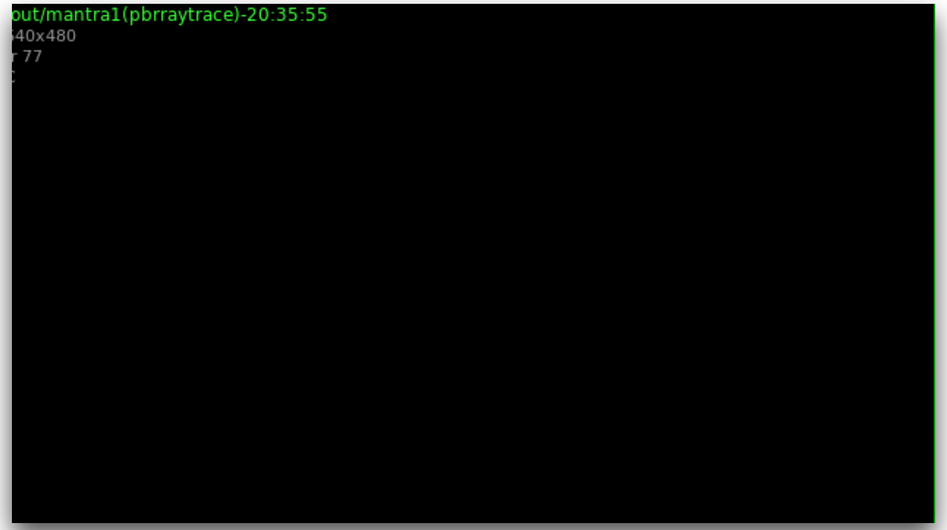
Deselect - Use SingleObject

SIDE EFFECTS SOFTWARE

Back at the Object Level...

Do a Render

Oops, We Have a Genuine Bug!



We promise to Fix it

SIDE EFFECTS SOFTWARE

The Work Around

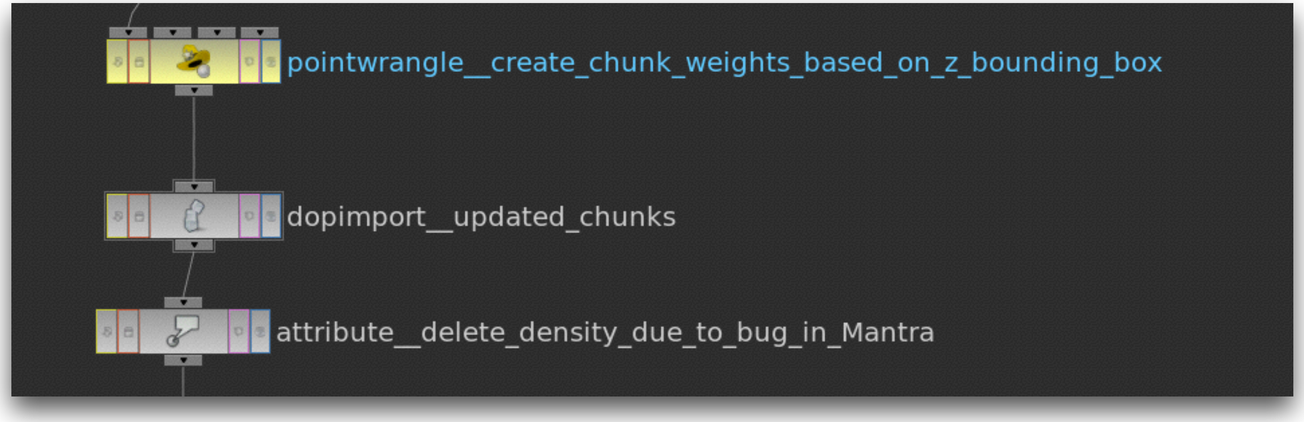
The problem is that SOHO - Part of the Mantra Renderer looks for the attribute “density.”

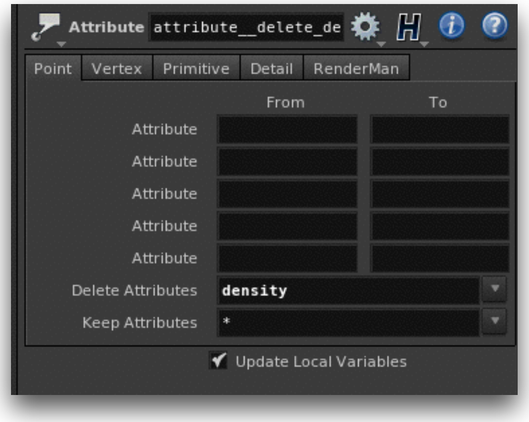
If Density is present it assumes that it needs to render a volume

Since we have no volumes present nothing is rendered

If a material is assigned without a density attribute it should render fine

The Work Around (cont.)





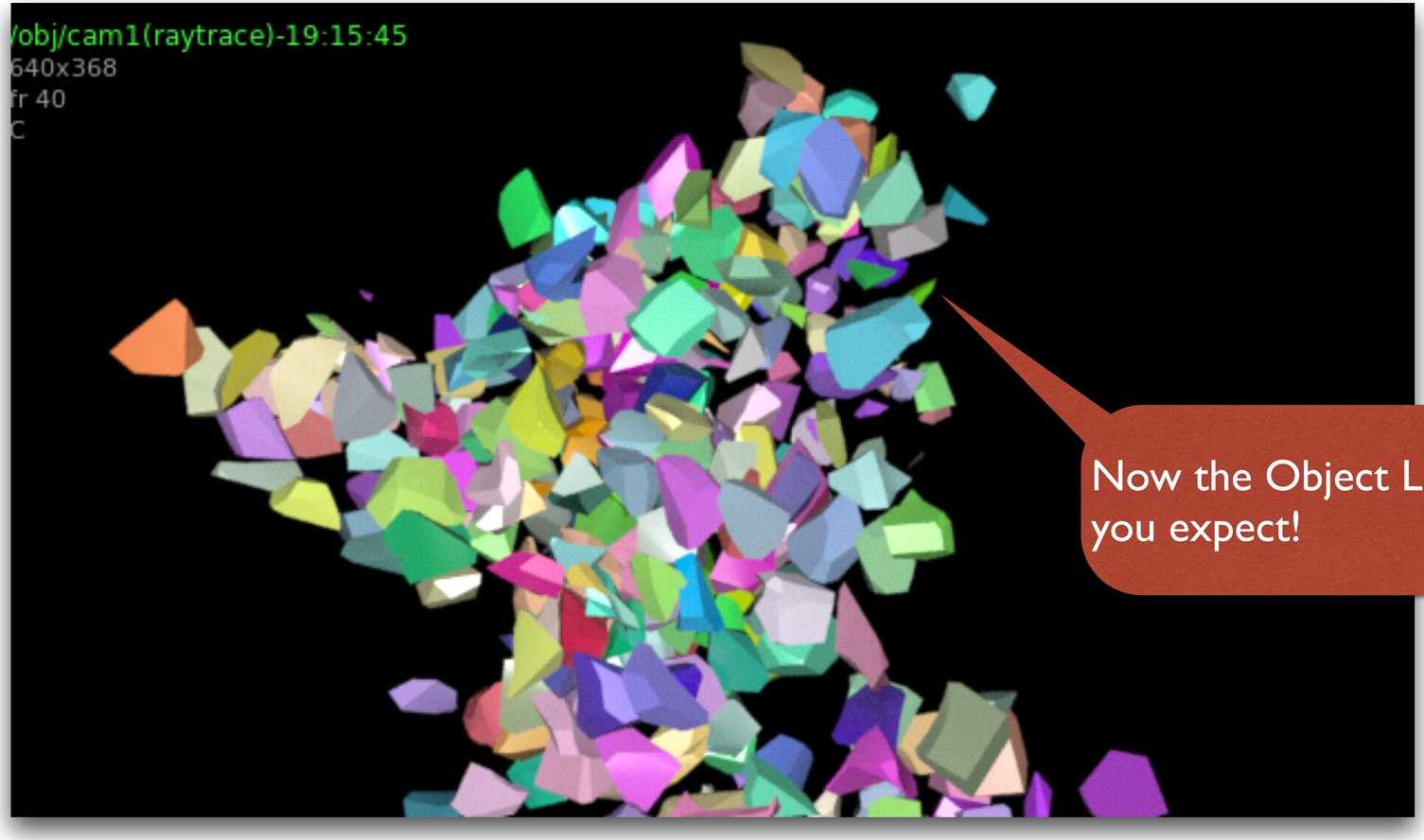
To fix this we will delete the “density” attribute

Append a Attribute SOP to the dopimport

Add “density” to the “Delete Attributes” parameter

SIDE EFFECTS SOFTWARE

Object Level Render



Now the Object Level renders what you expect!

SIDE EFFECTS SOFTWARE

Adding Character by Using Vexpressions

SIDE EFFECTS SOFTWARE

Adding Code to Start Explosion at Frame 24

Dive back into the AutoDOP Network

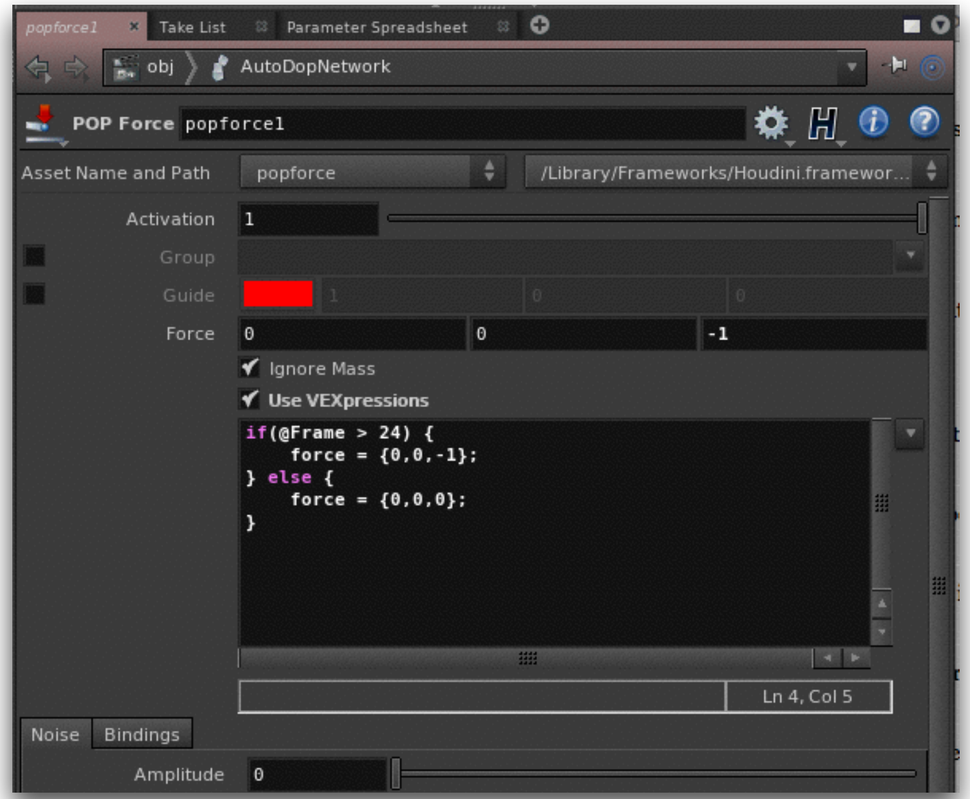
Select the POPForce DOP

Set Amplitude to 0

Select the first Vexpression Toggle

Write the following Vex Code

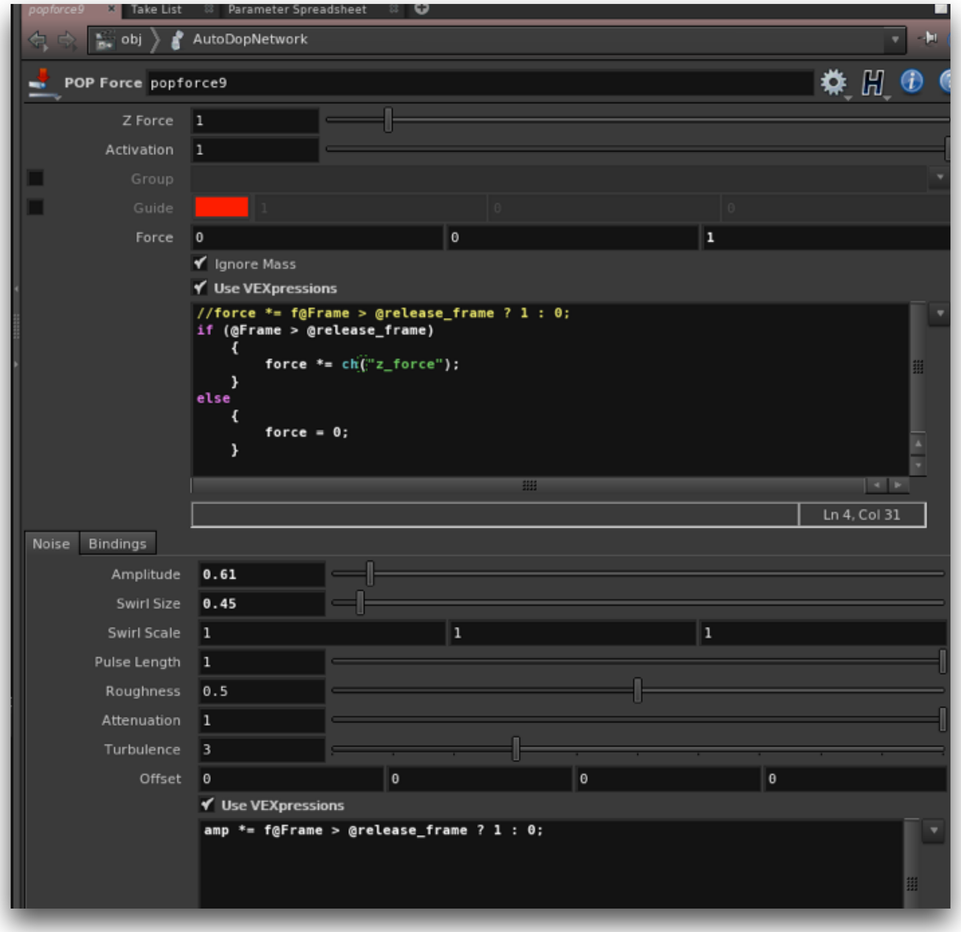
```
if(@Frame > 24) {
    force = {0,0,-1};
} else {
    force = {0,0,0};
}
```



Run the Simulation and note that the Chunks do not move until frame 24

SIDE EFFECTS SOFTWARE

POP Force Vexpressions



In the PointWrangle add a Spare parameter

Make it a float, name = z_force, label = Z Force

Default Value - 1

Range 0-10

Set the Force Parameter to (0,0,1)

SIDE EFFECTS SOFTWARE

Adding Code to Start Explosion at Frame 24

Dive back into the AutoDOP Network

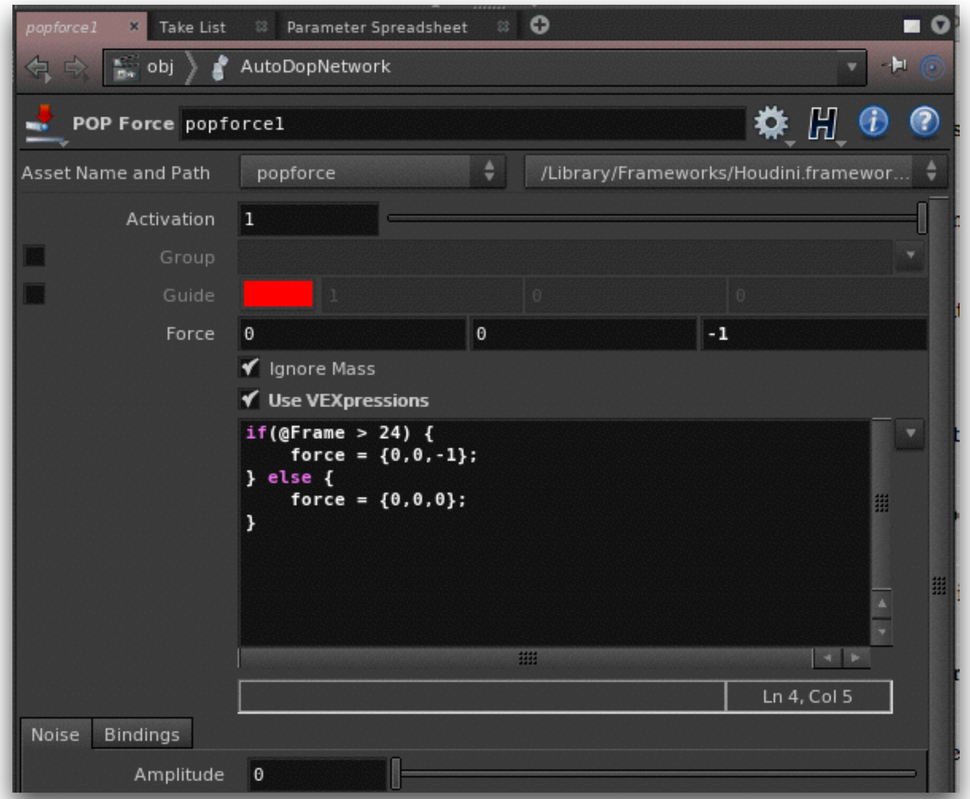
Select the POPForce DOP

Set Amplitude to 0

Select the first Vexpression Toggle

Write the following Vex Code

```
if(@Frame > 24) {
    force = {0,0,-1};
} else {
    force = {0,0,0};
}
```



Run the Simulation and note that the Chunks do not move until frame 24

SIDE EFFECTS SOFTWARE

Adding a Vexpression for the Amplitude too

Select the Vexpression right below the Amplitude Parameter

Set Amplitude to 0

Write the following Vex Code

```
if (@Frame > 24) {

    amp = 1.0;

}
```

Run the Simulation and note that the Chunks do not move until frame 24 but now when the do move we have activated the curl noise

SIDE EFFECTS SOFTWARE

Advecting the Chunks from Front to Back

SIDE EFFECTS SOFTWARE

H13 Approach to Sorting Release Frame

Append a PointWrangle SOP to the Assemble SOP

In the PointWrangle add a Spare parameter

Make it a float, name = duration, label = Duration

Default Value - 24

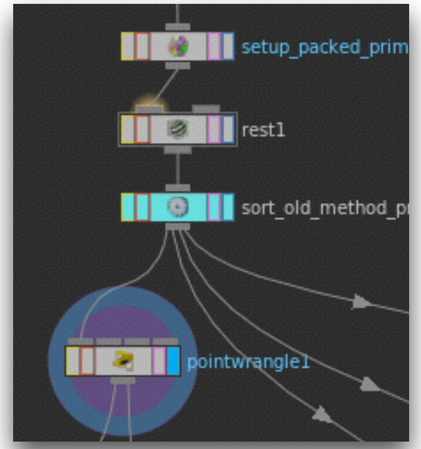
Range 0-72

Add another Spare Parameter

Make it a float, name = start_release, label = Start Release,

Default Value - 24

Range 0-72



SIDE EFFECTS SOFTWARE

H13 Approach to Sorting Release Frame (cont.)

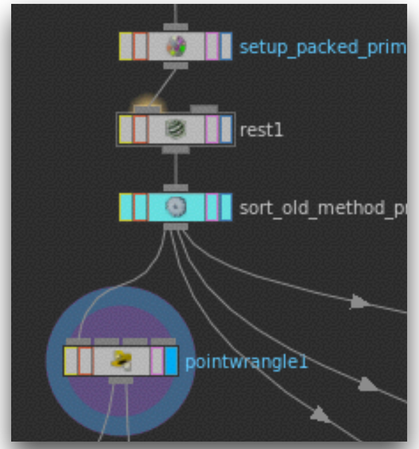
v@pos = relbbox(@P);

Here we make a new point attribute of type vector

relbbox takes a Point Position as an argument

and calculates the relative position with respect to

the bounding box



SIDE EFFECTS SOFTWARE

H13 Approach to Sorting Release Frame (cont.)

f@releaseweight = (1- @pos.z)*ch("duration");

In this expression we are calculation a weighting for chunk release time. We want chunks that are closest to camera to release first there fore we complement the z position

(1 - @pos.z) // weight with higher values chunks closer to camera

Then we multiply the weighting by the time in frames you want the chunks to release

f@releaseweight = (1- @pos.z)*ch("duration"); // release over "duration" frames

SIDE EFFECTS SOFTWARE

H13 Approach to Sorting Release Frame (cont.)

f@sortweight = (@ptnum/(@Npt-1.0))*ch(“duration");

Now we are going to incorporate the point number relative to the total number of points to calculate the sort weight. Again, we multiply it by duration.

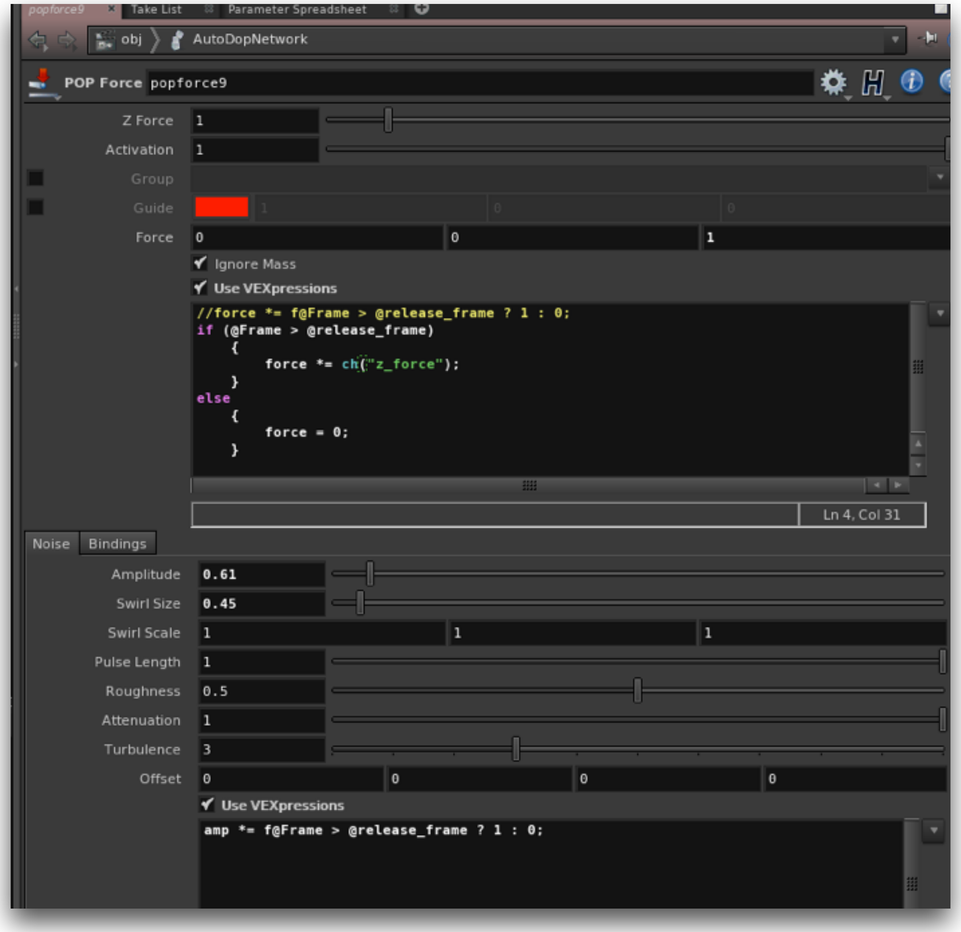
SIDE EFFECTS SOFTWARE

H13 Approach to Sorting Release Frame (cont.)

f@release_frame = ch(“start_release") + @releaseweight;

Finally we calculate each chunks release frame by adding the release weight by the duration so that no particle in back of another particle start moving before the front particle moves

POP Force Vexpressions (cont.)




```
if (@Frame > @release_frame)
   {
      force *= ch("z_force");
   }
else
   {
      force = O;
   }
```

First we are going to calculate the force of each chunk based on the attribute we created earlier “release_frame”

So now we go through each chunk and see if its attribute is > then the release frame

If it is greater I turn the force on by setting it to the “z-force”

else we turn off all forces so the chunk will not move

SIDE EFFECTS SOFTWARE

Adding Noise to the Force

Right now if you run the simulation the chunks come straight at you. Let us make it more organic by adding some curl noise.

amp *= f@Frame > @release_frame ? 1 : 0;

if the current Frame is greater then the release frame multiple amplitude by 1 if not do not allow any curl noise

SIDE EFFECTS SOFTWARE

End of Part 01 - Creating the LOGO and RBD

SIDE EFFECTS SOFTWARE

Part 02 - Creating the Emitters for the Particle Streams

SIDE EFFECTS SOFTWARE

Deciding 1 or 2 Simulations

We can do the particle trails in one of two ways

One big simulation - Have the particle trails calculated in the same DOP Network at the RBD simulation that Calculated the Chunks

Two separate simulations - Have a separate DOP Network calculate the Particle Trails

I decided on two Separate Simulations

If the simulations are large I can calculate them individually reducing the memory requirements

More flexibility in compositing

Breaking sims into separate structures easier to read and debug

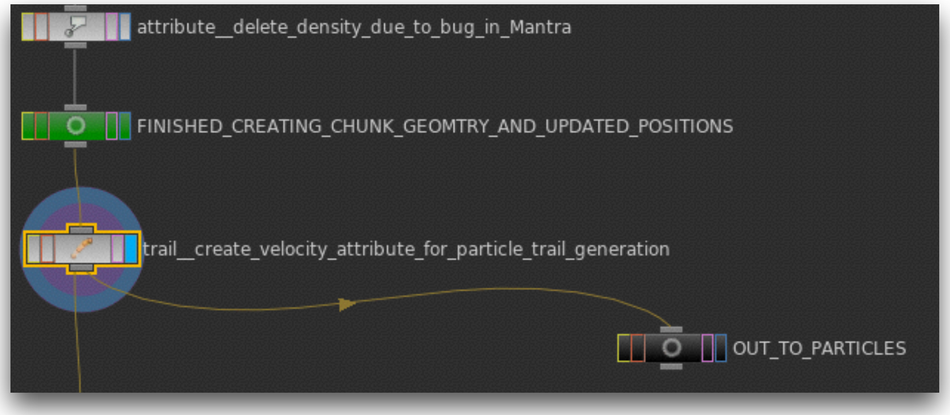
What We Want to Accomplish

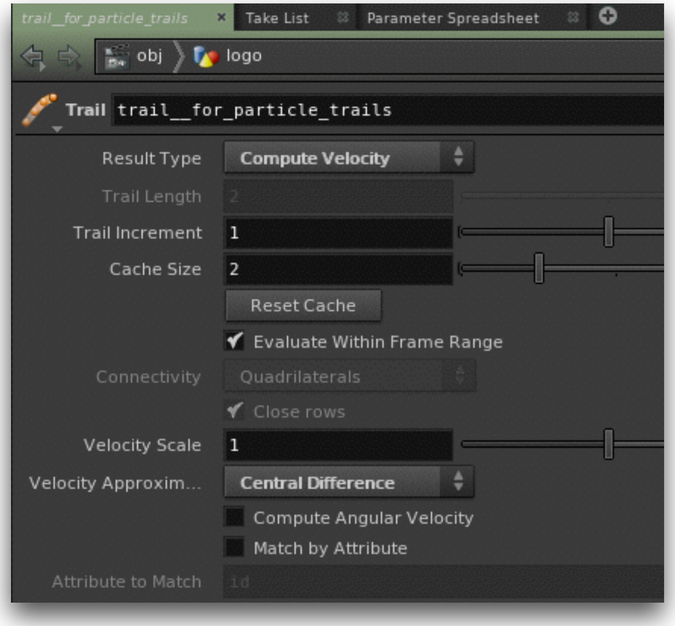
Creating Particles that emit from the surface of the chunks would be easy but not aesthetically pleasing

It would be better to isolate the back surface of Chunk and emit particles only from that surface

We also want to make sure the particles emit smoothly from the surface

Storing Velocity





We are now going to “store” the velocity of each chunk so we can create the proper particle trail

Dive into the Logo Explode Object

Append a Trail SOP to the Null or Attribute__delete_density

Append a Null - I renamed “OUT_TO_PARTICLES”

Keep the Render Flag on the Pack SOP

In the Trail SOP

- Result Type - Compute Velocity
- Velocity Approximation - Central Difference

SIDE EFFECTS SOFTWARE

Trail SOP

Trail surface node

Creates trails behind points.

The Trail op takes an input op and makes a trail of each point of the input op over the past several frames, and connects the trails in different ways. It will generate trails of any input geometry, whether it is a cube translating, a deforming surface, or particles. This is useful for multi-frame ghosting effects and temporal modeling.

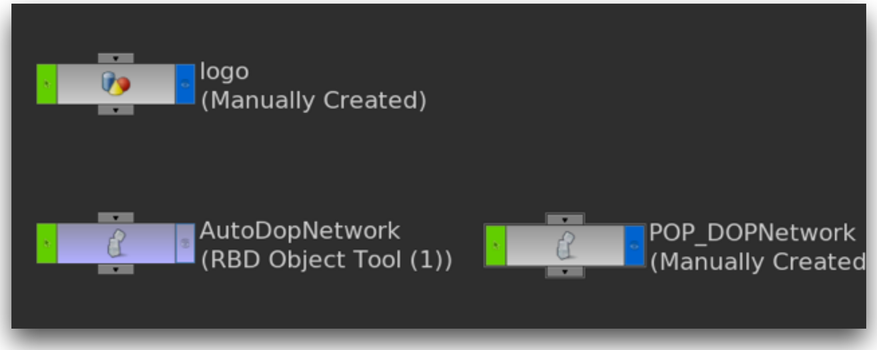
When using a Particle SOP or Spring SOP as input, it is important to keep the trail increment to integer values. Otherwise, the trail will not work well.

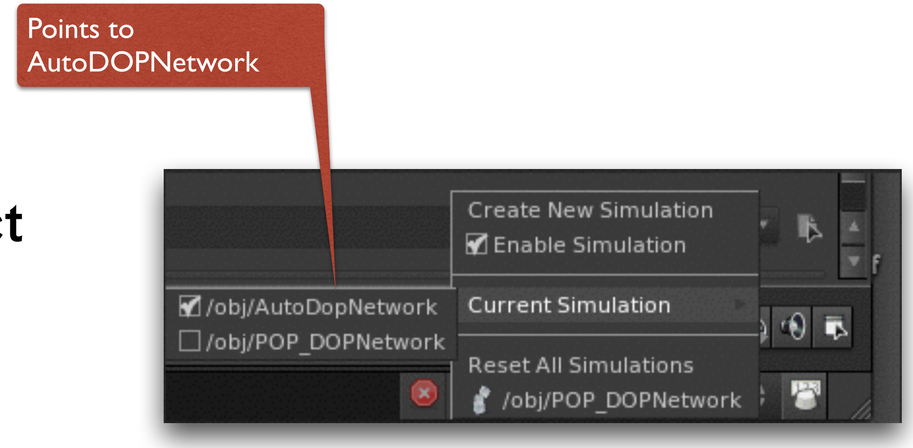
Velocity Approximation	This is the method that is used to approximate the velocity from the point positions. You can choose between **Backward Difference**, **Central Difference** and **Forward Difference**. Generally, **Central Difference** gives more accurate results.

Why I chose Central Difference

SIDE EFFECTS SOFTWARE

Creating a Second DOP Network





At the Object Level Drop Down a DopNetwork Object

Rename it something like POP_DOPNetwork

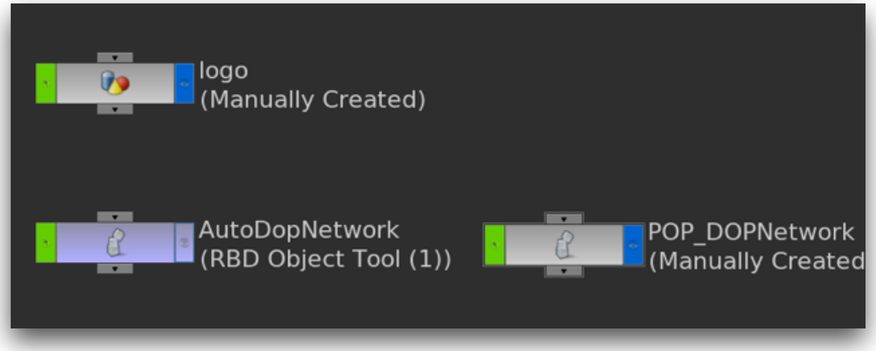
Click on the Brain in the bottom right of the Screen

Notice the current simulation is still pointing to the AutoDOPNetwork

We want to change the active Network to the POP_DOPNetwork

SIDE EFFECTS SOFTWARE

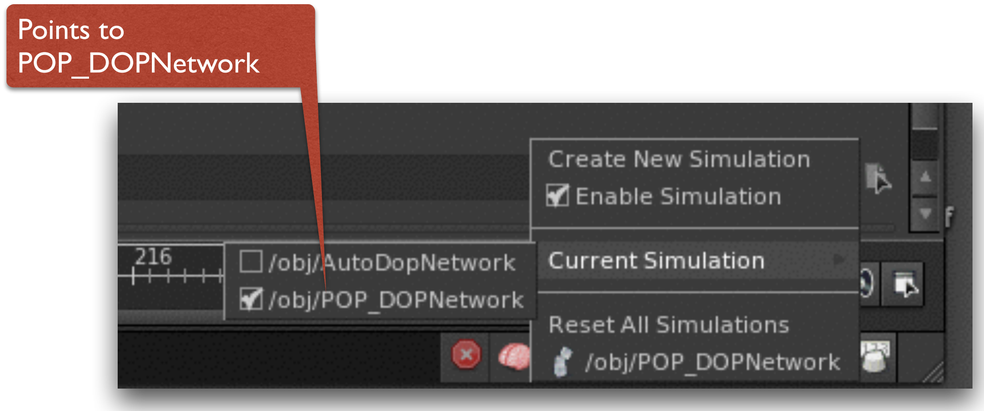
Making POP_DOPNetwork



Dive inside the POP_DOPNetwork

Now pop back up out of the POP_DOPNetwork

Click on the Brain in the bottom right of the Screen



Notice it is now pointing to the POP_DOP Network as the current simulation

SIDE EFFECTS SOFTWARE

Creating the Geometry Based Emitters for the Particle Stream

SIDE EFFECTS SOFTWARE

Geometry Based Emitter

At the Object level drop down a Geometry SOP

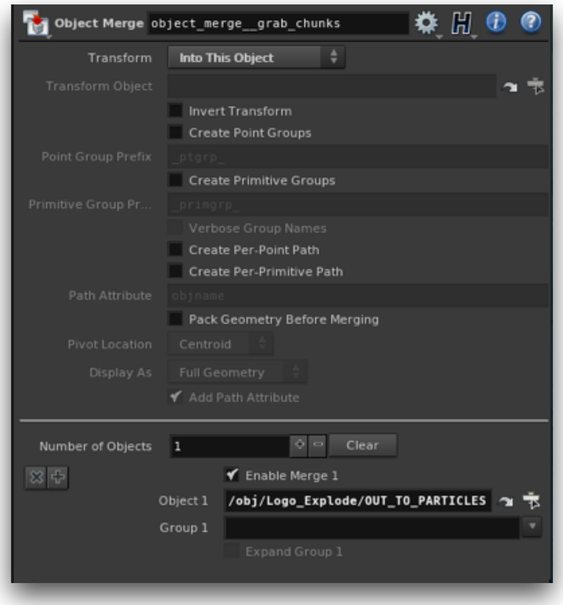
Rename it to
“Particle_Stream_Generated_From_Logo_Explode”

Dive inside

Drop down a Object Merge SOP

Object - /obj/Logo_Explode/OUT_TO_PARTICLES

Transform - Into this Object



SIDE EFFECTS SOFTWARE

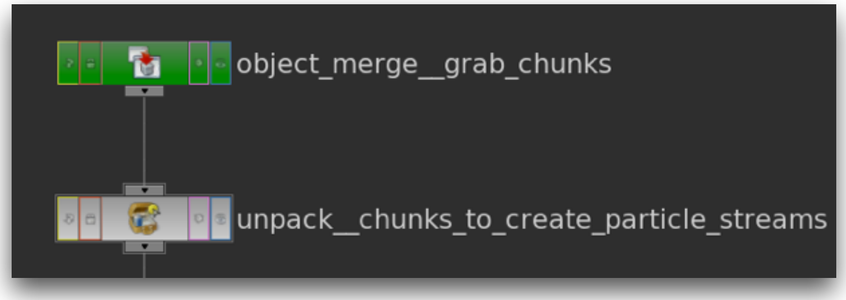
Unpacking Geometry

We need to unpack the geometry because we need to place emitters on each fragment.

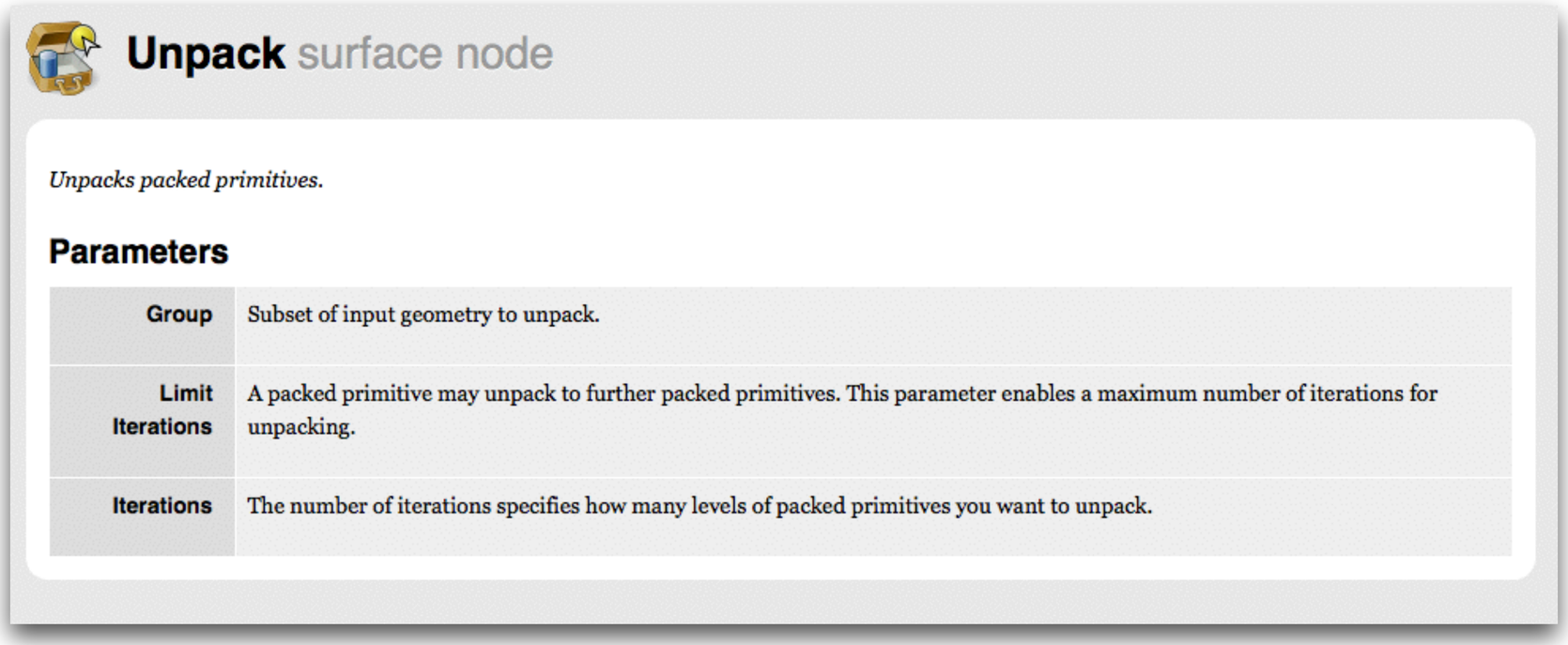
Imagine that the packed “chunks” are all contained in a ziplock bag. It is easy to cary around all the chunks in the bag. That is why it is so efficient. We are not carrying every single individual piece.

Now imagine if you want to paint each individual piece it would be impossible to paint while they are in the bag

Unpacking allows you to open the ziplock bag and select each piece for painting



Unpack SOP



Unpack surface node

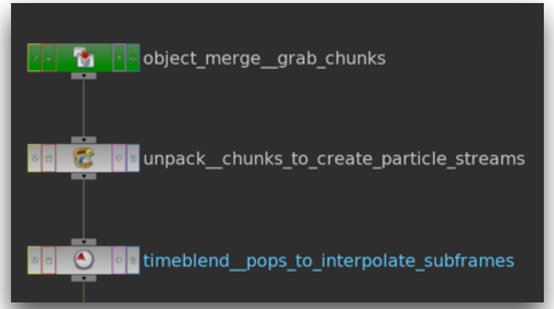
Unpacks packed primitives.

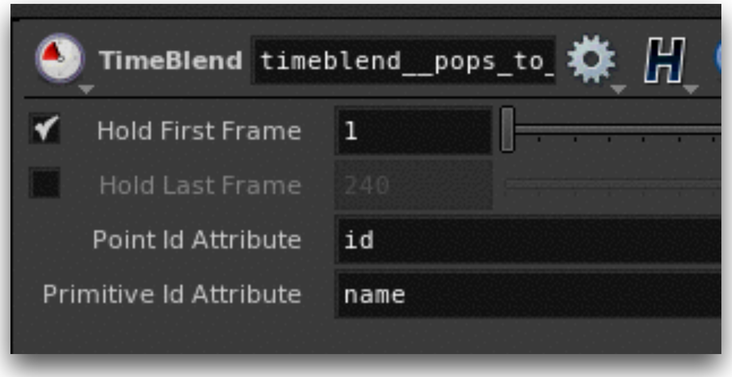
Parameters

Group	Subset of input geometry to unpack.
Limit Iterations	A packed primitive may unpack to further packed primitives. This parameter enables a maximum number of iterations for unpacking.
Iterations	The number of iterations specifies how many levels of packed primitives you want to unpack.

SIDE EFFECTS SOFTWARE

Making the Motion Smooth





Currently the Motion is calculated on the start of each frame

For the emitters we want to calculate subframes for smoother animation

In DOPs we would set are subsamples to a higher value then 1

Here, in SOPs we can use the Time Blend SOP

Append a Timeblend SOP to the Unpack SOP

- Hold Frame - 1
- Point id Attribute - id
- Primitive id Attribute - name

SIDE EFFECTS SOFTWARE

Time Blend

TimeBlend surface node

Blends intraframe values for geometry.

The Time Blend SOP will cook its input at both the start and end of the closest integer frame. The values are then blended according to the fractional position within the frame.

Optionally, one can also specify that before a start or after an end frame, the first or last geometry should be held.

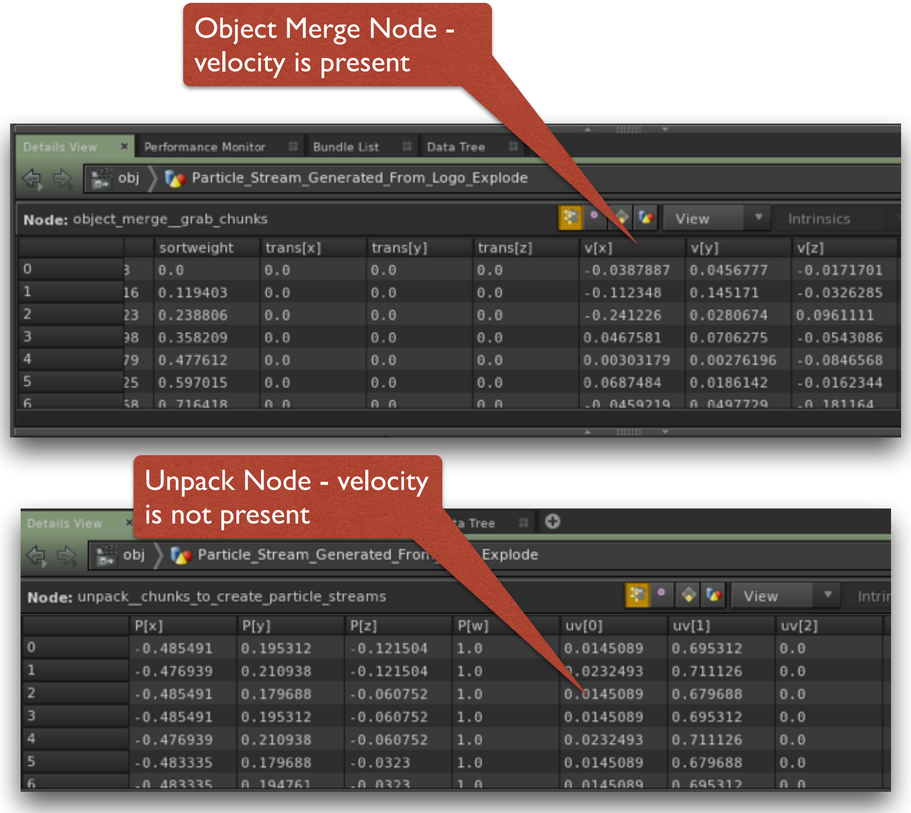
The Time Blend SOP can be used to change networks that only exist on whole frame steps into a network with proper intermediate results. This will only work if the topology of the geometry stays the same from frame to frame. It then allows for effects like multi-segment motion blur to correctly infer in between values. One such use is to turn a file sop that reads `foo$F.bgeo` into one that is defined smoothly over time.

Main Concept - Change Networks from whole frames to proper intermediate results

Parameters

Hold First Frame	Determines if the first frame should be clamped. If so, any evaluations before this frame value will instead evaluate at this frame value.
Hold Last Frame	Determines if the last frame should be clamped. If so, any evaluations after this frame value will instead evaluate at this frame value.
Point Id Attribute	If this attribute exists, it will be used to find corresponding points in the different blend shapes. For example, using id will allow particle systems to find matching points even if particles are deleted. If it does not exist or is blank, point numbers will be used for the matching.
Primitive Id Attribute	If this attribute exists, it will be used to find corresponding primitives in the different blend shapes. For example, using name will allow fluid systems to find matching volumes even if volumes are reordered. If it does not exist or is blank, primitive numbers will be used for the matching.

Hmm, Need to Compute Velocity Again...



I thought I computed velocity using the Trail SOP in the Logo_Explode Object

If I look at the Details View of the Object Merge velocity is there as a point attribute

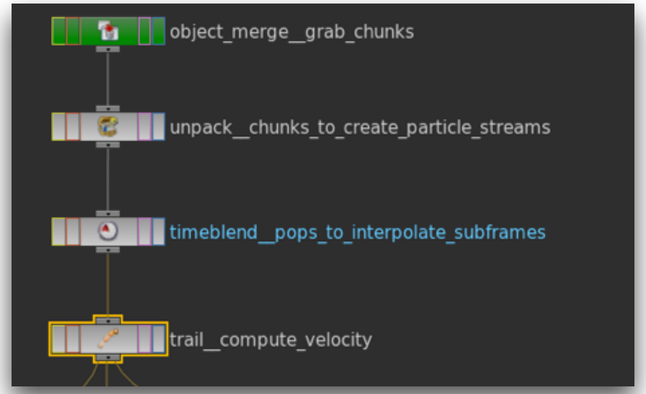
When I unpack there is no velocity

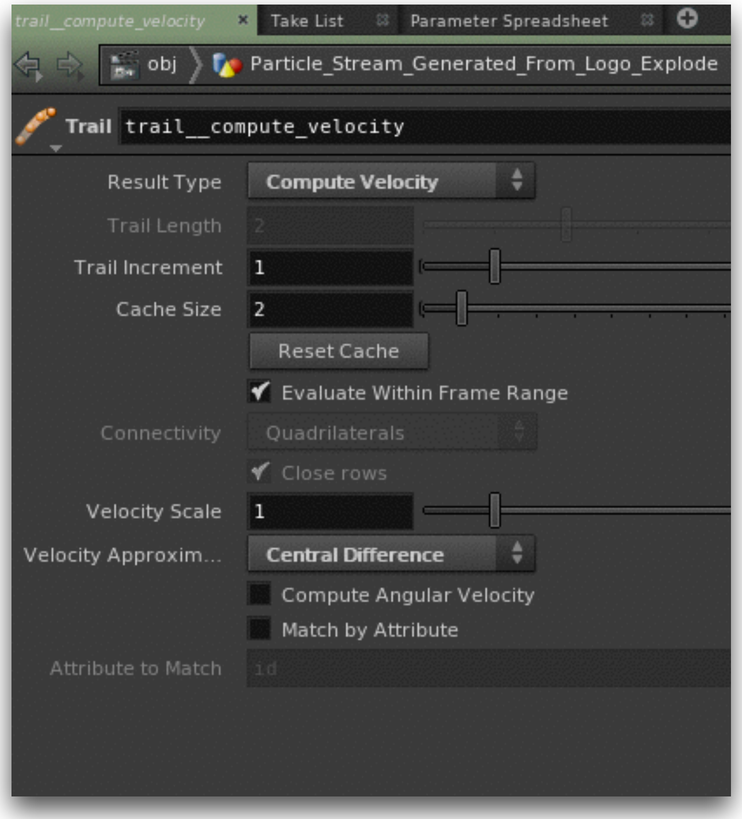
SIDE EFFECTS SOFTWARE

Append a Trail SOP

Recalculate Velocity

Append a Trail SOP





SIDE EFFECTS SOFTWARE

Where To Generate Particle Streams?

First, We probably do not want to place particle emitters on geometry that is not moving

There let us delete all primitives that are not moving (on a per frame basis)

We have to calculate the magnitude of the velocity for each primitive

Magnitude = $sqrt(Vx^2 + Vy^2 + Vz^2)$

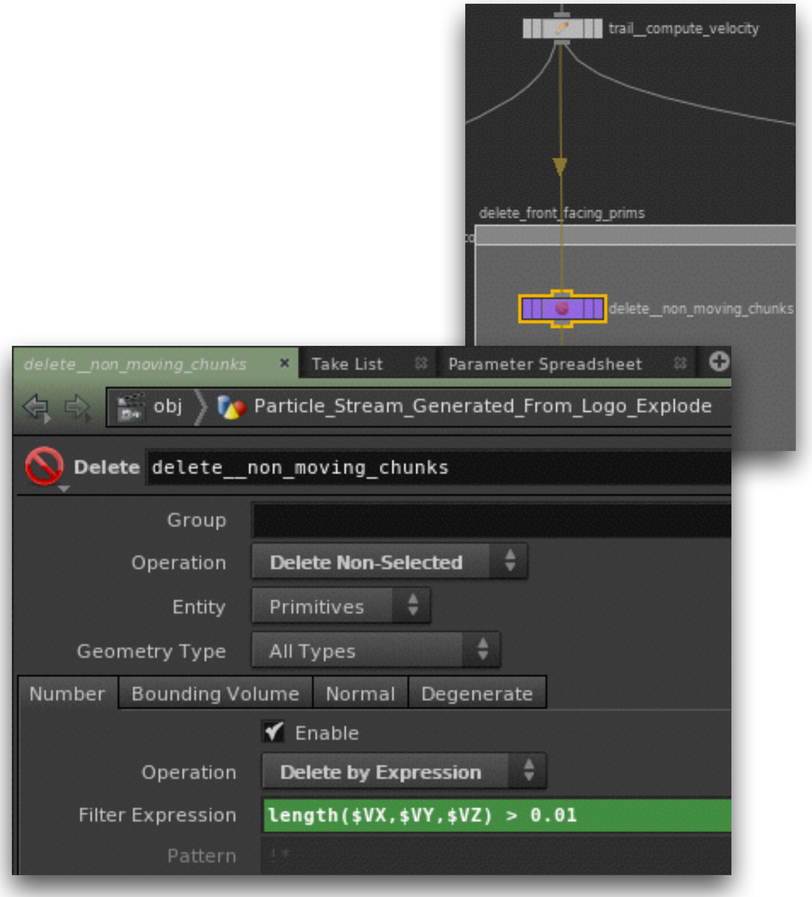
Append a Delete SOP (a Blast SOP will not work)

Operation - Delete Non-Selected

Number Tab

Operation - Delete By Expression

Filter Expression - length($VX, $VY, $VZ) > 0.01



SIDE EFFECTS SOFTWARE

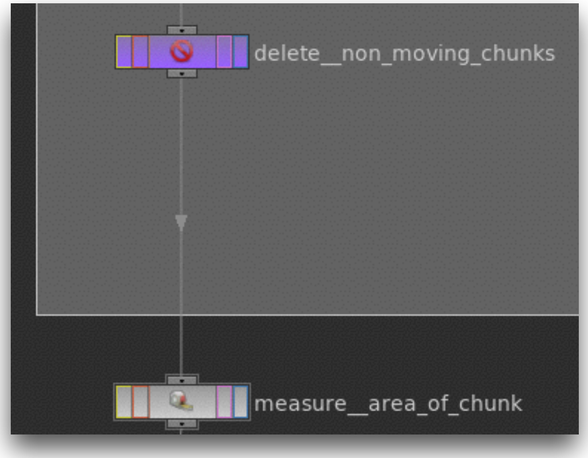
Why Can’t We Use the Blast SOP

Blast is designed to remove geometry that you select interactively in the viewport, as opposed to Delete which is a more procedural tool.

In addition to simply deleting unwanted parts of a model, the delete operator is useful for culling primitives from very dense geometry to speed up cooking of nodes downstream.

This delete operation can also delete profile groups (for example, 0.5 1.2-3.9) and mixed groups (for example, 0 0.5 4-12 1.2-3.9) causing the remaining hierarchy to update.

Where To Generate Particle Streams? (cont.)



We will eventually drop down a Scatter SOP on the remaining primitives to be the particle emitters

The Scatter SOP will distribute points equally on all prime.

This is probably not what we want

What would be better is for more points to be scattered on large primitives and on a few to be scattered on small prims

Append a Measure SOP

Type - Area

Where To Generate Particle Streams? (cont.)

So far we deleted prime that are not moving and calculated the area of each prim

Now, we really do not want points to be scattered on the forward faces of the moving prims.

To calculate the forward faces of the moving prims we can use the the velocity of chunks and the point normals.

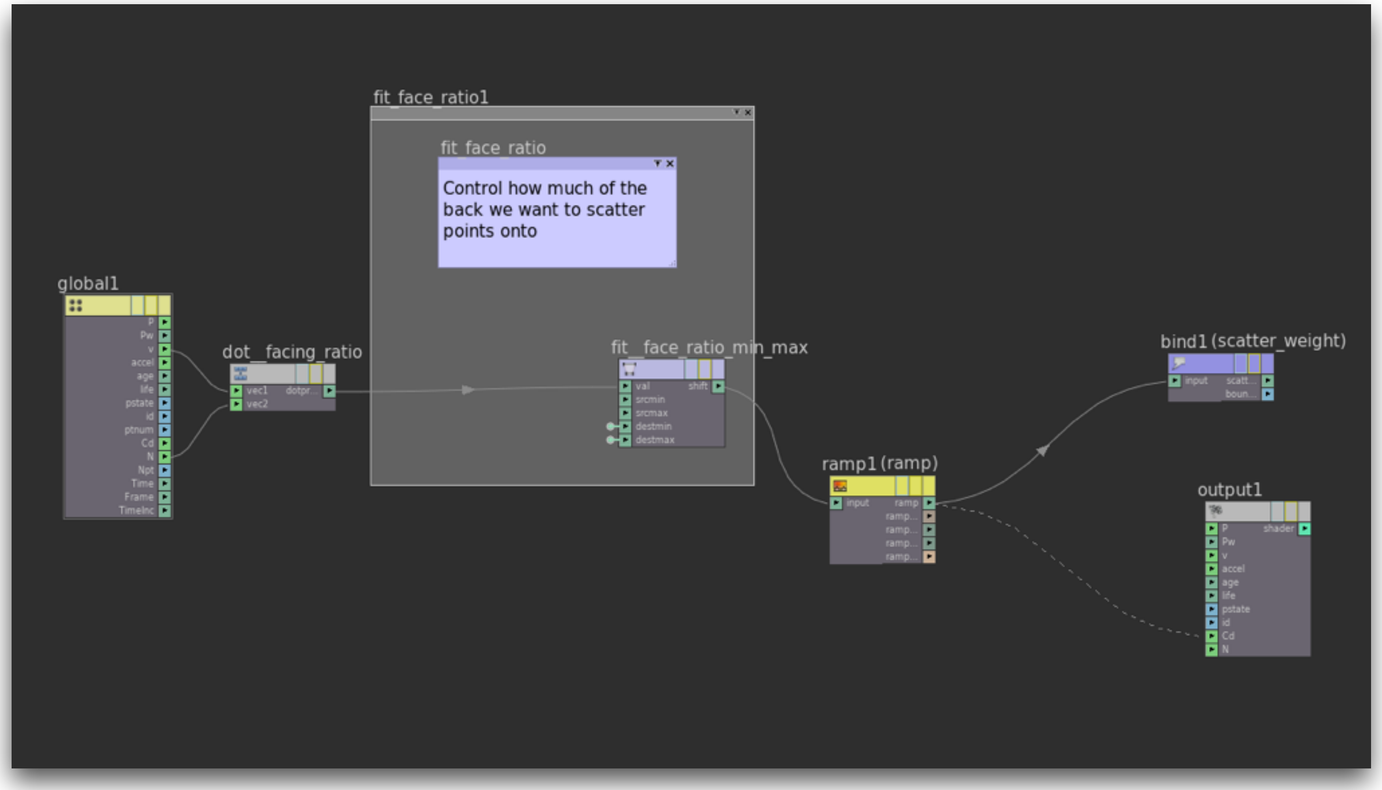
If the surface normal and velocity are facing the same direction we know that the prim is forward facing and we do not want points scattered on it

We can use a VOPSOP to calculate if the the two vectors (Point Normals and Velocity) are facing in the same direction

Calculating Direction of Point Normal and Velocity

Append a VOPSOP to the Measure SOP

Dive Inside and Layout the following Nodes



SIDE EFFECTS SOFTWARE

What is the VOPSOP Doing?

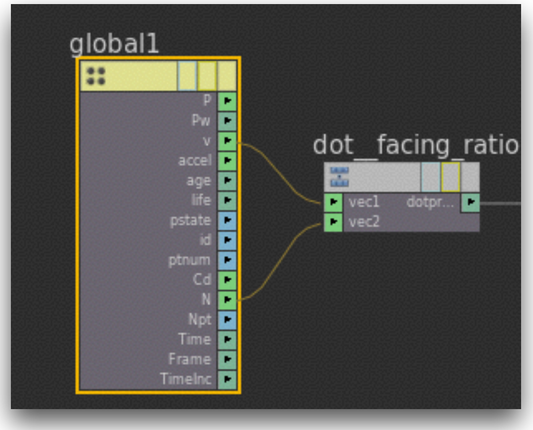
http://www.mathsisfun.com/algebra/vectors-dot-product.html

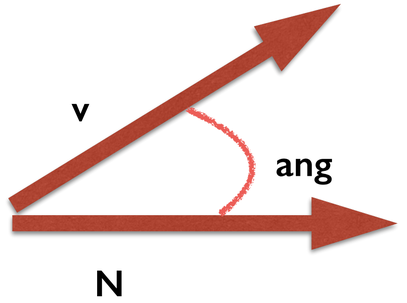
First we will take the dot product of Velocity and the Point Normal

v•**N** = |**v**|x|**N**|x cos(ang)

If **v** and **N** are facing the same direction |**v**|x|**N**|x cos(0)will be |**v**|x|**N**|x 1 but if the are 90 degrees apart the result will be 0.

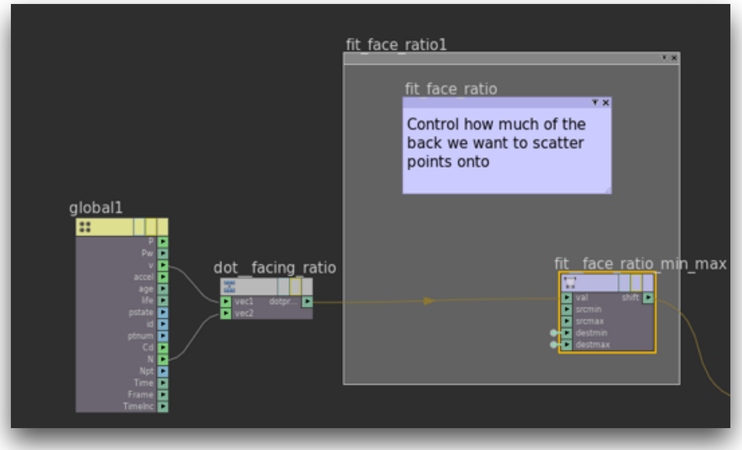
So we now know the closer v and N to the same direction the greater the result will be the further apart the values will go into the negative direction

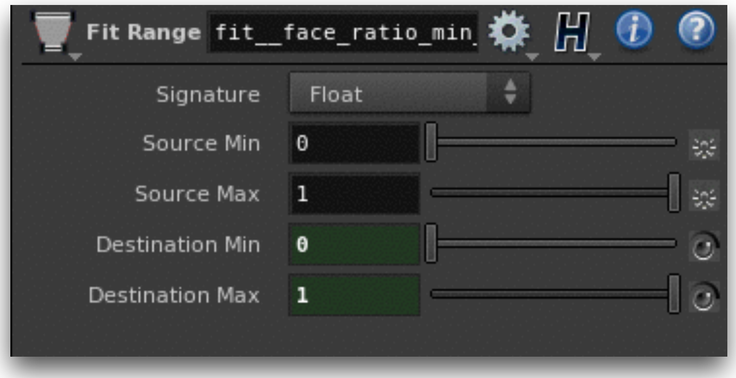




SIDE EFFECTS SOFTWARE

What is the VOPSOP Doing? (cont.)





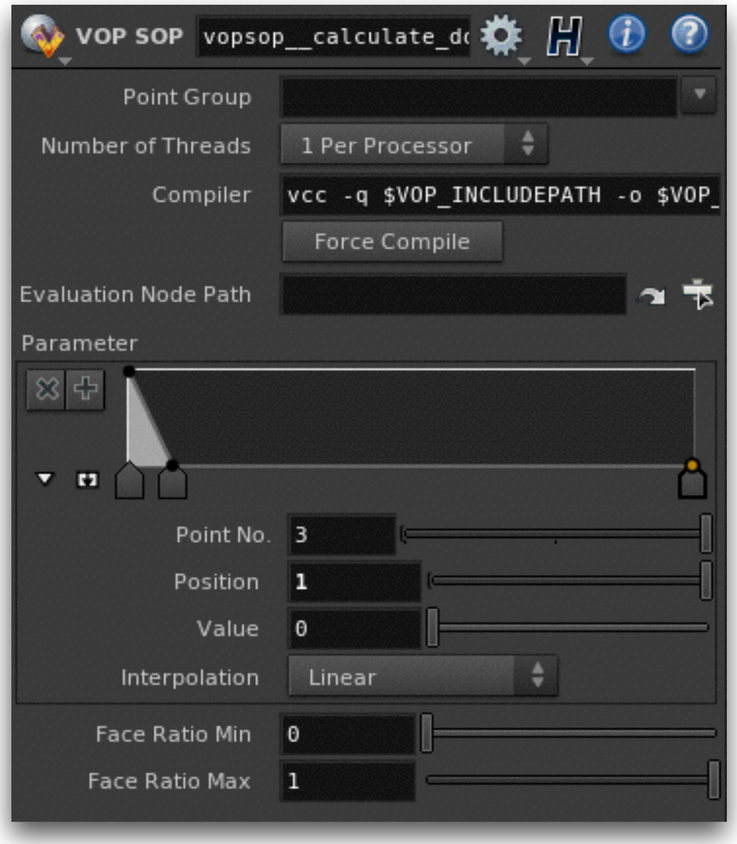
We might not want the emitters to be on the full range of primitives

For instance we might now want particle streams coming from the sides of the chunks

By appending a Fit operator we can control the range of emissions

SIDE EFFECTS SOFTWARE

What is the VOPSOP Doing? (cont.)

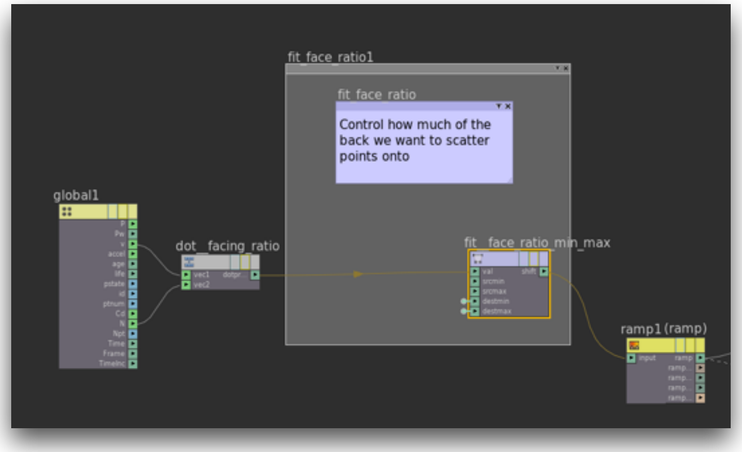


Right now the points would be scattered on the wrong side of the Chunks.

We can use a Ramp to negate the side the emitters will be placed and at the same time give the artist control over how much of the chunk should receive emitters

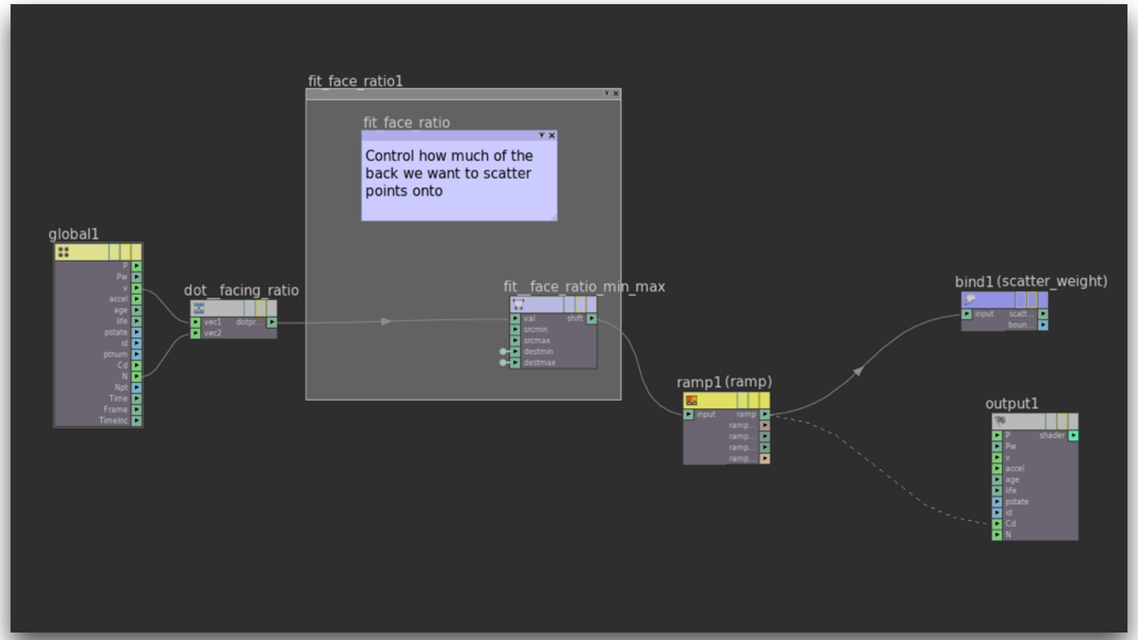
Append a Ramp - Set the ramp to the image on the left

If the values are close to zero place emitters



SIDE EFFECTS SOFTWARE

What is the VOPSOP Doing? (cont.)

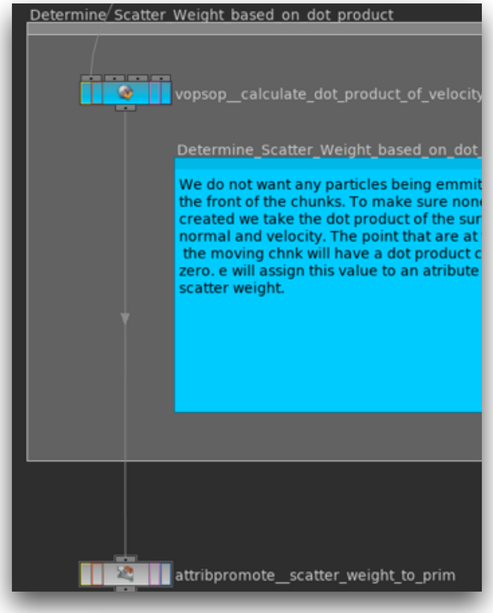


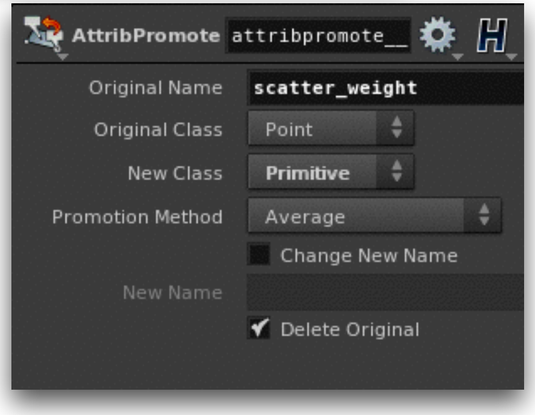
Append a Bind Export operator and name it “scatter_weight?

We will use this attribute as one of the factors in deciding where to place emitters

SIDE EFFECTS SOFTWARE

Back at the Geometry Level





The VOPSOP calculated a Point Attribute

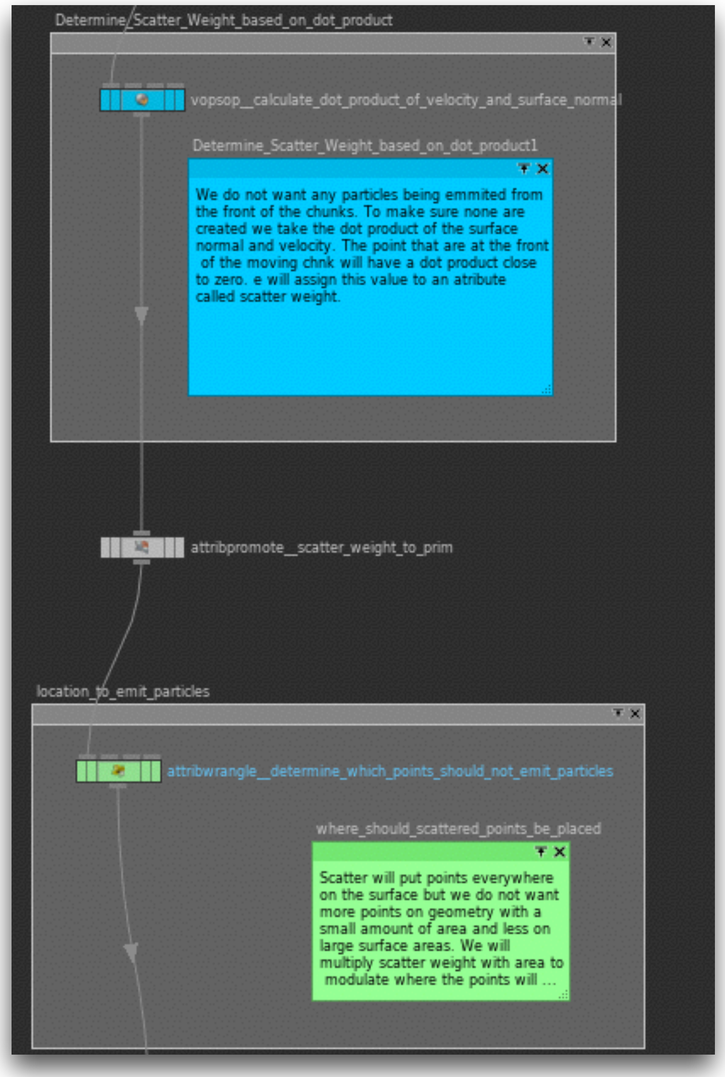
We want to calculate emitters at the Primitive level

Append a Attribute Promote

- Original Name - scatter_weight
- Original Class - Point
- New Class - Primitive
- Promotion Method - Average

SIDE EFFECTS SOFTWARE

Putting it all together

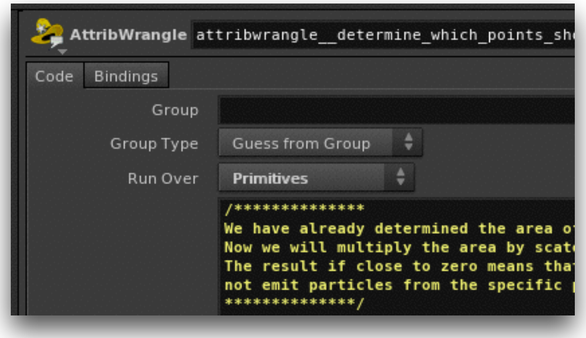


We are ready to calculate where to place are emitters.

- We know the area of each primitive
- We know the direction of each primitive

Append an Attribute Wrangle

- Run Over - Primitives



SIDE EFFECTS SOFTWARE

Putting it all together (cont.)

By multiplying the area we measured by the scatter_weight we will have a value that tells us to place emitters mostly where the geometry is large and facing away from the direction of movement

```
@area *= @scatter_weight;
```

Putting it all together (cont.)

If the area calculated is close to zero delete the emitter by setting @ dead to zero

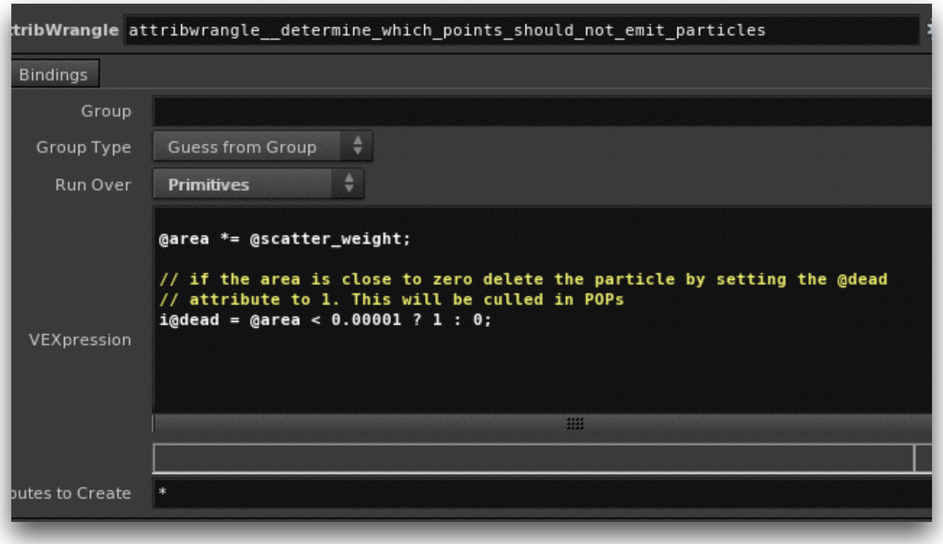
```
i@dead = @area < 0.00001 ? 1 : 0;
```

or if you like the long format

```
if (@area < 0.00001) {
      i@dead = 1;
} else {
      i@dead = 0;
}
```

If you want to previz the area you can just add

```
@Cd = @area*100;
```



SIDE EFFECTS SOFTWARE

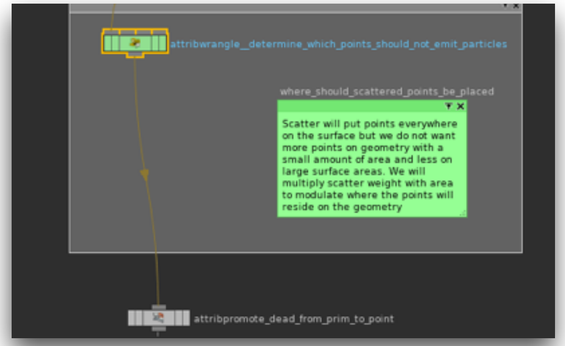
Promoting Dead

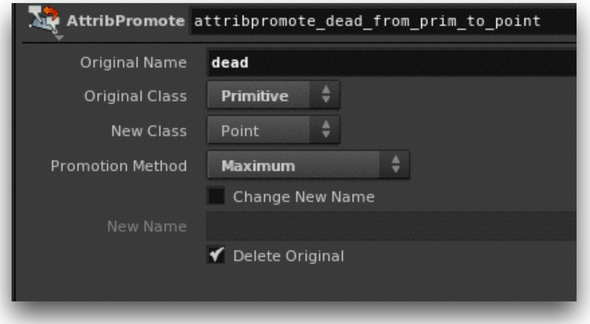
dead is currently a Primitive Attribute

We want to use dead to eventually delete particles

Append an Attribute Promote

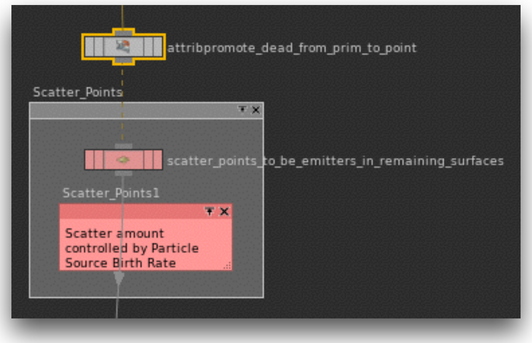
- Original Name - dead
- Original Class - Primitive
- New Class - Point
- Promotion Method - Maximum

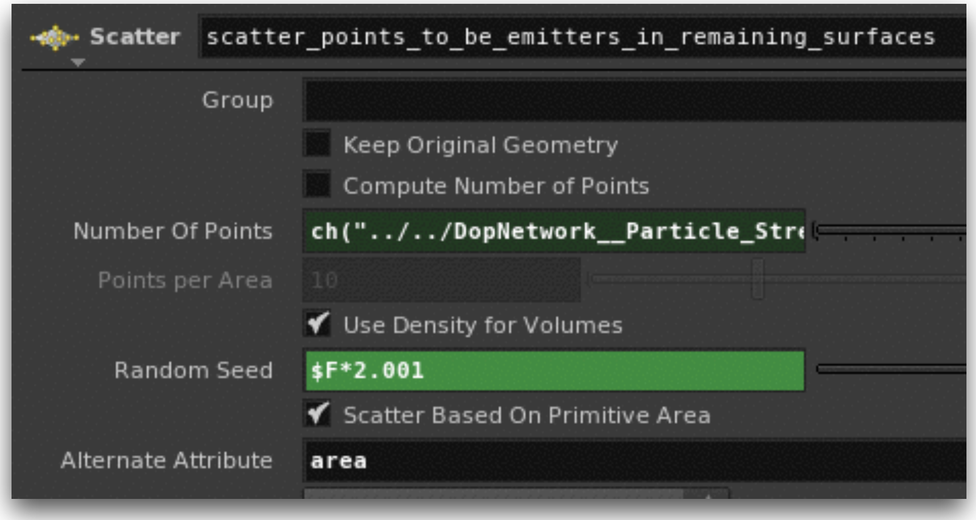




SIDE EFFECTS SOFTWARE

Scatter the Emitters





Yes, finally ready to scatter emitters

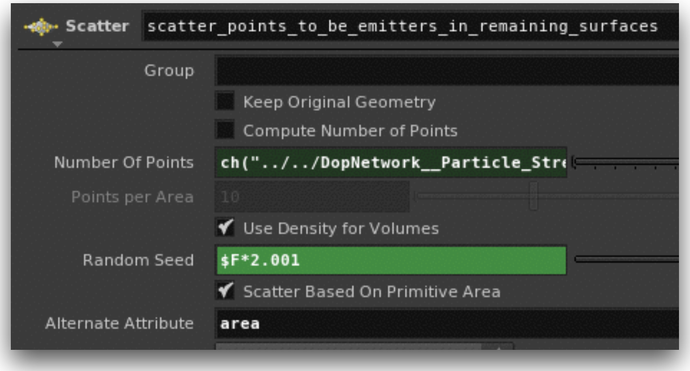
Append a Scatter SOP

Number of Points:

```
ch("../../DopNetwork__Particle_Stream_Generated_From_Logo_Explode/logo_explode_source/impulserate")
```

SIDE EFFECTS SOFTWARE

Scatter the Emitters (cont.)



We want the emitter to be located at a different location on the surface for each frame

```
Random Seed - $F*2.001
```

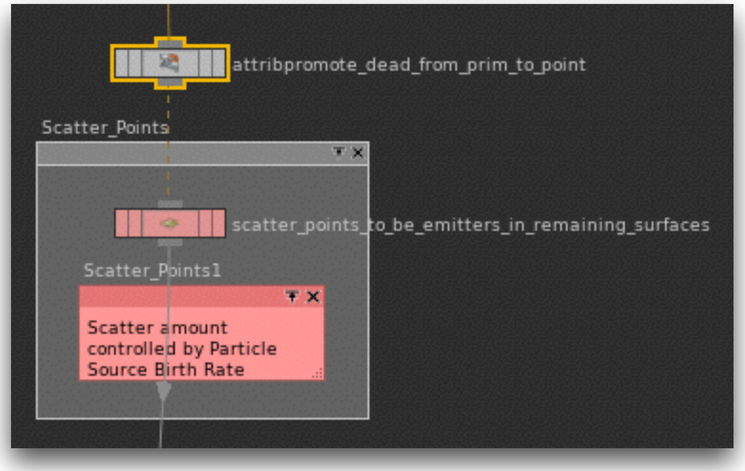
Append a Scatter SOP

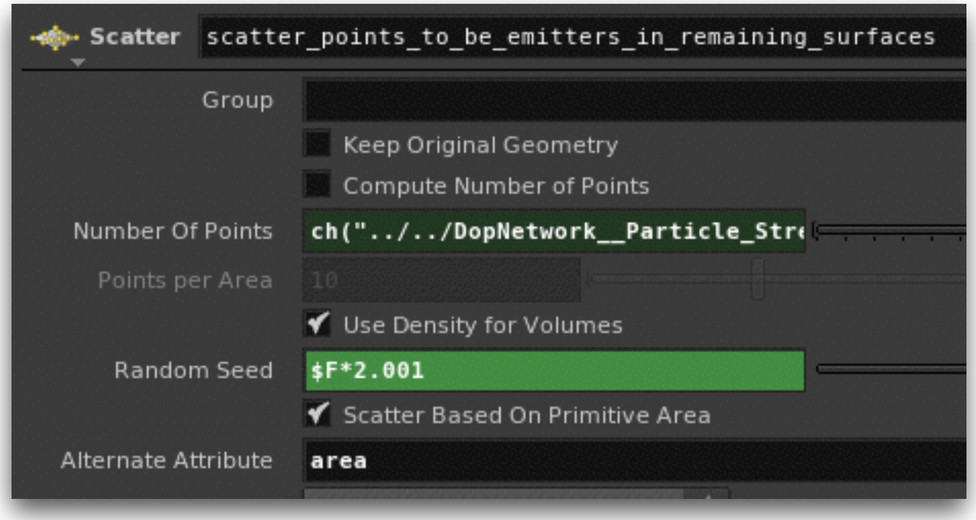
Number of Points:

ch(“../../DopNetwork__Particle_Stream_Generated_From_Logo_Explode/logo_explode_source/impulserate")

SIDE EFFECTS SOFTWARE

Scatter the Emitters (cont.)



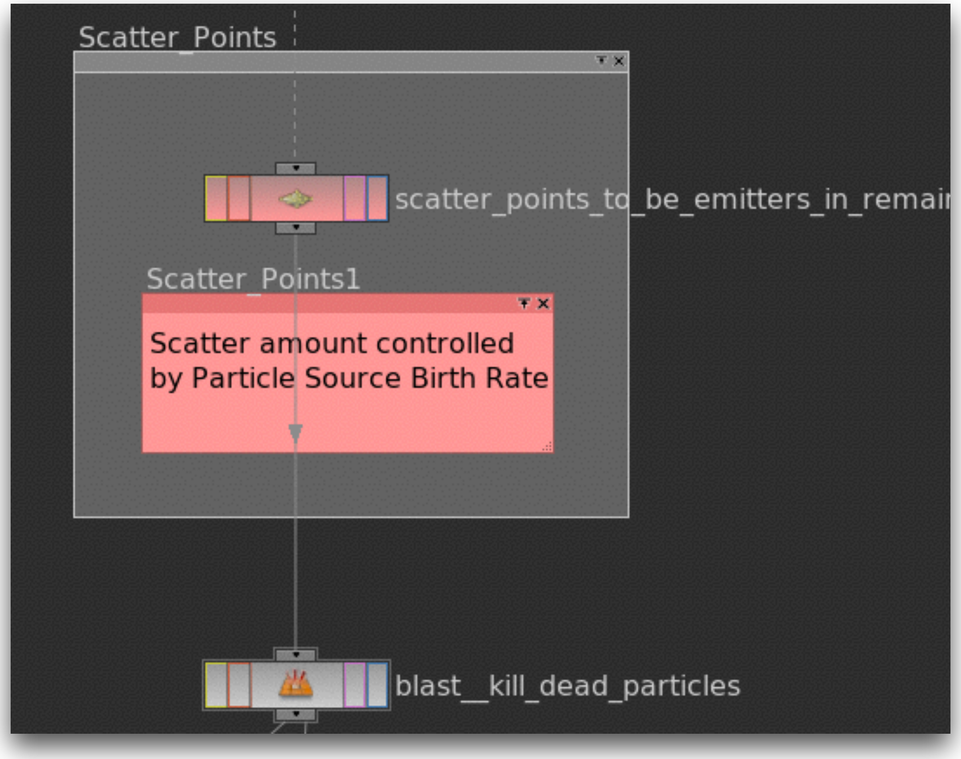


Now we want to scatter the emitters only in the area that the “area” attribute has calculated

Alternate Attribute - area

SIDE EFFECTS SOFTWARE

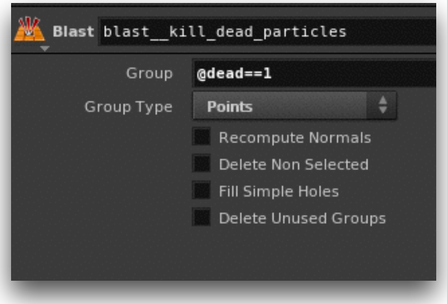
Kill the Dead Emitters



Now we want to kill the emitters that should not emit particle streams

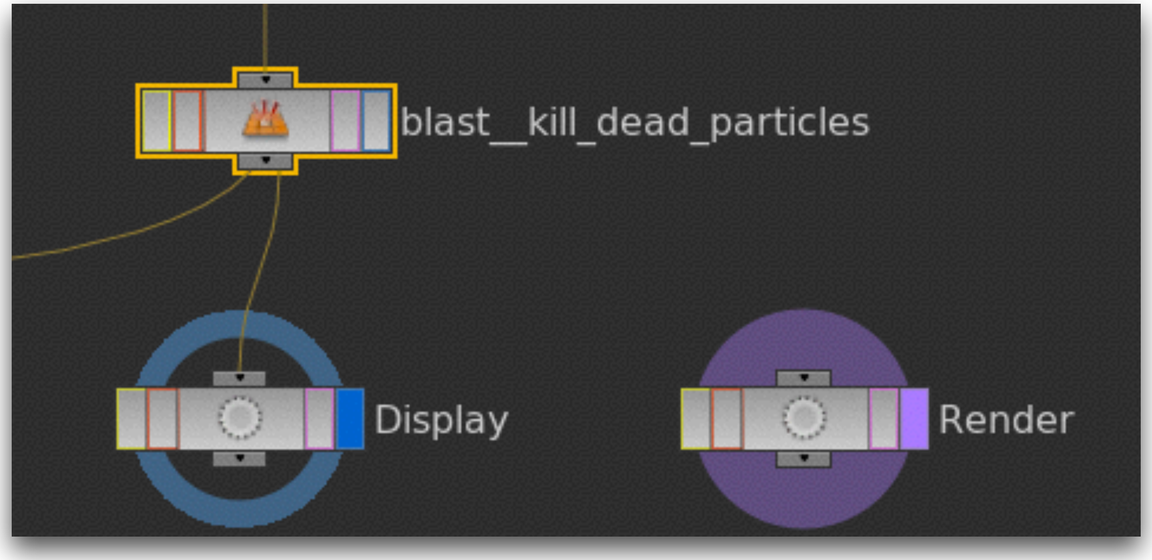
Append a Blast SOP

Group - `@dead==1`



SIDE EFFECTS SOFTWARE

Do Not Render the Emitters



You probably want to see the emitters in the Viewport but the should not be rendered

Add two Nulls one for Display, the other for Render. The Render is NOT wired to anything

End of Part 02 - Creating the Emitters for the Particle Stream

SIDE EFFECTS SOFTWARE

Part 03 - Creating the Particle Streams

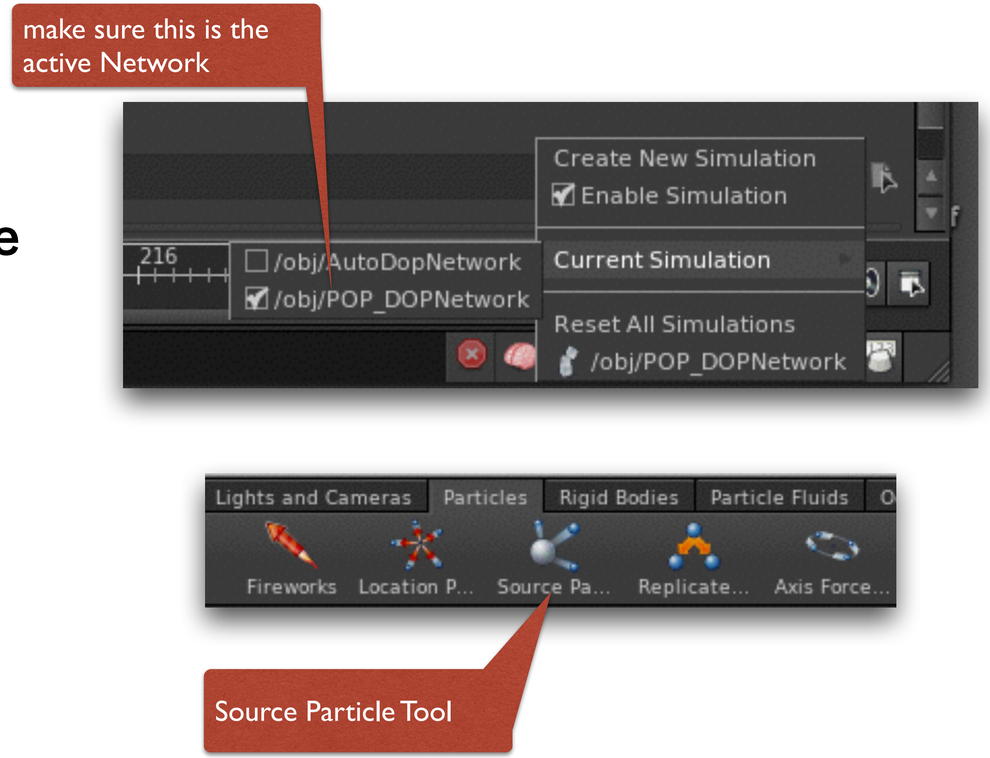
SIDE EFFECTS SOFTWARE

Creating the POPDopNetwork

Make Sure the POPDOPNetwork is the Active Network

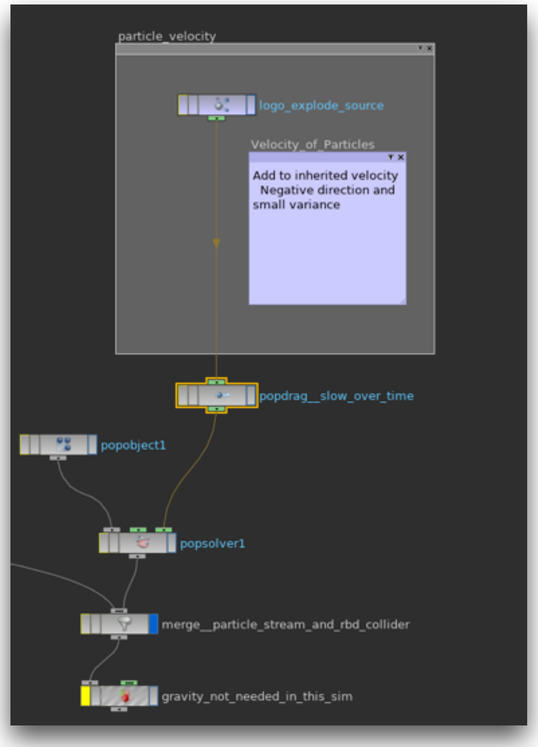
At the Object Level select the Logo_Explode_Source (the Object that contains the emitters)

In the Particle Shelf Tool Tab select the Source Particle Tool



SIDE EFFECTS SOFTWARE

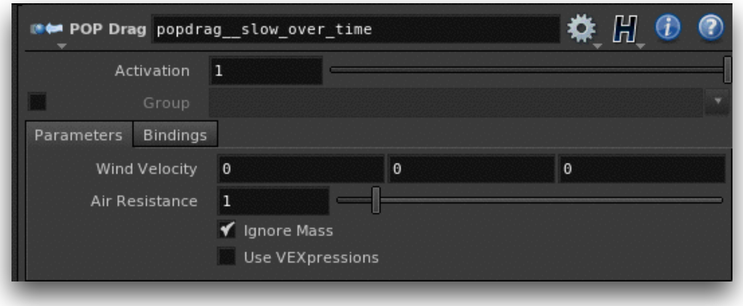
POPDOPNetwork



Dive back into the POPDOPNetwork and append a POPDrag Force

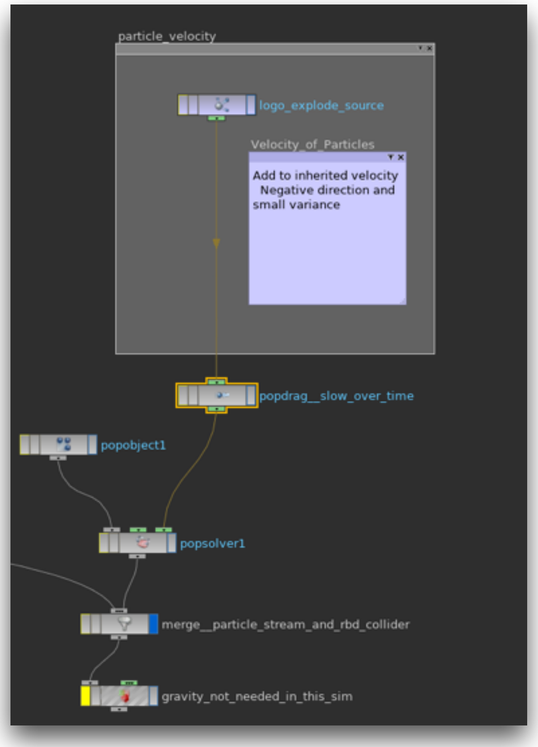
Air Resistance -1

Bypass the Gravity SOP. We do not want the particles to fall in this simulation



SIDE EFFECTS SOFTWARE

POPDOPNetwork (cont.)



We want the Particles to steam backwards with very little variance

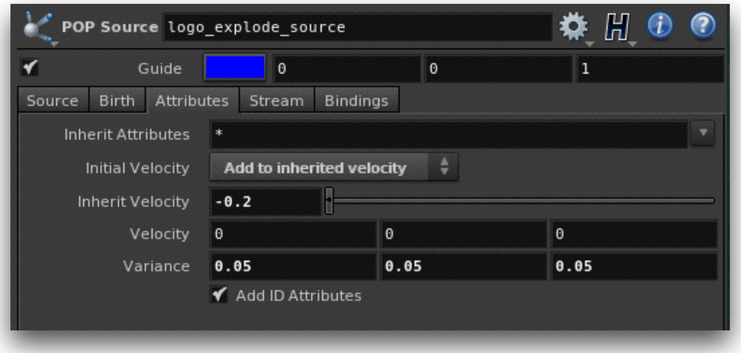
In the logo_explode_source (POP Source)

In the Attributes Tab

Initial Velocity - Add to inherited velocity

Inherit Velocity - -2

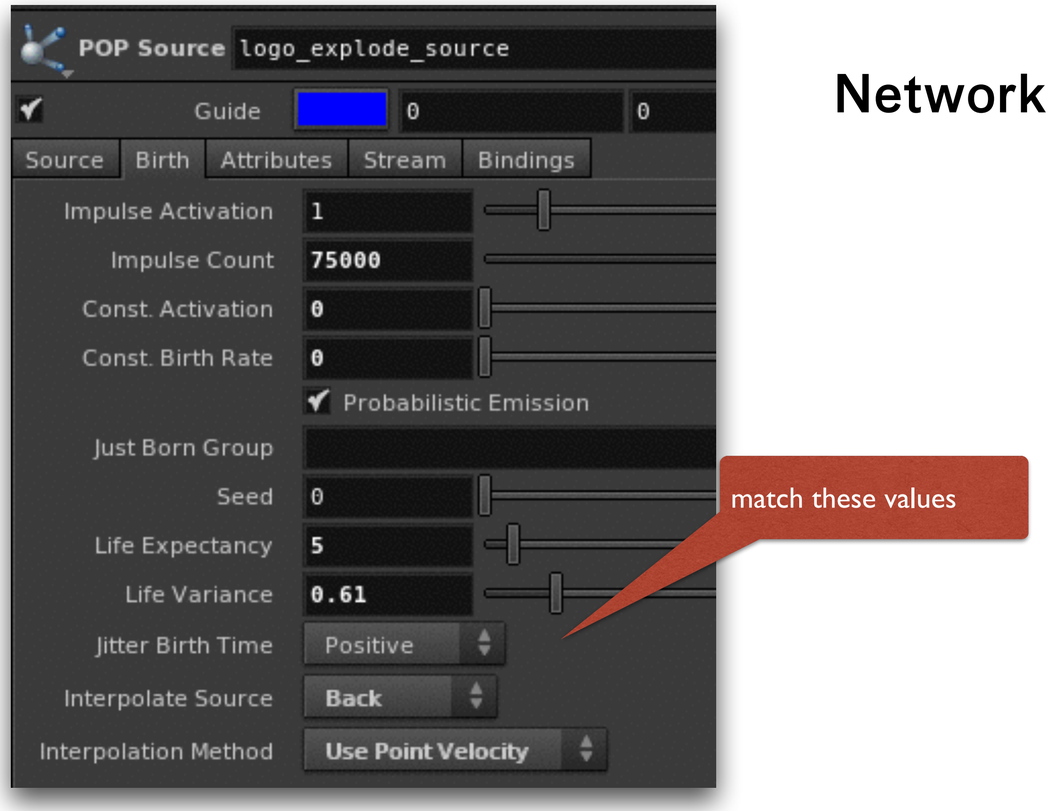
Variance 0.05, 0.05, 0.05



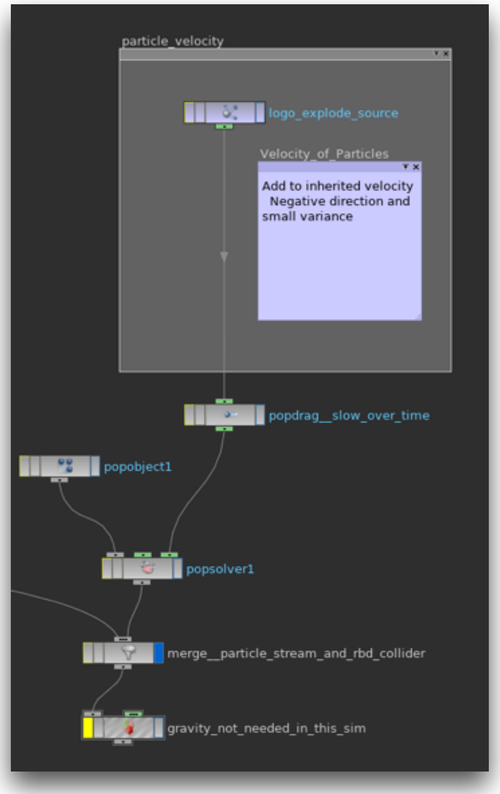
SIDE EFFECTS SOFTWARE

POPDOPNetwork (cont.)

Birth Tab



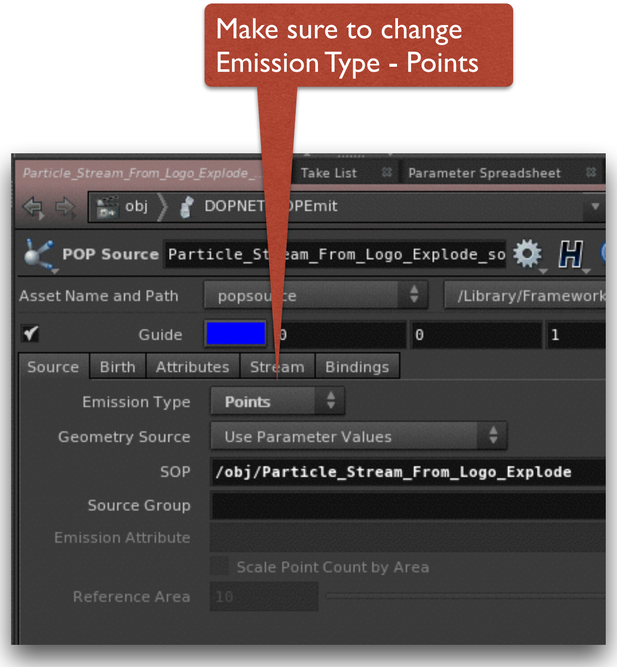
Network



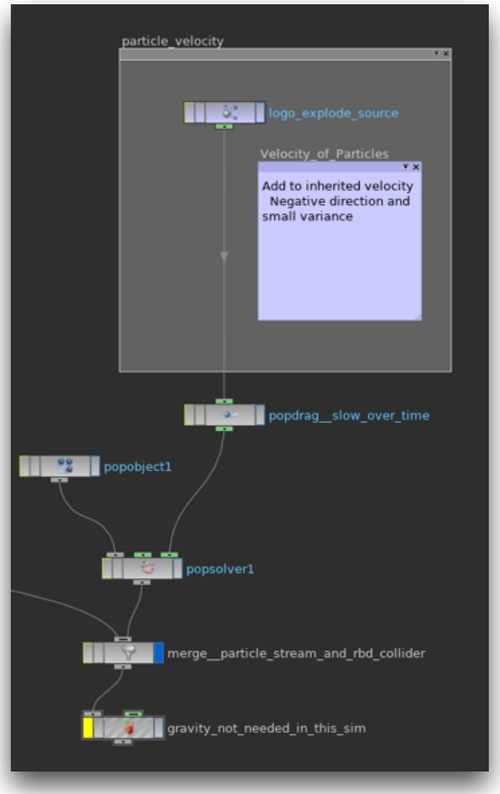
SIDE EFFECTS SOFTWARE

POPDOPNetwork (cont.)

Source Tab



Network



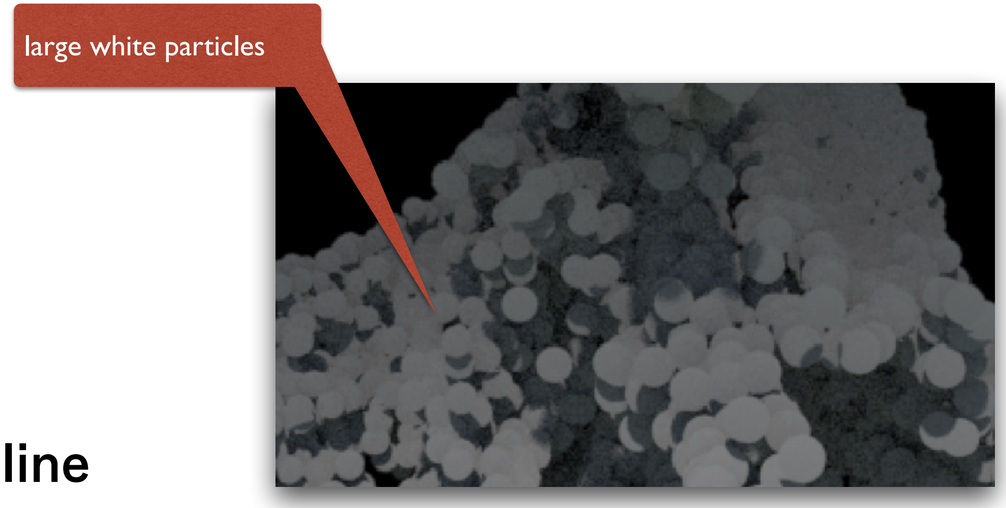
SIDE EFFECTS SOFTWARE

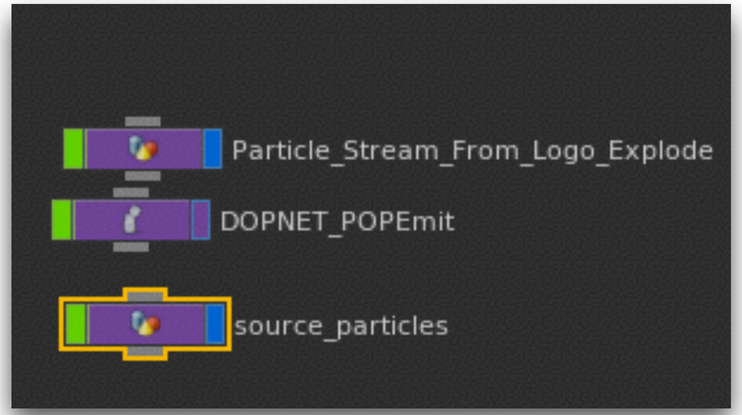
Source Particles

Right now if we render the particles will be large and white

We know that particle size is controlled by scale just line curve width is controlled by width

We can control the size of the particles base on normalized age Dive into the Source_Particles Object





Source Particles (cont).

Append an Attribute Wrangle node to the import_source

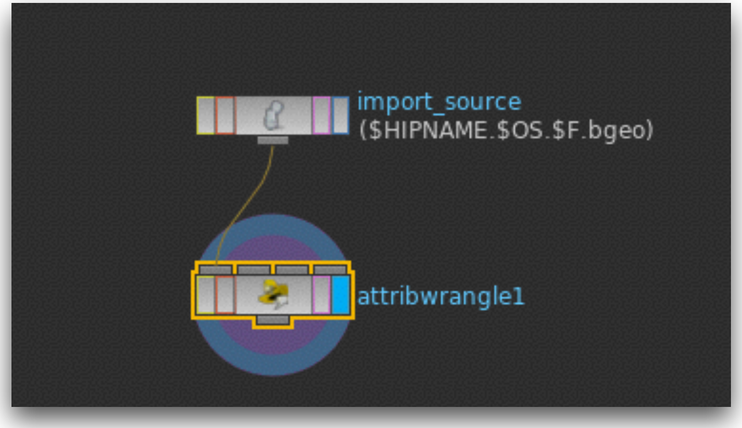
In the Attribute Wrangle make sure we are running over points

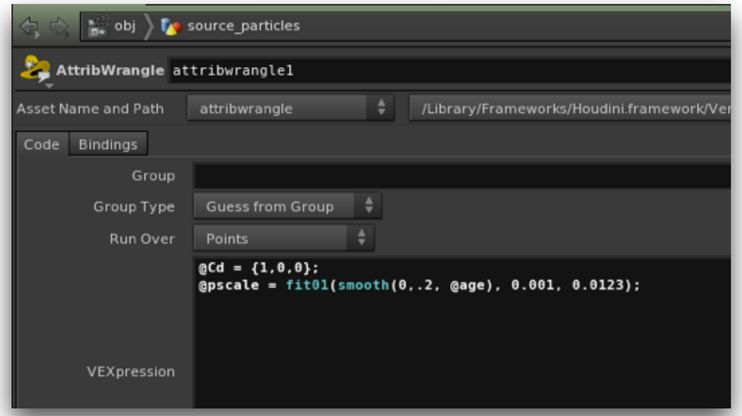
To make the particles red

```
@Cd = {1,0,0};
```

To make the size of the particles larger over time

```
@pscale = fit01(smooth(0,.2, @age), 0.001, 0.0125);
```





SIDE EFFECTS SOFTWARE

Smooth

Search Find

smooth VEX function

Computes a number between zero and one.

1. `float smooth(float value1, float value2, float amount)`
2. `float smooth(float value1, float value2, float amount, float rolloff)`

Computes a number between zero and one. Returns 0 if the amount passed in is less than or equal to value1, 1 if the amount is greater than or equal to value2.

If the amount is between value1 and value2, a smooth (easin/easeout) interpolation is computed. If a rolloff is specified, the inflection point of the blend will be shifted.

If the rolloff is greater than 1, the shift will be to the right, if the rolloff is less than 1 (and greater than 0), the shift will be to the left.

Fit01

fit01 VEX function

Takes the value in the range (0, 1) and shifts it to the corresponding value in the new range (`nmin`, `nmax`).

1. `float fit01(float value, float nmin, float nmax)`
2. `vector fit01(vector value, vector nmin, vector nmax)`
3. `vector4 fit01(vector4 value, vector4 nmin, vector4 nmax)`

Takes the value in the range (0, 1) and shifts it to the corresponding value in the new range (*nmin*, *nmax*).

SIDE EFFECTS SOFTWARE

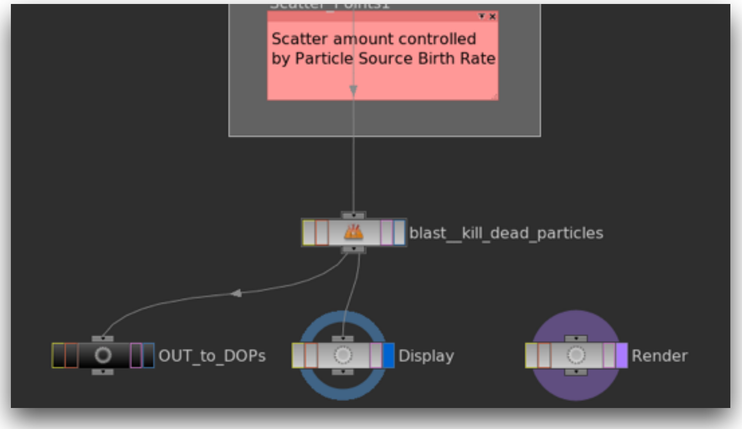
End of Part 03 - Creating the Particle Streams

SIDE EFFECTS SOFTWARE

Part 04 - Making the Particles Collide with the Chunks

SIDE EFFECTS SOFTWARE

Getting the Chunks for Particle Collision

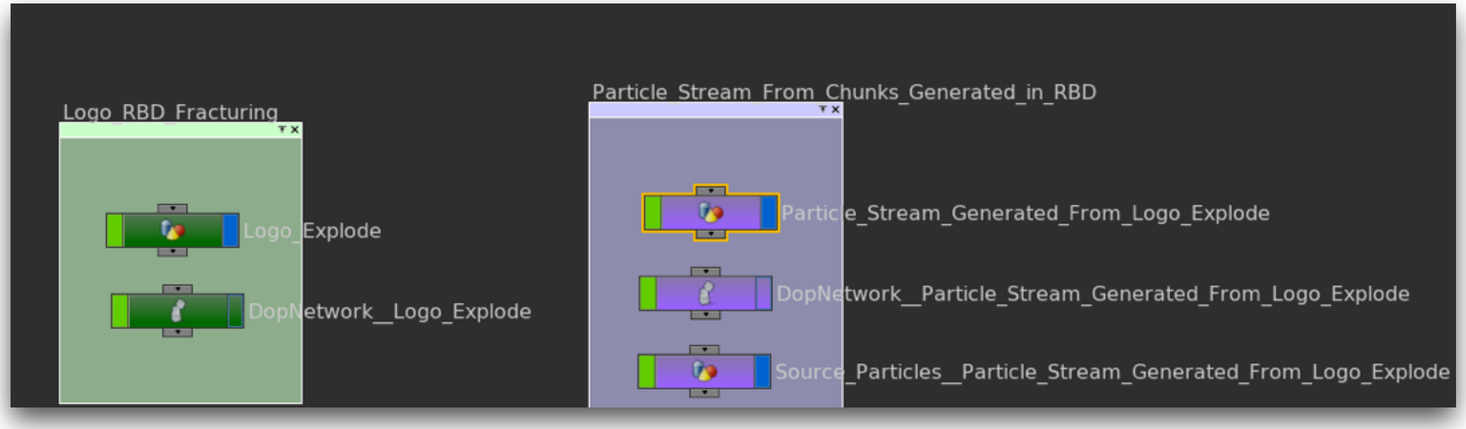


At the Object Level dive back into the Particle_Stream_Generated_From_Logo_Explode

Add one more Null next to the Display

Name it OUT_to_DOPs

Our Object Level Network so far



SIDE EFFECTS SOFTWARE

Creating the RBD Static Object for the Chunks

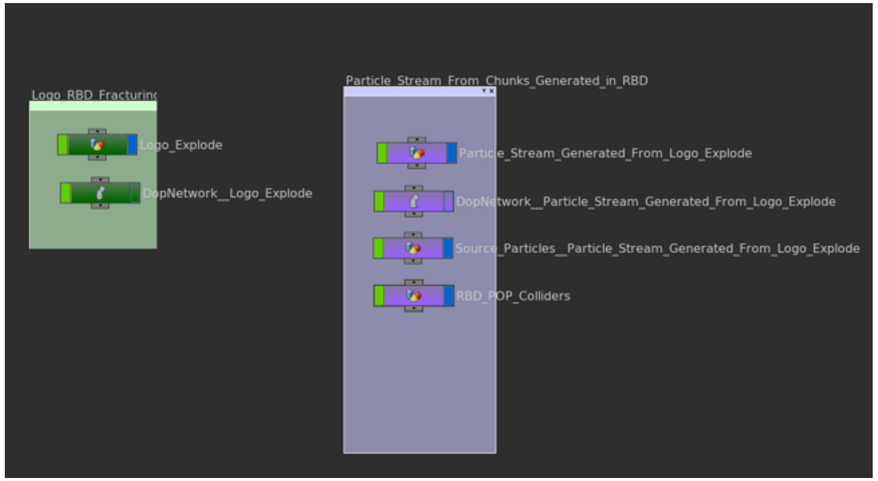
We need to Create an RBD Static Object for the particles to collide with the Chunks

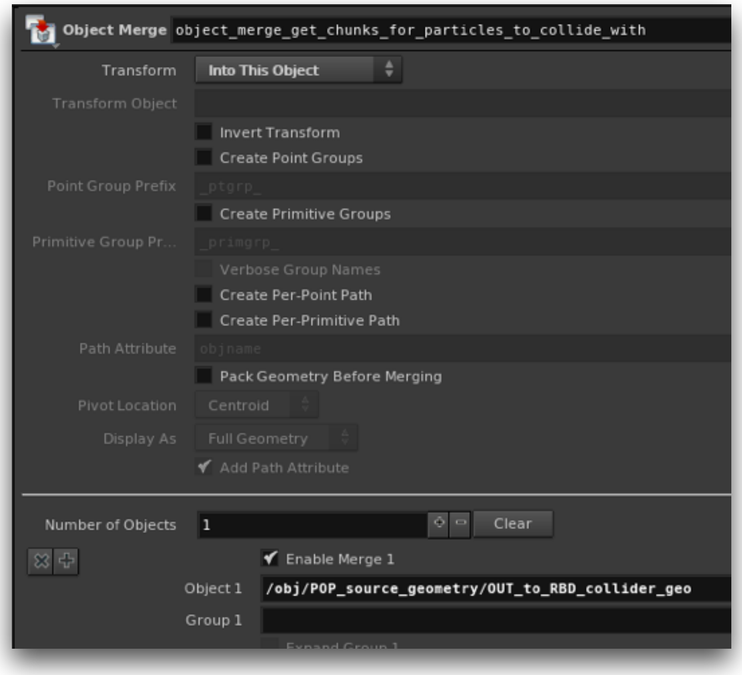
At the Object Level drop down a Geometry Object

Name it - RBD_POP_Colliders

Dive inside and drop down a Object Merge SOP

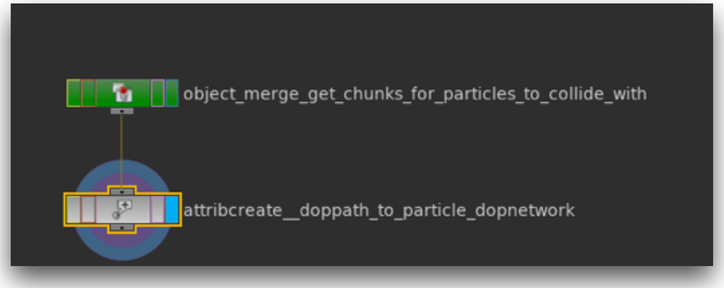
Object 1 - /obj/Particle_Stream_Generated_From_Logo_Explode/OUT_to_RBD_collider_geo

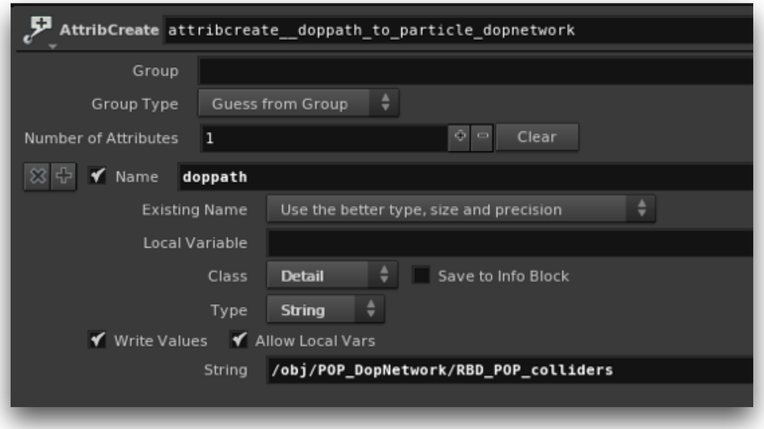




SIDE EFFECTS SOFTWARE

Creating an Attribute for the Static Object





We need a Detail Attribute to point where the geometry for the Static Object is located

Append a Attribute Create to the Object Merge

Name - doppath

Class - Detail

Type -String

String - /obj/POP_DopNetwork/RBD_POP_colliders

SIDE EFFECTS SOFTWARE

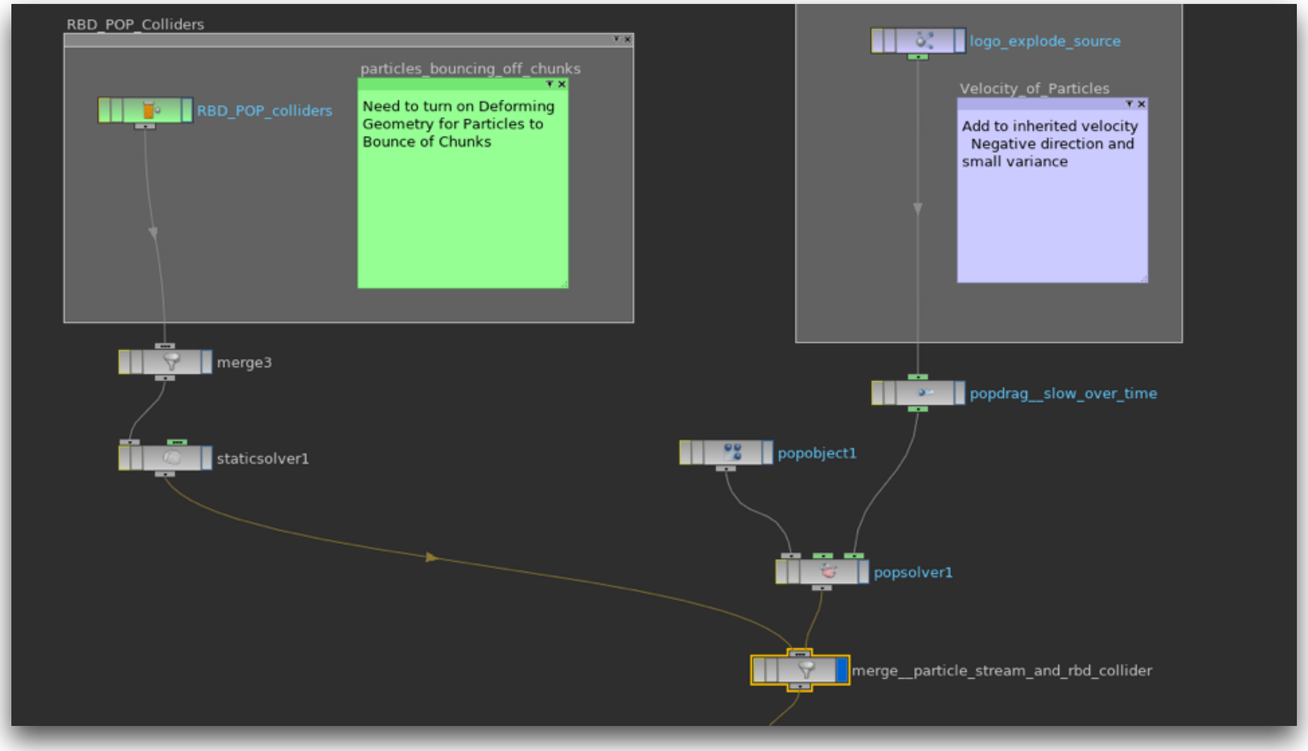
Creating the Static Object

At the Object Level make sure the POPDOP Network is active

Select the RBD_POP_Collider Object

In the Particle Shelf Tool Tab select Static Object

Dive back into the POPDOP_Network and see the results



SIDE EFFECTS SOFTWARE

End of Part 04 - Making the Particles Collide with the Chunks

SIDE EFFECTS SOFTWARE

Part 05 - Adding Materials

SIDE EFFECTS SOFTWARE

Packed Geometry

Because we are using Packed Geometry we will have to assign materials in a slightly different manner

The material SOP can not be used.

Instead we will use the PackEdit SOP

SIDE EFFECTS SOFTWARE

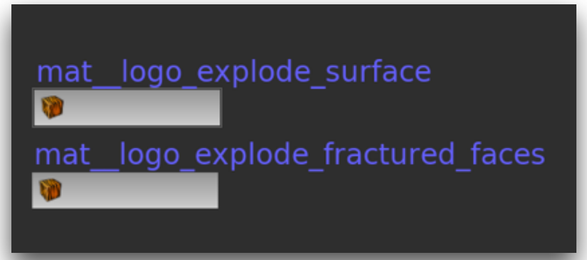
Adding Materials to the Chunks

At the Object Level drop down a SHOPNET

Name it - Materials_Logo_Explode

Dive Inside

Drop Down a couple of Material_Shaders

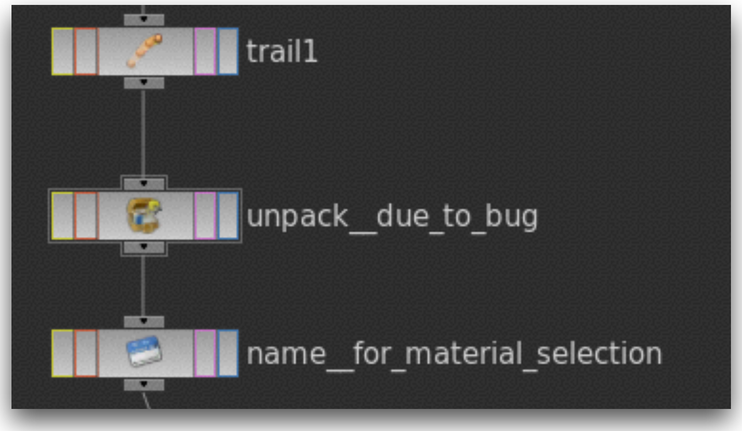


Name Material Shader 1 - mat_logo_explode_surface

Name Material Shader 2 - mat_logo_explode_fractured_faces

SIDE EFFECTS
SOFTWARE

Unpack for Material Assignments



At the Object Level Dive Inside the Logo_Explode Object

Due to a current limitation of Packed Primitives before we use the Name SOP and PackEdit we will have to Unpack

After the Trail SOP or Attribute SOP append a Unpack SOP

Then Append a Name SOP

Logo Explode Materials

For the Name SOP

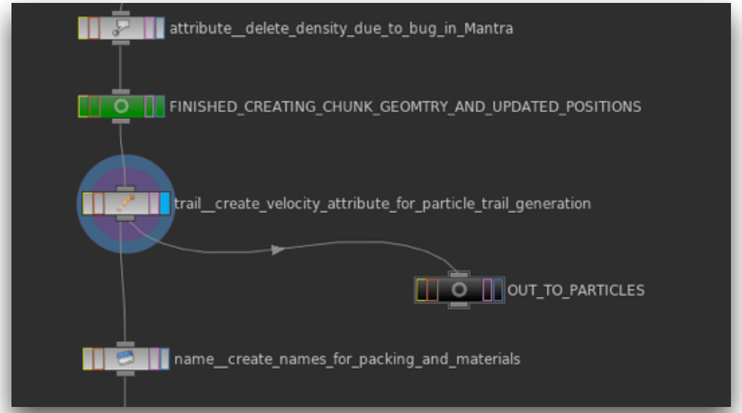
Attribute - name

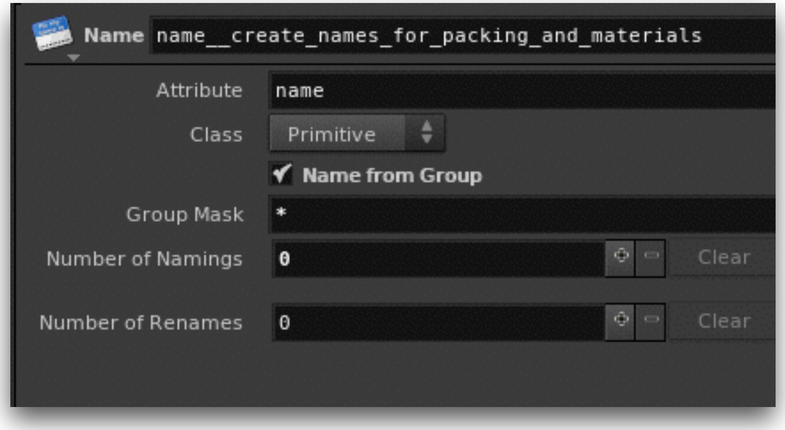
Class Primitive

Name from Group - Selected

Group Mask *

Currently, we just created the prim attribute “name” but there are no values

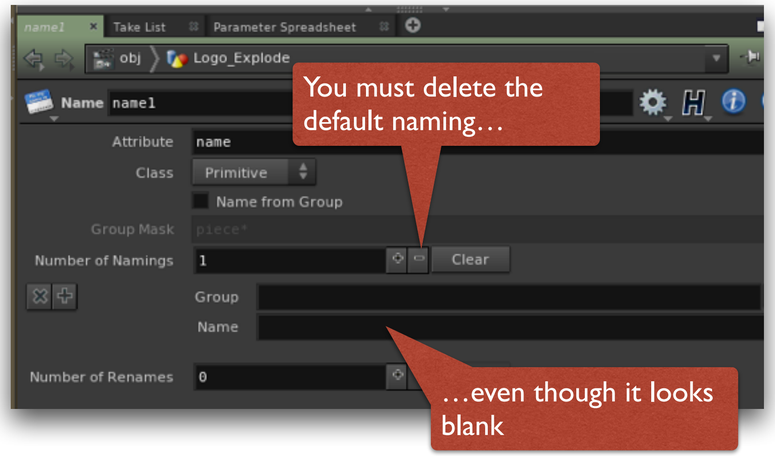


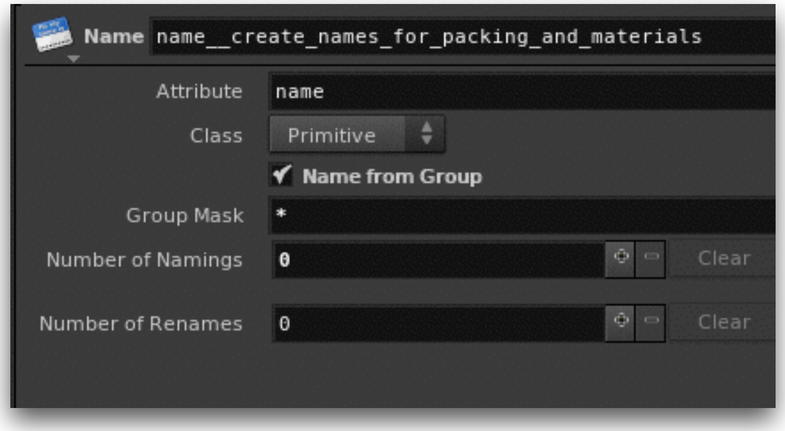


SIDE EFFECTS SOFTWARE

Warning about the Name SOP

When you drop down the Name SOP there is already one Naming
This MUST be deleted or else there will be naming that is blank





SIDE EFFECTS SOFTWARE

Name SOP

Name surface node

Creates a "naming" attribute on points or primitives allowing you to refer to them easily, similar to groups.

Naming points or primitives

This node sets an attribute (called `name` by default) on the selected points/primitives to a certain value. For example, you can select a single face, or small set of faces, on some polygonal geometry, and "name" it `foo` (that is, create a primitive attribute `name` with the value `foo`) on the faces. You can then refer to the primitives in subsequent nodes using `@name=foo`.

This is very similar to groups: using `@name=foo` is similar to creating a group `foo` and refering to it using `@foo`. However, name attributes have two differences from groups:

- Names are more efficient than groups. For example, you can assign

  different names to hundreds or thousands of individual faces, while creating hundreds of groups, each with a single member, would be problematic.

- Names enforce the constraint that each primitive can only have

  one name value at a time (that is, a primitive can't be included in both `@name=foo` and `@name=bar`), whereas a primitive can be a member of multiple groups. This constraint may be useful in some workflows.

Mantra uses named primitives to determine how to bind volumes to shaders during rendering. The Volume Vop node uses named primitives to determine binding if **Autobind** is on.

SIDE EFFECTS SOFTWARE

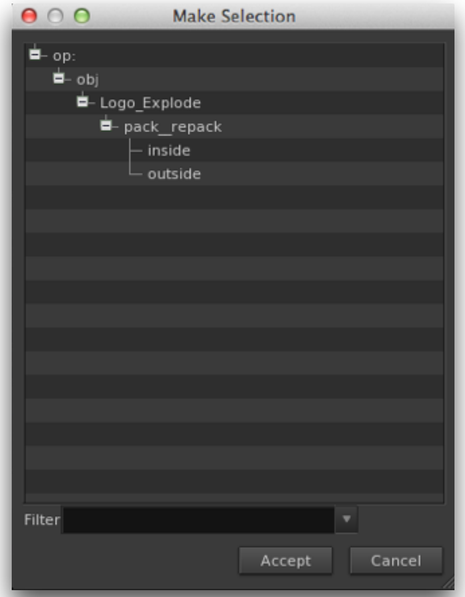
Logo Explode Materials (cont.)

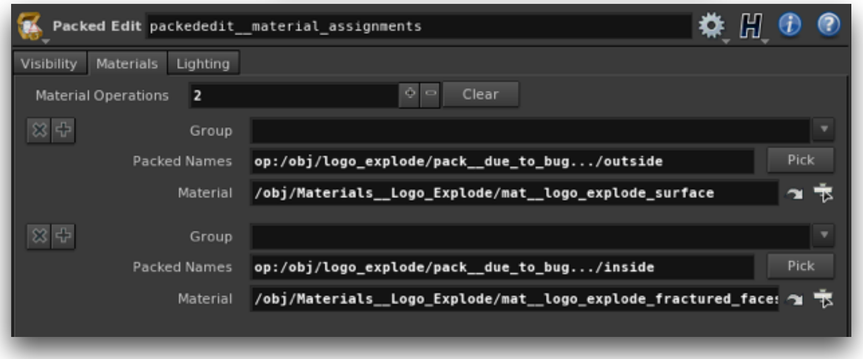
Append a Pack SOP (You might not have to do this. This is because of a bug in the current build I was using)

Append a PackEdit SOP

In the Pack Edit SOP go to the Materials Tab

- Packed Names - Click on the Pick Button
- In the Tree View that Pops up click down to “pack_repack_inside”
- Click Accept
- Material - Select the surface material you created
- Click the + button and repeat the process for the outside fragments





SIDE EFFECTS SOFTWARE

End of Part 05 - Adding Materials

SIDE EFFECTS SOFTWARE

Part 06 - Volumetric Lighting

SIDE EFFECTS SOFTWARE

Add a Camera

At the Object Level drop down a Camera directly facing the oncoming Chunks

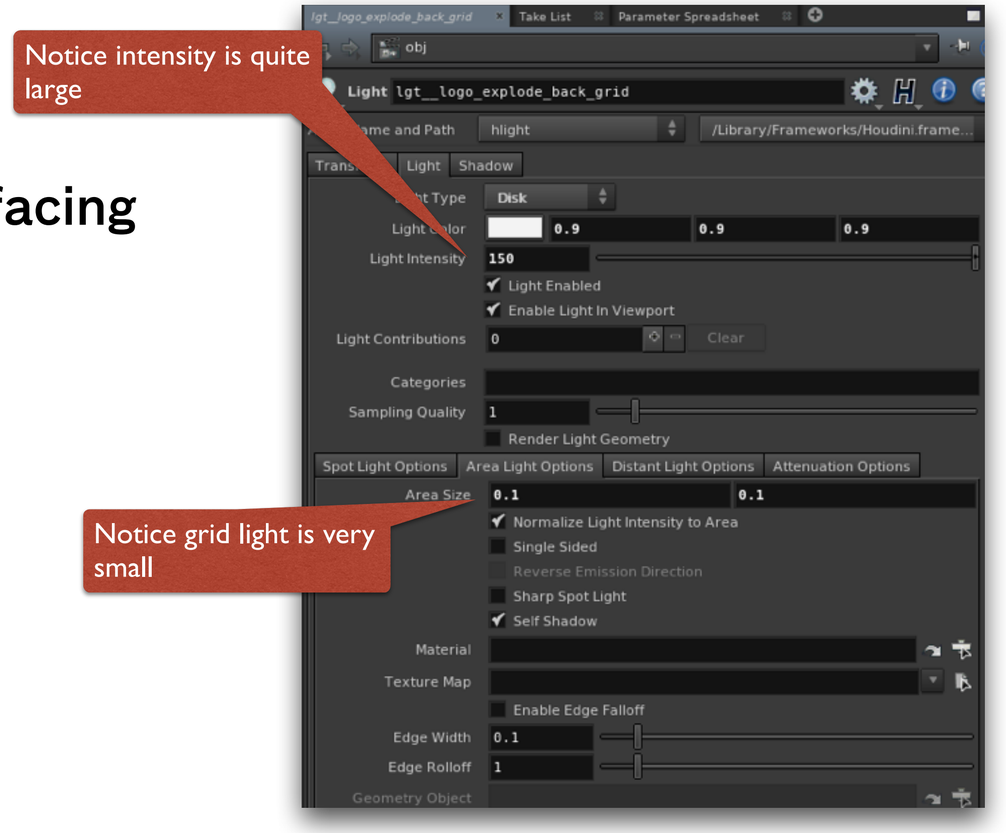
I named it “cam_logo_explode_front_50mm_16_9”

SIDE EFFECTS SOFTWARE

Add a Light

Add a small Grid Light behind the Logo facing the direction of the camera

I named it “lgt_logo_explode_back_grid"



SIDE EFFECTS SOFTWARE

Create the Volume

At the Object Level Drop down a Geometry Object

Rename it - “volume_frustum__for_light_rays”

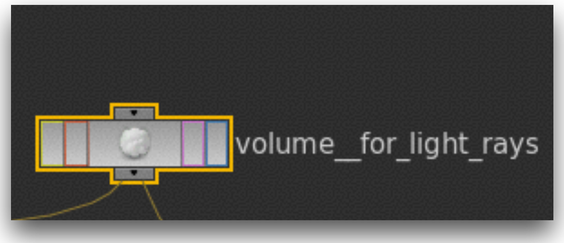
Drop inside

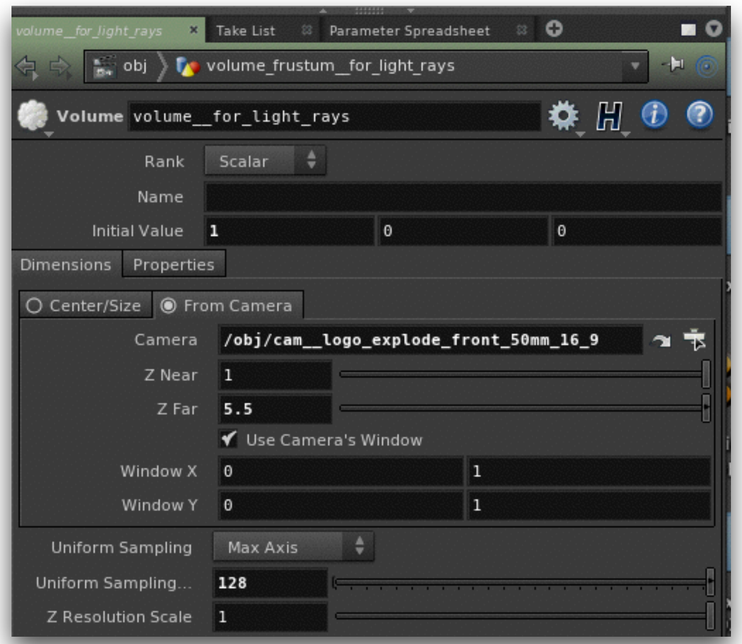
- Drop down a Volume
- Make the initial Value 1,0,0
- Rank - Scalar

In the Dimensions Tab Select the “From Camera” Tab

- Set Camera to - “cam__logo_explode_front_50mm_16_9”
- Set the Z Near - 1
- Set Z far - 5.5
- Uniform Sampling - Max Axis
- Uniform Sampling - 128

Render





SIDE EFFECTS SOFTWARE