

WINTER EFFECTS FOR COMPUTER GRAPHICS

MASTER THESIS

UDHAYA SANKAR RAVI SANKAR

(i7811900)

MSc COMPUTER ANIMATION AND VISUAL EFFECTS - 2009

NCCA

CONTENTS

1. ABSTRACT	4
2. INTRODUCTION	5
3. PREVIOUS AND CURRENT WORKS	6
4. INITIAL RESEARCH	8
5. TECHNICAL BACKGROUND	8
6. CREATING TOOLS FOR THE ANIMATORS	9
WINTER FX SNOW ACCUMULATOR	9
6.1.1 TECHNICAL ASPECTS	11
6.1.2 ALGORITHMS	11
6.1.3 IMAGES	15
6.2 ICICLES GENERATOR	16
6.2.1 TECHNICAL ASPECTS	18
6.2.2 ALGORITHMS	19
6.2.3 IMAGES	24
6.3 SNOW BY TEXTURES CREATOR	26
6.3.1 TECHNICAL ASPECTS	28
6.3.2 ALGORITHMS	29
6.3.3 IMAGES	32
6.4 PAINTING ON SURFACES TO CREATE SNOW	35

6.4.1 TECHNICAL ASPECTS	37
6.4.2 ALGORITHMS	38
6.4.3 IMAGES	42
7. SHADING	42
8. RENDERING	44
9. CONCLUSION	45
10. REFERENCES	46

1.ABSTRACT

In this project, I present a new method to create winter effects for a totally 3D environment and can also be used for live action integration. The techniques used here are mainly for snow accumulation for various objects in the scene, like houses, gardens cars and for even a huge environment which can be changed to a winter season look. The whole idea was an inspiration from the movie *POLAR EXPRESS (2004)* directed by **Robert Zemeckis**. The totally computer animated feature film was done by **SONY IMAGE WORKS**. The Environment in this movie is completely stylized and the snow effects are stunning. Based on this idea I wanted to create an automation which can be easily used by the artist to create Snow accumulation and Icicles for various objects in the shot.

2. INTRODUCTION

Winter season in any place is always one of the nature's most beautiful creations. The combination of snow accumulation and falling is very interesting in an artistic way and as well as the physics involved in it. It changes the whole mood of the environment and gives a very fresh illusion to our eyes.

There are lots of effects to create a particular mood for an environment in films and games. As an example we can take a war scene, where we see lots of smoke and fire to actually bring the shot more involved to the audience and it adds up as a technique for good story telling. There are loads of research behind fluid dynamics for creating smoke and fire. Water simulations are also a very interesting challenge in computer graphics. Snow is a very important physical effect in our world but there is not more research towards it for simulating in computer graphics.

It still remains as a time consuming and tedious process for the animators to actually create this sort of snow accumulation effect.

For few movies the modelers had to actually model the shape of snow on various objects in the shot. Then they follow by tweaking for a good visual impact. Icicles are also modeled for the foreground objects to show more details and photo realism.

Snow accumulation and falling has a very complex physics in the nature by itself. In order to simulate very accurately is a very big challenge to the research and development team.

3. PREVIOUS AND CURRENT WORKS

One of the approaches for snow accumulation on objects was Fearing (2000) [1] “Computer modeling of Fallen Snow”. The system calculates the amount of snow a particular surface can receive, and then simulating the interactions with the other objects. It also calculates the stability of the surface and how the snow moves from unstable areas. It is basically shooting particles and calculating the surfaces where snow can accumulate. It also generates mesh out of them.

Nishita et al.(1997) [2] presents a modeling and rendering method for snow by using meta balls. In this the density distribution of the snow is defined by using meta balls.All the meatballs have their radius, center and density at center of each of he meta ball.The snow mesh is given my the isosurfaces of potential fields defined by the set of meatballs(Nishita et al.1997).

Moeslund et al.(2005) [3] talks about snow flake generation, motion and interaction with the wind and accumulation. According to them snow flakes are formed by ice srystals, taken this into consideration they model a snow flake by combining triangular polygons to random manner. Accumulation is done by dividing the scene into larger entities which have the same orientation and calculating where exactly snow has to be accumulated. It is basically emitting snow from the sky and determining collision detection.

There are many research behind snow falling but considering zero thickness. Icicles also form the basic and the finalized look for a winter shot. They can be made as a special effect or by CG.

An article in the sciencedaily.com explains about the shape of the icicles. A science video by the university of Arizona Martin Short turned to mathematics to find out the shape of icicles and why they are thin in the edges. It basically explains The height is proportional to the radius to the four-thirds(Martin Short).

Live action movies such as Band of Brothers, Independence Day, 101 Dalmatians, Alien vs Predator, Gladiator and lots more have used special effects done by the company called Snow Business located in Gloucestershire, United Kingdom. They actually provide solutions for all the type of winter effects needed for a particular shot. They also work for commercials and television productions. The various effects supplied are

- 1) Settled snow cover
- 2) Falling snow
- 3) Display dressing snow
- 4) Frost

- 5) Ice
- 6) Arctic effects
- 7) Icicles
- 8) Snowballs, snowmen and igloos
- 9) Indoor falling snow



fig 1. Image from Snow Business



fig 2. Image from Snow Business



fig 3. Image from Snow Business

4. INITIAL RESEARCH

I first started looking into several movies like “Polar Express”, “The Day after tomorrow “and “The Band of brothers”.

I had several photographs which I clicked in this winter.

I started approaching artists and asking them their view in the usability of the tool.

I got ideas from them and got a path to start working on various ideas.

I met Sofronis Efstathiou, programme Leader MA3D NCCA, Bournemouth University and showed some of my initial tests in the working of the tool and then he suggested me to add a few interesting features which will make the artist really happy to model the snow accumulation.

I had a chat with the other artists like Dennis and Kishore Guntuku from MA 3D computer Animaton NCCA. and got feedbacks from them as well.

5. TECHNICAL BACKGROUND

The winter effects for computer graphics will be entirely created with side effects Houdini. It is the most high ended 3D package which is most commonly used for effects and dynamics for feature film and games.

In major I am focusing on the snow accumulation on various objects in the shot, Icicles generation for both static and dynamic effects and rendering them.

I have mainly focused on three different ways for snow accumulation in which I believe it would be easier for the artist to accomplish the sort of accumulation and the flexibility in the controls so that the shot can also be driven by the Art Director.

Icicles can be created in static and in dynamic modes. The Icicles can be grown on any surface, viewed in a curve mode for faster interaction with the user.

Dynamic formation of icicles are the most interesting part in the Icicle Generator tool. They are very flexible so the artist can achieve lots of effects from it.

6. CRAFTING TOOLS FOR ANIMATORS

6.1 WINTER FX SNOW ACCUMULATOR

The first tool that I have created is called as the winterFX snow Accumulator. The tool takes the advantage of Houdini's powerful procedurally based work flow. It basically gives the artist a lot more freedom to select the parts in the scene and then accumulate snow on top of it. The artist can select as much as objects that he wants and then the tool gives him an option of selecting parts from that particular object to accumulate snow.

It mainly asks for a snow direction and the spread angle to start the accumulation. In this tool we don't have to do any particle simulation to determine any of the collisions. In one click the accumulation is done and giving the user a lot more controls to the look.

Technically, In order to define the place of accumulation my tool works on the base of normal information of each and every primitive in the object.

The direction given by the user and the spread angle decides those primitives having similar normals.

It is completely interactive so we can go back and forth. The tool accepts any model from any 3D application as an .obj file. I have tested in models from Maya and XSI.

The interface is developed to be fairly simple for anyone to understand in the first sight.

The interface allows the user to navigate between the Selection Area – which actually shows the user the parts that he has selected to start the accumulation. He can also navigate through the points and Snow mesh.

The interface gives the artist controls like the amount of snow to be accumulated, The random seed can be changed in real time to see the automatically generated snow surface.

I have given two building method controls, the MetaBalls and ParticleFluidSurface. The same sort of result can be achieved by both the building methods, ParticleFluidSurface seems to work faster than the Metaballs. These controls are given mainly because of their interaction with the other objects. If the snow is built with MetaBalls then they can be fused easily with the other snow objects in the scene, but particleFluidSurface cannot really fuse with the neighbor snow object and does not give a good visual impact in this case. If as a separate object it is advisable to go for it because they are faster than MetaBalls. MetaBalls have to be converted to polygons in the final stage.

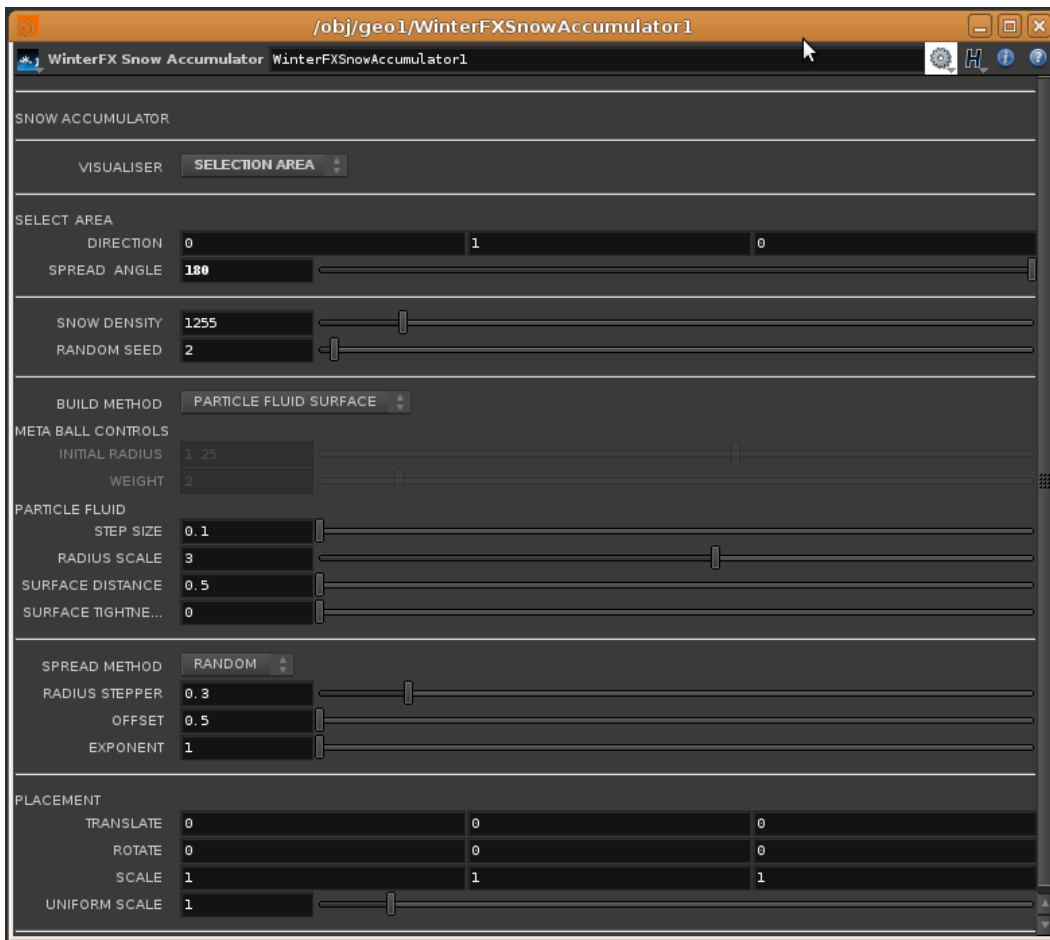


fig.4 WinterFX Snow Accumulator

There are two Spread Methods for the accumulation of snow. The first one is the uniform distribution function where the radius of every single MetaBall or Particle Fluid Surface point radius will appear to be the same, their results are good but if more details are needed then extra Displacement maps have to be plugged in at render time. The Random distribution function gives each and every MetaBall or Particle Fluid Surface point radius to be randomly generated so the snow appears with more details in the view port itself.

The random distribution function has controls like the Radius Stepper, Offset and the Exponent to vary the generated mesh. The controls are all very interactive.

6.1.1 TECHNICAL ASPECTS

The whole generation is done by using Houdini's powerful node based work flow.



fig 5 Snow Accumulator procedural work flow

The given geometry comes along with normal information so we have to query the normal for each and every primitive in the object selected. Using that information the user can set his direction of accumulation in xyz. When the individual primitives are selected then for those particular primitives we can scatter points using the scatter SOP in Houdini. MetaBalls or Particle Fluids can be replaced onto those point positions to generate the mesh. The random distribution for these builds cannot be directly applied so, the system generates color for each and every point and based on that color value the radius is also randomly taken and scattered on top of the points.

6.1.2 ALGORITHM

Begin

Read in the User geometry input.

If (Visualiser == SelectionArea) **Then**

View the input geometry

Read the values of the Direction and Spread Angle attributes.

Select Primitives of Similar Normal Directions and Spread angles.

Delete the remaining primitives in the geometry

Show only selected Primitives.

Else if (Visualiser == Points) **Then**

Read the values of the Direction and Spread Angle attributes.

Select Primitives of Similar Normal Directions and Spread angles.

Delete the remaining primitives in the geometry.

Read the values in Snow Density and the Random Seed.

Scatter Points randomly on the selected Primitives based on Snow Density
Random Seed.

Turn on Display Points in the view port to visualize the points.

Else if (Visualiser ==Snow Mesh) Then

View the input geometry

Read the values of the Direction and Spread Angle attributes.

Select Primitives of Similar Normal Directions and Spread angles.

Delete the remaining primitives in the geometry

Scatter Points randomly on the selected Primitives based on Snow Density
Random Seed.

If (Build Method == ParticleFluidSurface)

If (Spread Method == Uniform)

Read the values in Step Size,Radius Scale,Surface
Distance, Surface Tension Attributes.

Create ParticleFluidSurface and generate mesh.

Else If (SpreadMethod == Random)

Create Color Values for each and every point based
On point numbers.

Create Radius Stepper, Offset and Exponent
Variables.

Radius =

$((\$CR+ch("offset"))*ch("radius"))^{ch("exponent")}$

Create ParticleFluidSurface and generate mesh.

Else If (Build Method == Meta Balls)

If (SpreadMethod == Uniform)

Read the values in Initial Radius and Weight

Attributes

Create MetaBall Surface

Else If (SpreadMethod == Random)

Create Color Values for each and every point based
On point numbers.

Create Radius Stepper, Offset and Exponent
Variables.

Radius =

$((\$CR + \text{ch}(\text{"offset"})) * \text{ch}(\text{"radius"}))^{\text{ch}(\text{"exponent"})}$

Create MetaBall Surface.

End

6.1.3 IMAGES

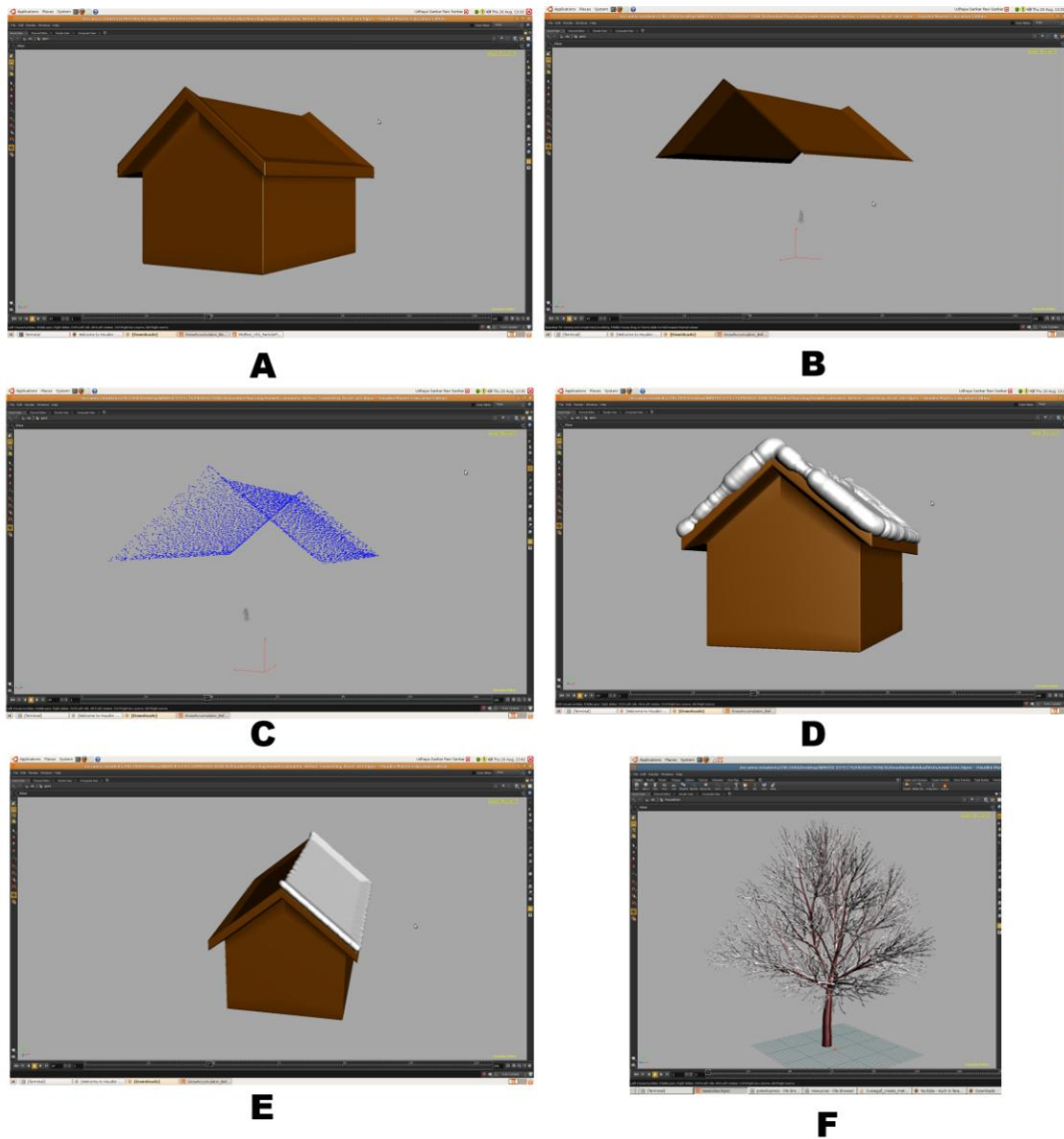


fig 6 Screen grabs from Houdini

- A VIEW MODEL (SELECTION AREA)
- B SELECTED AREA
- C VIEW POINTS
- D SNOW WITH RANDOM SPREAD (just random values entered)
- E SNOW WITH XY DIRCTION AND UNIFORM SPREAD
- F TREE WITH UNIFORM SPREAD

6.2 ICICLE GENERATOR

The second tool that I created for the winterFX is the Icicle generator. The basic design was to create icicles and get them working according to the Icicles from the company Snow Business. Icicles are really complicated in the way they are formed. They are actually formed due to the melting of ice on roof tops and gets sharp in their ends. Icicles can also be formed on tree tops or even near water falls.

They are really dangerous in a way that they might fall down and hurt some one under it because they have a huge mass and very sharp edge.

According to Physicist Martin Short “Icicles have a certain mathematical shape, and this mathematical shape is universal among icicles. He also adds up that the height is proportional to the radius to the four thirds”.(Physicist Martin Short)

I have designed my icicle generator keeping in mind these information, but then again for a CG environment it is more than stylization than realism. So the controls have to be really flexible to create any sort of shape or any styles to it.

They also come in two modes static and dynamic. Static dynamics are created in one click and have lots of controls to get the look that the user is looking for, Dynamic Icicles are really interesting they can start at any given frame and given a life variance to sort of make them stop elongating. The shape has full control till the end.

The UI consists of all the necessary controls needed. We can create this effect on any object. Icicles can take normal information from point attributes and decide the direction for the icicles as well.

Icicles can be created in the edges of an object which are flat and open and also on the surface of the given geometry.



fig.7 Icicles Generator UI

The Icicle generator comes with a compact UI. The user can build the effect from scratch. We have control on Edge or surface static, dynamic and all other visualization Parameters.

The dynamic icicles are basically created from Particle dynamics within Houdini. It has an inbuilt noise function with which the user can use it not only for icicles but also other effects like Random Tentacles growth on an object and stuff.

6.2.1 TECHNICAL ASPECTS

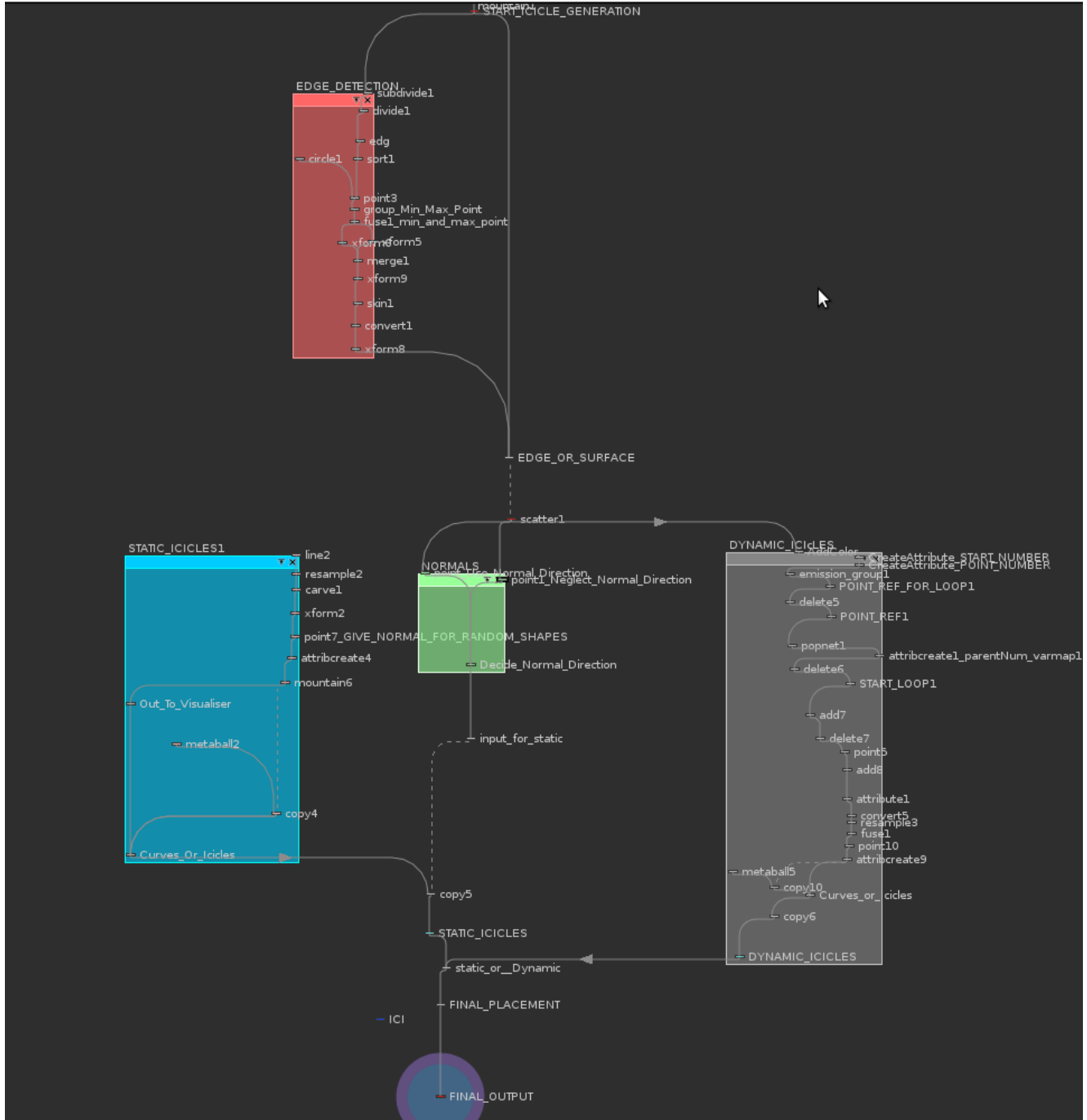


fig.8 Icicles Generator Procedural work flow

6.2.2 ALGORITHM

Begin

Read Geometry input

If (Generate On == Surface) **Then**

If(Generation Type == Static)

Read Values from Number of icicles and random seed attributes

For (0,Number of icicles given by the user,++)

If(Visualiser == curves) **Then**

If(Direction==Use Normal Direction) **Then**

Append Line SOP and make it a nurbs. Set the origin to (0,0,0). Make the direction to (0,-1,0),Distance 1 and number of points in the line = 10 and of a order 4.

Append Resample SOP and read in the value of Points Spacing from the user.

Append Carve SOP and in the SecondU attribute.use an Expression:

Stamp("../copyDel", "size", 0) * ch("Static_random_length")

It means. Generate a random number from a stamping Variable in the copyDel SOP and multiply with the user Given Random Length Step for giving random lengths to all the icicles specified in Number of Icicles.

Transform pivot to the Y max by usig the Expression bbox("../carve1",D_YMAX). For the Y Attribute and \$CEX,\$CEZ for the X and Z axis.

Create color value for all the points generated in the line.

Use the expression:

```
(1-($PT/$NPT)) * fit01((rand(stamp("../copy5","siz",0))),  
ch("Static_Min_Radius"),ch("Static_Max_Radius"))
```

This creates Color Values on the points of the line ranging From Red to Black.

Create random normal directions for all points. Use the

Expression: fit01(stamp("../copyDel","siz",0),-1,1)

This gives a random value ranging from -1 to 1.

Create pscale attribute for controlling the radius of Icicles

From top to bottom. The pscale attribute will be based on the color that we have created in the previous expression.

The final pscale value will be.

```
(($CR+ch("offset"))*ch("radius"))^ch("exponent")
```

The radius,offset and exponent will be getting values from Radius Stepper, Offset,Exponent.which is under the shape Refinement options.

Append a mountain SOP to give random shape variation to the curves generated.

Append a point SOP and give \$NX , \$NY , \$NZ in the use Normal tab

Else If(Direction==Neglect Normal Direction) **Then**

Append a point SOP and make the normals (0,0,0)

Show the generated curves copied on the scattered points.

If else(Visualiser == IcicleMesh)

// same algorithm has to be followed as above//

Follow the same procedure as the Visualiser==Curves Condition. And instead of showing curves in the Last, **Append** a Metaball SOP and copy it to the Curves. So on Each and every point on the Curve we have a meta Ball Scaled from Max to Min radius values.

else If(Generation Type == Dynamic)

If(Visualiser == curves) Then

Create a color attribute for all the points in the R channel.

Use the expression \$PT/\$NPT, so colors will be assigned based on Point numbers because \$PT = pointNumber.

Create a START_NUMBER Attribute.

Assign it the value:

```
Int(fit01(rand($PT),ch("birth_start"),ch("Birth_End")))
```

Basically we are generating. A number for each and every point given by the user, to decide in which frame each Icicle should start.

The Birth Start and Birth End will give the Minimum and Maximum frame in which they have to be generated.

Create a POINT_NUMBER Attribute. Using a point sop and the variable \$PT.

Append a Group SOP and give an expression in the point tab and

Choose Filter by Expression.

$\$STARTNUMBER == \F . So points having $START_NUMBER$ equal to the current frame number they will get grouped.

Append a delete SOP and keep only the selected Points by the group above.

Append a POP Network SOP and use the selected points as a Particle.

Append a wind POP and drag to control the particle Direction
The user can give controls from the Noise and Wind Attributes
From the UI to control these nodes.

Append a split POP and emit particles from the initial particles.

Append a speed limit SOP and restrict the behaviour of the Secondary particles.

The length of each curve elongating can be decided by the initial Particle's Life Span given by the user Given from the UI(Maximum Life and Variance parameters)

Append an Add Sop and create a curve out of the particles.

Append a resample POP and give more number of points.

So the curves which are formed from the points start elongating and finally stop when the Initial Particle Dies.

Show the curves dynamically growing.

Else If(Visualiser == Icicles) **Then**

Follow the same procedure as mentioned above to create the Curves and then

Append a meatball and copy SOP with a color attributr for all the points

$((\$CR+ch("offset"))*ch("radius"))^ch("exponent")$

Where the offset, Radius Stepper and the exponent come from The UI.

So the meatballs with the curves of points and a color Attribute will give the Icicles an Elongating Structure. The controls for Direction : Set Initial Velocity, Inherit Velocity and add to inherit velocity can be directly piped Into the POP network by the UI and the User's Values

Else If(Generate On== Edge)

Follow the same Procedure for Creating the Icicles But for the Except The Points are scattered on the edge. For that I have implemented my

Edge Detection Algorithm:

Read in the geometry input.

Append a SubDivide SOP to give more resolution to the geometry.

The User can control the level of detail by using the UI control, Edge Tolerance.

Append a Divide SOP and click on Remove Shared Edges.

Append a Group SOP and Group all the Points and give them a unique Name called EDGE_SELECTED.

Append a SORT SOP and sort the point numbers by the vertex order.

Append a circle SOP and Make its points sit on the Grouped Points.

Append a group SOP again to determine the starting and ending point By using the point numbers.

Turn on Group by range and type in the following expression.

START/END = 0 , npoints("../sort")-1

SELECT_OF = 1 , npoints("../sort")-1

Group the starting and ending point and **FUSE** them by using a **FUSE** SOP.

Append a transform SOP and Create one more copy of the circle that we

Have created. Scale it down a little bit so that it would help to create a Surface out of it.

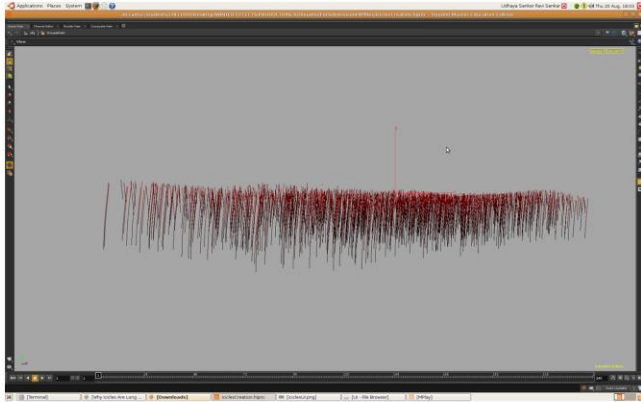
From the resulting Surface we can scatter points and then create our Icicles and curves according to them.

End For

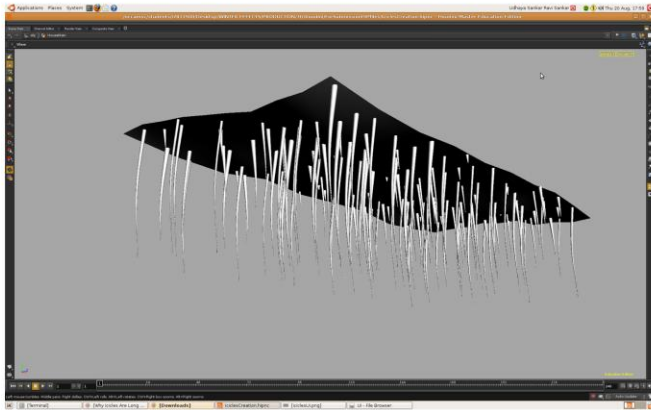
End

6.2.3 IMAGES

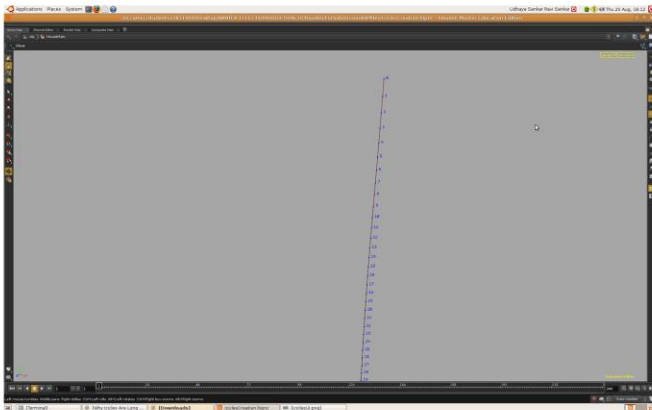
- A) Curves generated for Icicles. The color of the curves range from red to black. Metaballs take those color information and set their radius.
- B) The Icicles formed due to those curves and taken their color information for setting their radius
- C) Curve made up of points and the order in which points are placed.
- D) Sample render with the shader developed
- E) Icicles hanging from top.
- F) The particle network used to generate the Initial and secondary particles.



A



B

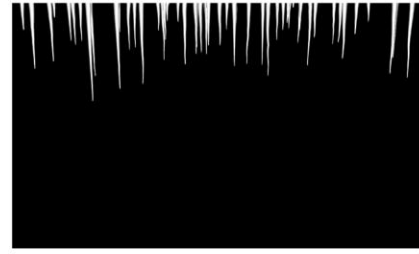


C

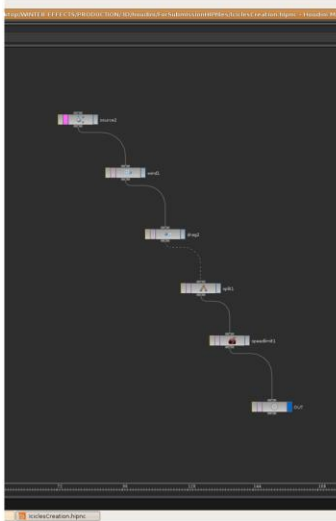
fig.9 Icicles Generator Functions



D



E



F

fig.10 Icicles Generator Functions

6.3 SNOW BY TEXTURES CREATOR

My third tool to create the Accumulation of snow is called as the snow by textures creator. After having a short meeting with a few artists like Dennis and Shan from the MA 3D NCCA, They said a tool on which that they can give just a texture file to generate the snow accumulation effect. It is actually a very handy tool to be used in the production pipeline. It is very similar to creating displacement maps or bump maps. The artist can just unwrap a model that he wants to be accumulated with snow and it just does it very quickly, It basically takes in an apha channel from the snow map that the artist provides and just apply it on the model to get the snow accumulated very fast.

If the user likes to just a project textures on top of the model in a particular axis even that is possible with this tool. Simple objects don't even need to be unwrapped. UV projections can be used to project a texture and get the accumulation done very quickly. I have mainly made use of the VOPS and SHOPS networks in Houdini for the final algorithm and execution. I have referred from odforce.net, understood it well and changed it according to my necessity.

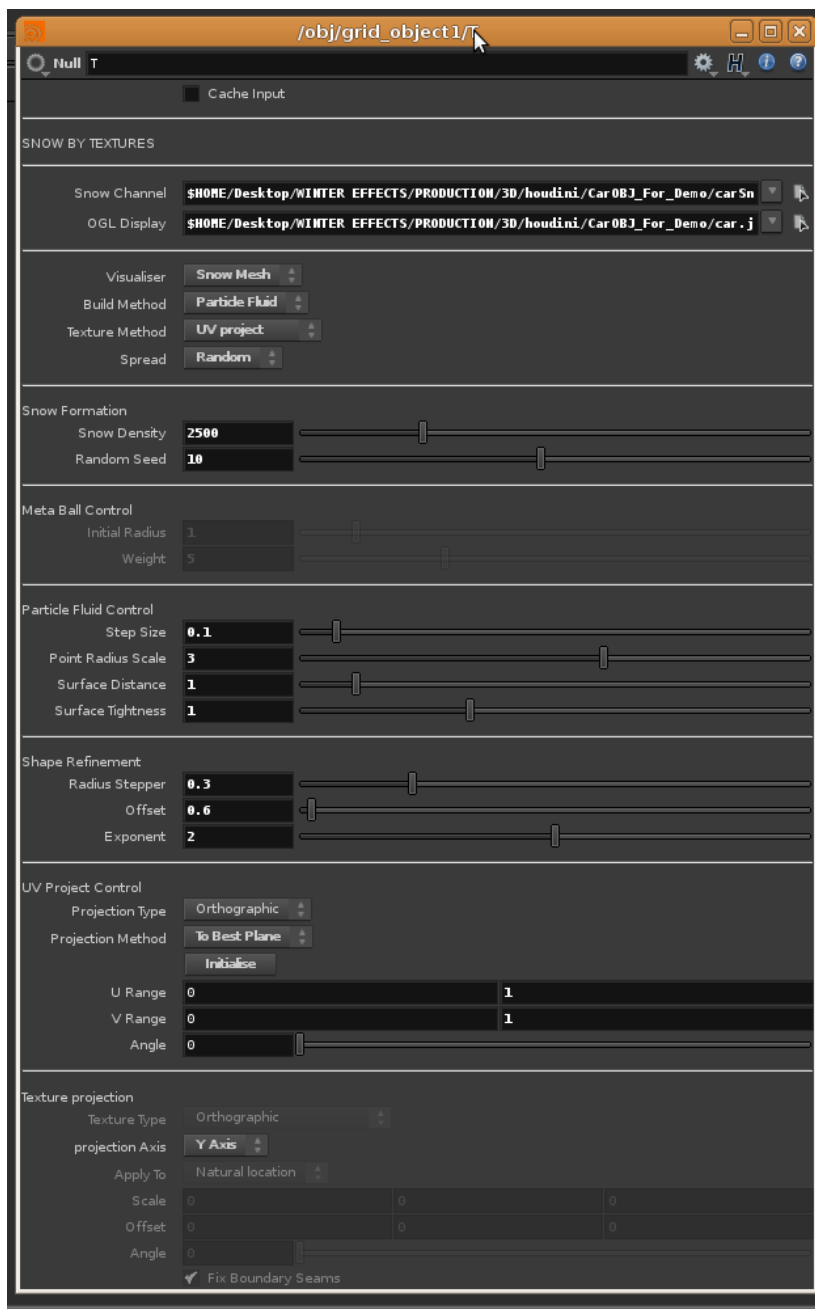


fig 11 Snow by Textures UI

The Snow by textures comes with a very compact UI. It actually demands for the Snow channel in the first place and OGL Display for the user to view in the viewport. Just plugin the textures and the results are automatically generated. Similar to the other tools

in this winterFX bundle, this tool has the visualizing options and both building methods like Metaballs and particleFluid surfaces.

6.3.1 TECHNICAL ASPECTS

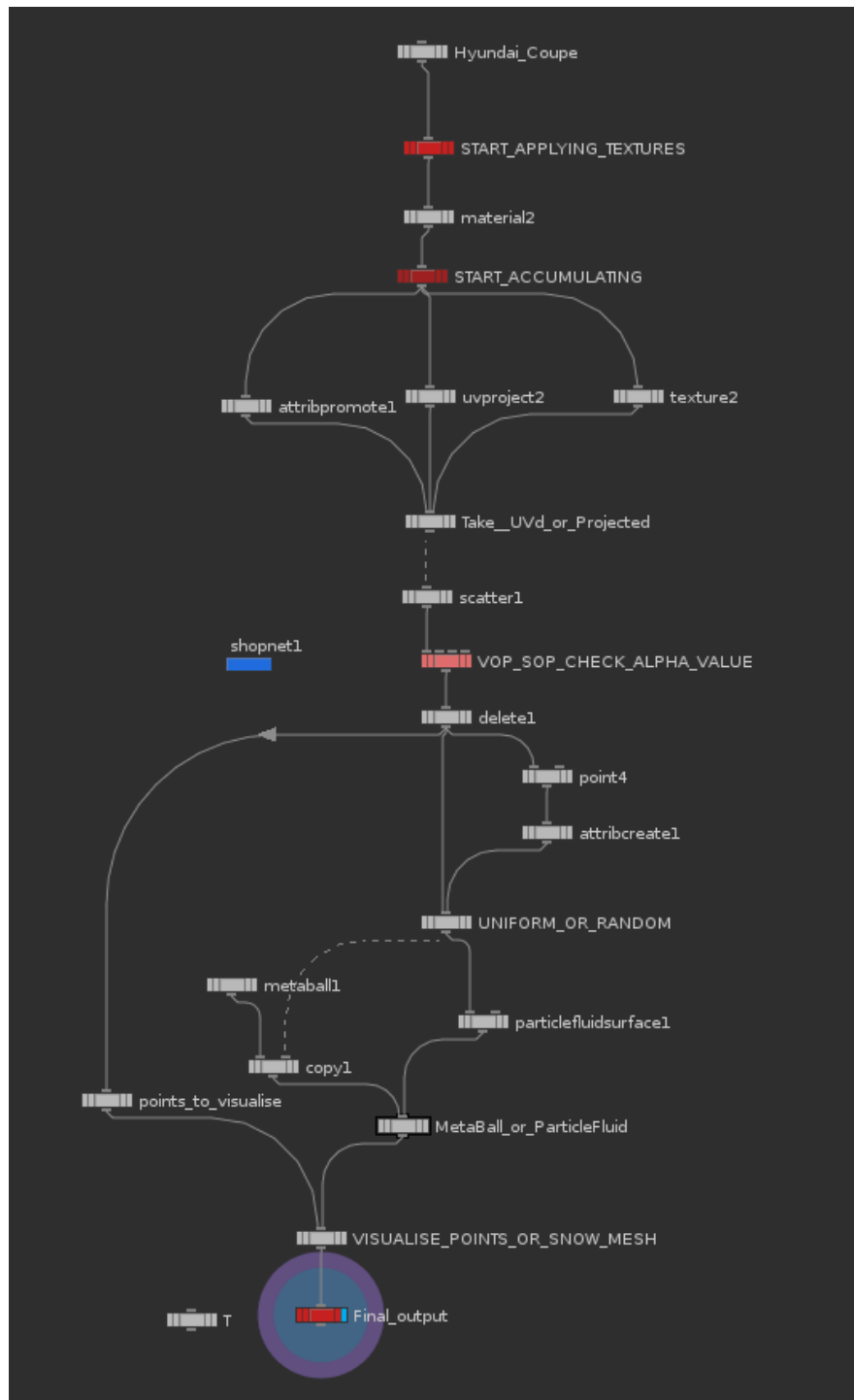


fig 12 Snow by Textures procedural work flow

6.3.2 ALGORITHM

Begin

If (Visualiser==Points) **Then**

If (Texture Method==UV unwrap) **Then**

Read Information from the geometry input.

Append a Material SOP and in the material Attributes choose the path of the VOP surface in the SHOP Network. So Geometry will be assigned a texture. It basically takes the UV information from the input geometry and combines it with the given texture to finally apply it on the model. (referred from odforce.net)

Append an attribute promote SOP and change the original UV information from Vertex class to the point class and change the Promotion method to average . Only Projections directly with point informations.

Append a scatter SOP and the Snow Density parameter from the UI controls the Number of points and the seed value as well.

Append a VOP SOP to Calculate the alpha values. It basically takes in the texture and the UV information of the object. It calculates the alpha channel from the given texture and send it out for decision making.

Append a Delete SOP and filter out the points whose alpha values are found to be zero.

Send the Remaining points for visualization.

Else If(TextureMethod == UV projection)

Read Information from the geometry input.

Append a Material SOP and in the material Attributes choose the path of the VOP surface in the SHOP Network. So Geometry will be assigned a texture. It basically takes the UV information from the input geometry and combines it with the given texture to finally apply it on the model.

Append a uvproject SOP and take in inputs from the user for Projection Type and projection Method. Take also information from U range V Range and angle

Append a scatter SOP and the Snow Density parameter from the UI controls the Number of points and the seed value as well.

Append a VOP SOP to Calculate the alpha values. It basically takes in the texture and the UV information of the object. It calculates the alpha channel from the given texture and send it out for decision making.

Append a Delete SOP and filter out the points whose alpha values are found to be zero.

Send the Remaining points for visualization.

Else If(TextureMethod == Texture projection)

Read Information from the geometry input.

Append a Material SOP and in the material Attributes choose the path of the VOP surface in the SHOP Network. So Geometry will be assigned a

texture. It basically takes the UV information from the input geometry and combines it with the given texture to finally apply it on the model.

Append a texture SOP and get in the values from the user for Texture Type, Projection Axis, Scale, offset and angle.

Append a scatter SOP and the Snow Density parameter from the UI controls the Number of points and the seed value as well.

Append a VOP SOP to Calculate the alpha values. It basically takes in the texture and the UV information of the object. It calculates the alpha channel from the given texture and send it out for decision making.

Append a Delete SOP and filter out the points whose alpha values are found to be zero.

Send the Remaining points for visualization.

Else If (Visualiser==Snow Mesh) **Then**

Follow the same procedure till the scattering and filtering out only the Points of alpha more than zero.

If (Build Method==Particle fluid)**Then**

If (Spread==Uniform)

Append ParticleFluidSurface SOP and build mesh

Show Snow

Else If(Spread==Random)

Append Point and attribute Create SOP to assign color Variables to the points. In this case we are creating red Color based on the point number.This gives the pscale

Variable for the snow building.

The Final radius of the snow created will be,

$((\$CR+ch("offset"))*ch("radius"))^ch("exponent")$

The radius stepper,offset and the exponent will be given by

The user.

Else if (Build Method==Metaballs)Then

If (Spread==Uniform)

Append Metaball SOP and build mesh

Show Snow

Else If(Spread==Random)

Append Point and attribute Create SOP to assign color

Variables to the points. In this case we are creating red

Color based on the point number.This gives the pscale

Variable for the snow building.

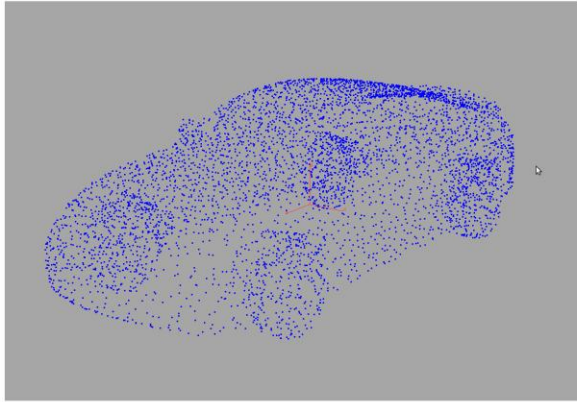
The Final radius of the snow created will be,

$((\$CR+ch("offset"))*ch("radius"))^ch("exponent")$

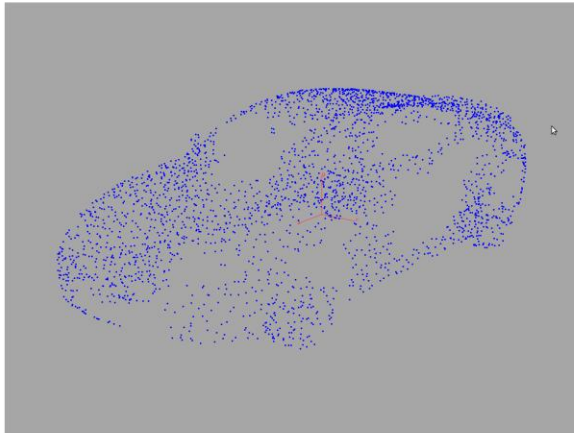
End

6.3.3 IMAGES

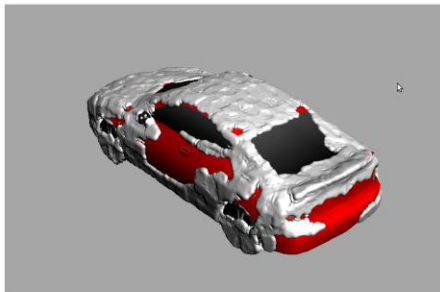
- A) Points created on an object using the scatter tool.
- B) Points deleted by the VOP SOP and alpha value queries.
- C) Snow Accumulated on the car object.
- D) Showing all the details in snow.



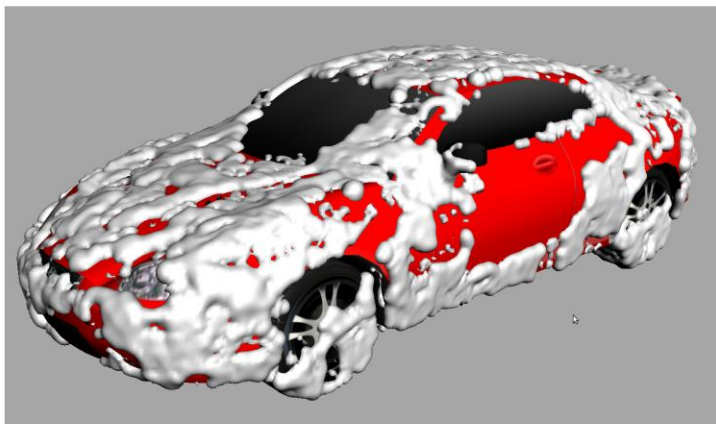
A



B



C



D

fig 13 stages of accumulating snow on cars

VOP SOP for calculating alpha

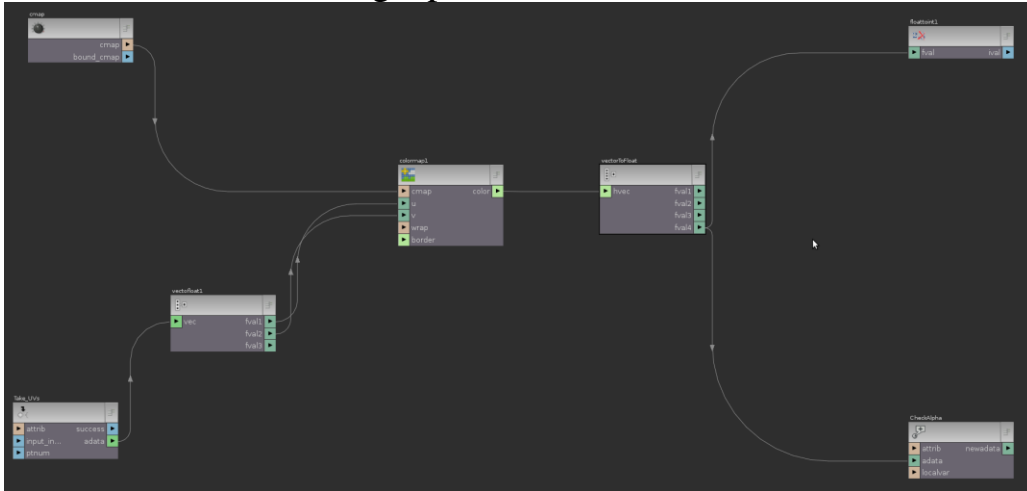


fig 14 VOP SOP Network(referred from odforce.net)

SHOP network

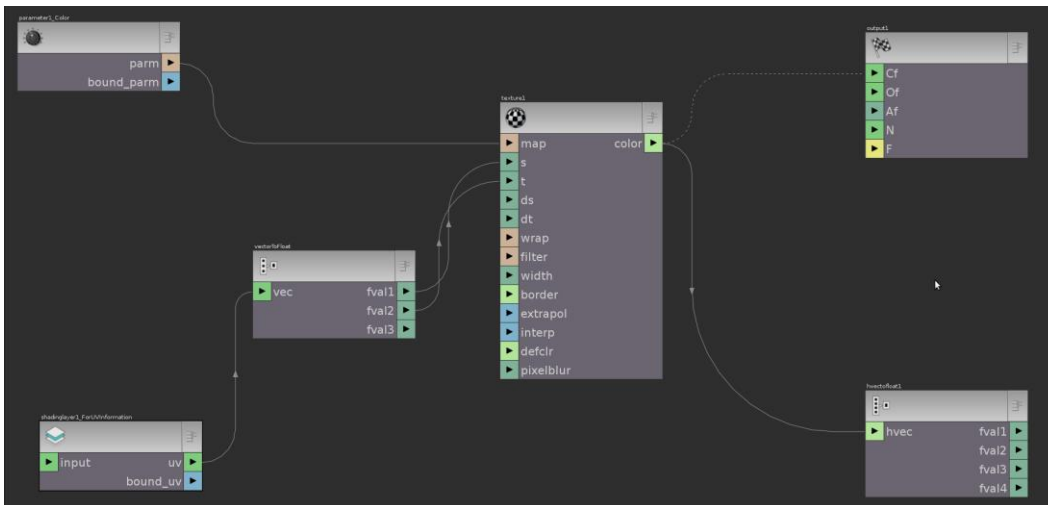


fig 15 SHOP Network

UV and Snow Map

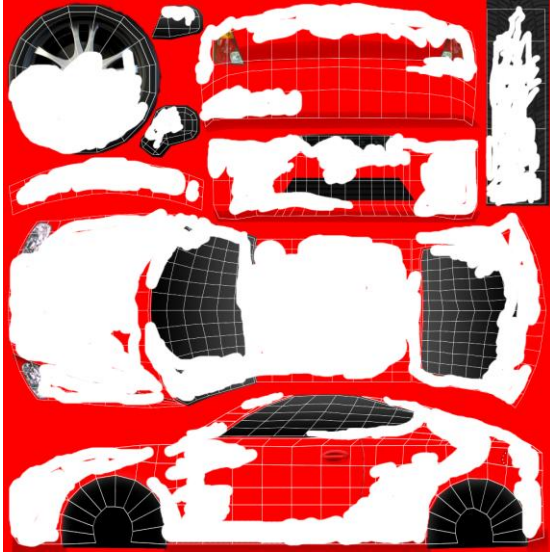


fig 16 UV and SNOW Map

6.4 Painting on Surfaces To Create Snow

The fourth way of creating the winterFX is the Painting on Surfaces to create snow accumulation. This tool is really handy when it comes to controlling more details. The user can interactively select any primitive and start painting directly on top of it. We have two options in this tool one is painting points based on the area of primitives(referred from sidefx.com/tutorilas) and the other one is the paint by point color. Both of these have their advantages in different situations but their results are all similar. One advantage with the point color method is that we can also delete certain unwanted particles away. The disadvantage is that we cannot add painting option in an digital Asset very efficiently so this particular tool can serve as a separate function for performing the necessary operations. It will really be a very helpful tool for quick modifications.

The user interface has various controls, Which will be explained below as follows.

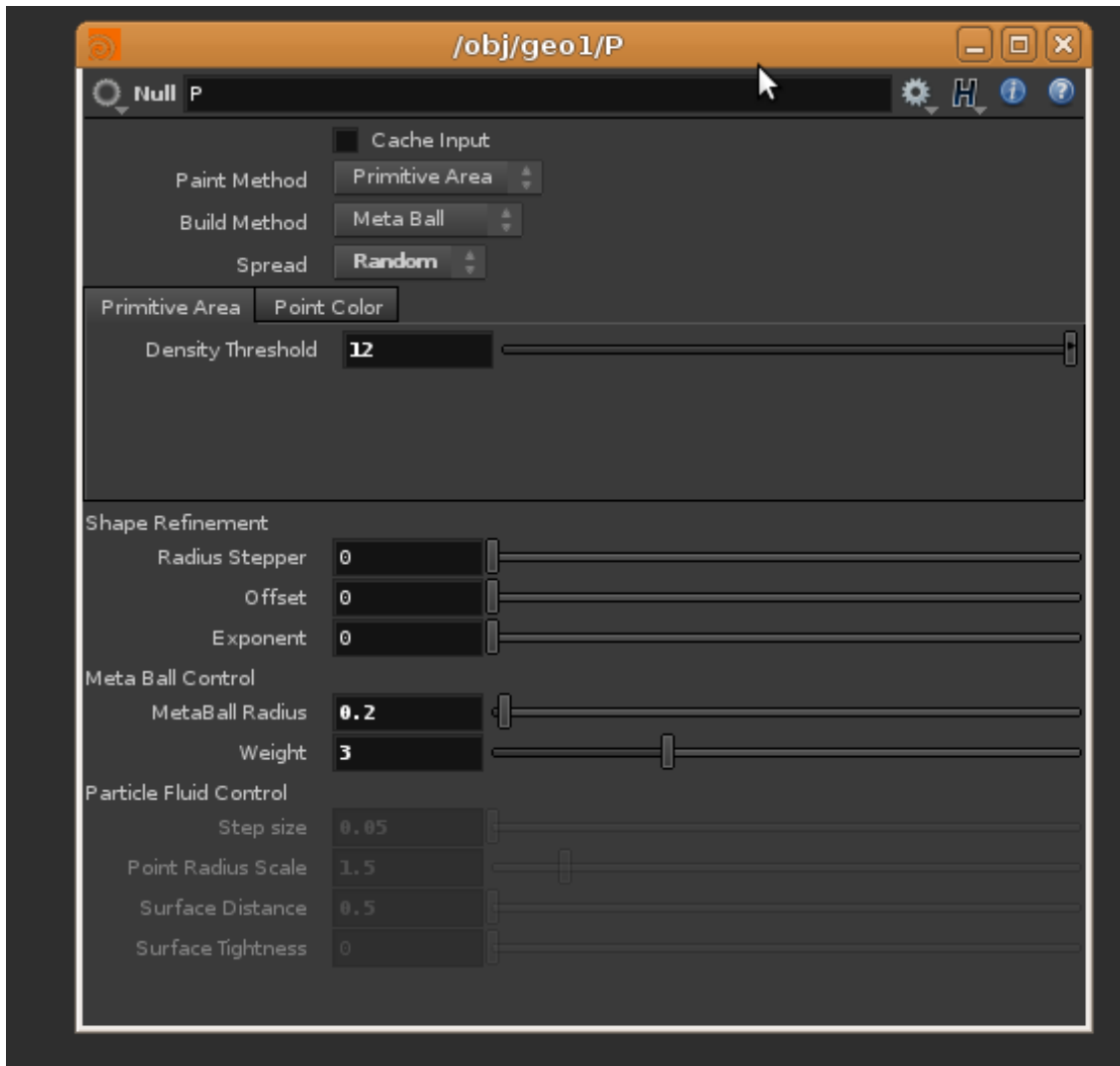


fig 17. painting tool user interface

6.4.1 TECHNICAL ASPECTS

The network of the paint tool is shown as follows.

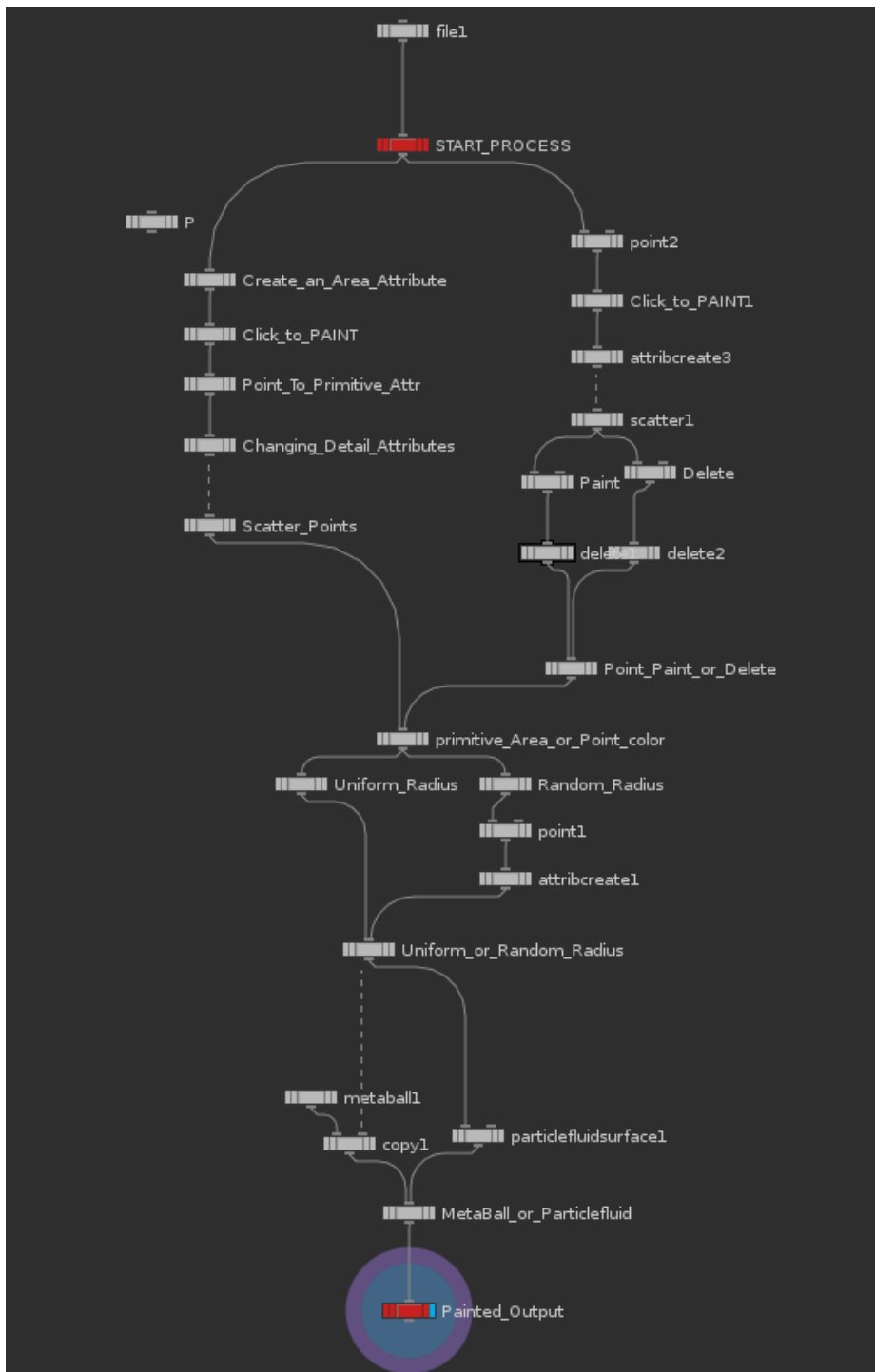


fig 18. painting tool procedural workflow

6.4.2 ALGORITHM

Begin

If (paintMethod == PrimitiveArea) **then**

If(Spread==Uniform) **then**

Create an attribute called area by using the Attribute Create SOP

Append a paint SOP and make it override the created area attribute.

Change the original Class from point to primitive Class.

Change the original Primitive Class to a Detail Class and give it a new name called as New_Area. The promotion method must be in sum

Append a scatter SOP and In the Number of points option give this expression.

detail(“../Changing_Detail_Attributes”,”Sum_of_Area”,0) *

ch(“Density_Threshold”)

Scatter should be based on primitive area and in the alternate attribute type in area because it wont take it automatically.

If(Build Method == MetaBall)

Append copy SOP and a meatball to build the surface.

If Else(Build Method ==PartcileFluidSurface)

Append Particle fluid Surface SOP and generate the mesh.

Else If(Spread==Random)

Create an attribute called area by using the Attribute Create SOP

Append a paint SOP and make it override the created area attribute.

Change the original Class from point to primitive Class.

Change the original Primitive Class to a Detail Class and give it a new name called as New_Area. The promotion method must be in sum

Append a scatter SOP and In the Number of points option give this expression.

**detail(“./Changing_Detail_Attributes”,”Sum_of_Area”,0) *
ch(“Density_Threshold”)**

Scatter should be based on primitive area and in the alternate attribute type in area because it wont take it automatically.

Append a point SOP and create a color attribute for the red channel

Expression is \$PT/\$NPT

Create an attribute called pscale and let it have the value

$$Pscale = ((\$CR+ch(“offset”))*ch(“radius”)^ch(“exponent”))$$

If(Build Method == MetaBall)

Append copy SOP and a meatball to build the surface and it will

Take the pscale automatically.

If Else(Build Method ==PartcileFluidSurface)

Append Particle fluid Surface SOP and generate the mesh.

The pscale has to be given as an attribute and it takes it for the Radius.

Else If (paintMethod == Point Color) **then**

If(Spread==Uniform)then

Append a point SOP and give the points in the geometry a black color

So I have (0,0,0) in a ll the color channels.

Append a paint SOP to it and stat painting white color so have (1,1,1) in the FG color.

Append a attribute Create SOP and give it a value of \$CR.

Append a Scatter SOP and the number of points come from the UI

If(Point paint mode==Paint)

Append a Group SOP and fit in a new attribute called as paint tolerance

Enable group by number and give an expression

$$\$MASK < ch(“Paint_Tolerance”)$$

and the value comes from the UI.

Else If(Point paint mode==Delete)

Append a Group SOP and fit in a new attribute called as Delete tolerance

Enable group by number and give an expression

\$MASK > ch("Delete_Tolerance")

and the value comes from the UI.

If(Build Method == MetaBall)

Append copy SOP and a meatball to build the surface.

If Else(Build Method ==PartcileFluidSurface)

Append Particle fluid Surface SOP and generate the mesh.

Else If(Spread==Random)**then**

Append a point SOP and give the points in the geometry a black color

So I have (0,0,0) in all the color channels.

Append a paint SOP to it and start painting white color so have (1,1,1) in the FG color.

Append a attribute Create SOP and give it a value of \$CR.

Append a Scatter SOP and the number of points come from the UI

If(Point paint mode==Paint)

Append a Group SOP and fit in a new attribute called as paint tolerance

Enable group by number and give an expression

\$MASK < ch("Paint_Tolerance")

and the value comes from the UI.

Else If(Point paint mode==Delete)

Append a Group SOP and fit in a new attribute called as Delete tolerance

Enable group by number and give an expression

\$MASK > ch("Delete_Tolerance")

and the value comes from the UI.

Append a point SOP and create a color attribute for the red channel

Expression is $\$PT/\NPT

Create an attribute called pscale and let it have the value

$Pscale = ((\$CR+ch("offset"))*ch("radius"))^ch("exponent")$

If(Build Method == MetaBall)

Append copy SOP and a meatball to build the surface and it will

Take the pscale automatically.

If Else(Build Method ==PartcileFluidSurface)

Append Particle fluid Surface SOP and generate the mesh.

The pscale has to be given as an attribute and it takes it for the

Radius.

End

6.4.3 IMAGES

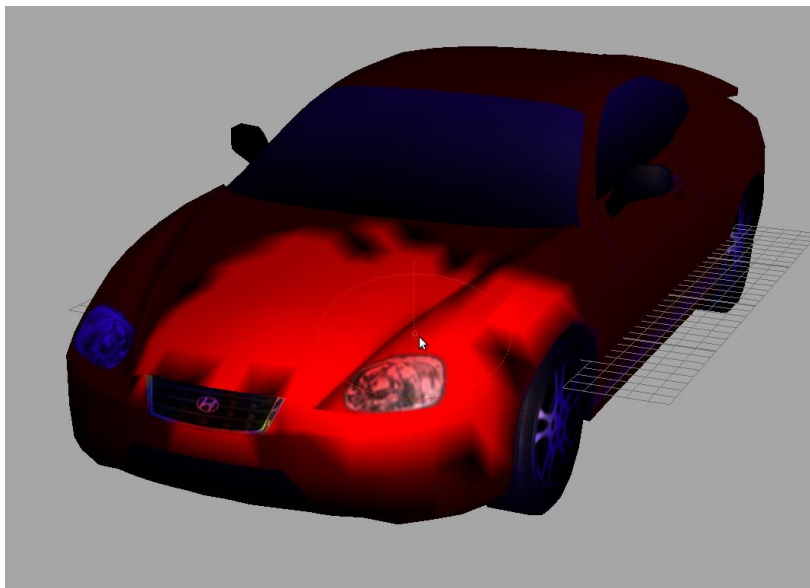


fig 19. paint mode

The Red area is what the user has painted. So the accumulation should calculate there.

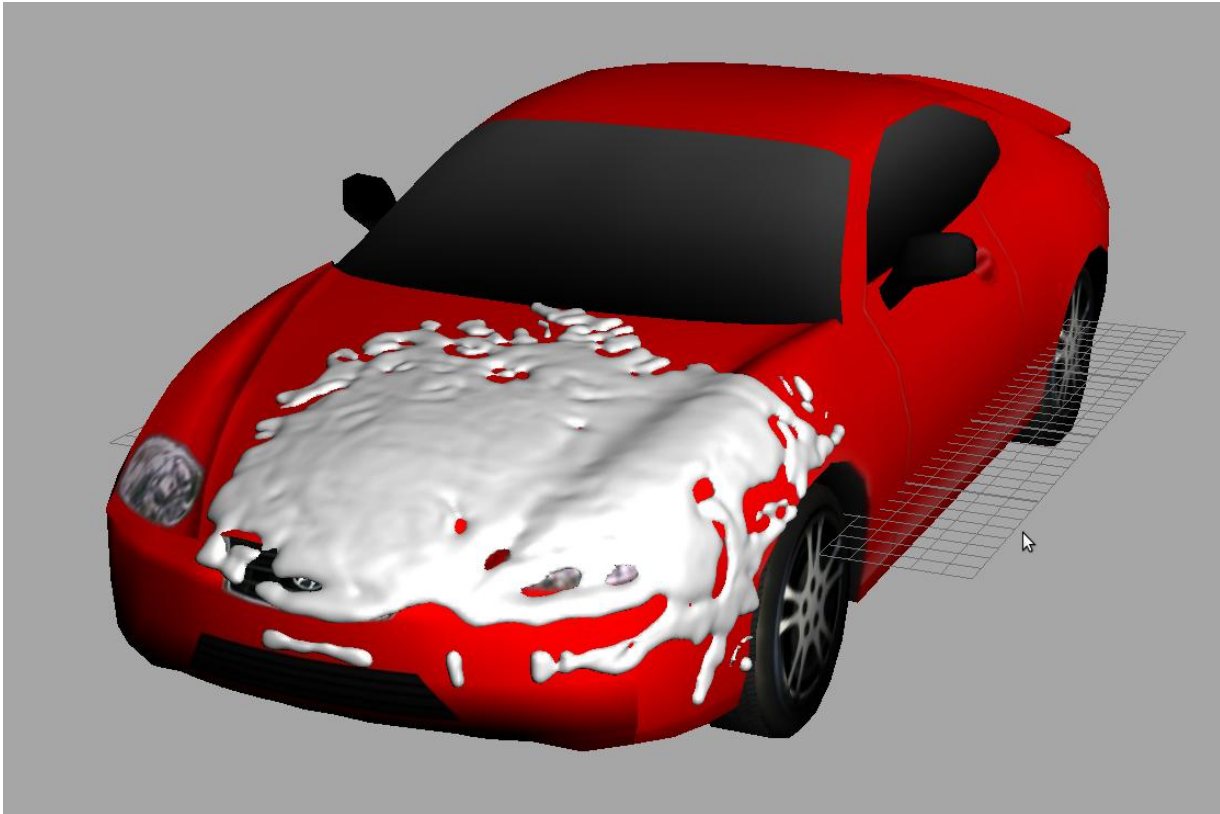


fig 19.The final accumulation of snow due to painting.

7. SHADING

For the accumulated snow I have designed my own shaders to demonstrate the snow feel. Below shown are screen grabs in the development of the shader. Fig A is the VEX Builder Surface Shader for the snow. It takes in the lighting information from the scene and calculates the shaders. It requires normalized normals to do the calculation so I append a normalized normalize operator to do that. I am creating the specular pass using a anti-aliasing noise. The bump mapping for the shaders are also driven through the anti-aliasing noise.

The VEX BUILDER surface shader

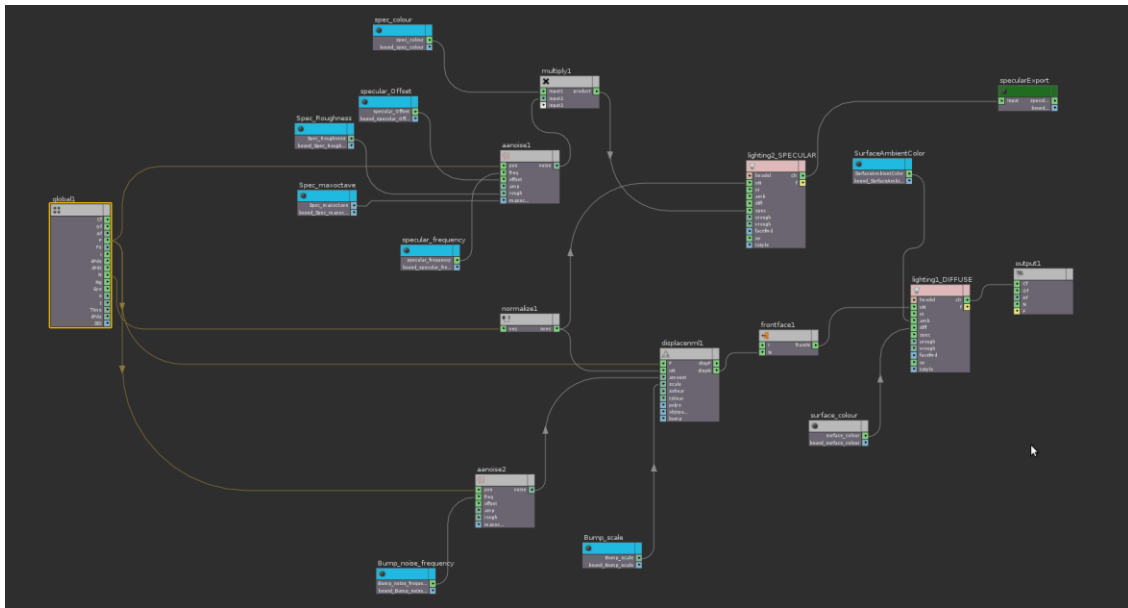


fig 19.The VEX BUILDER surface shader

The VEX BUILDER displacement shader

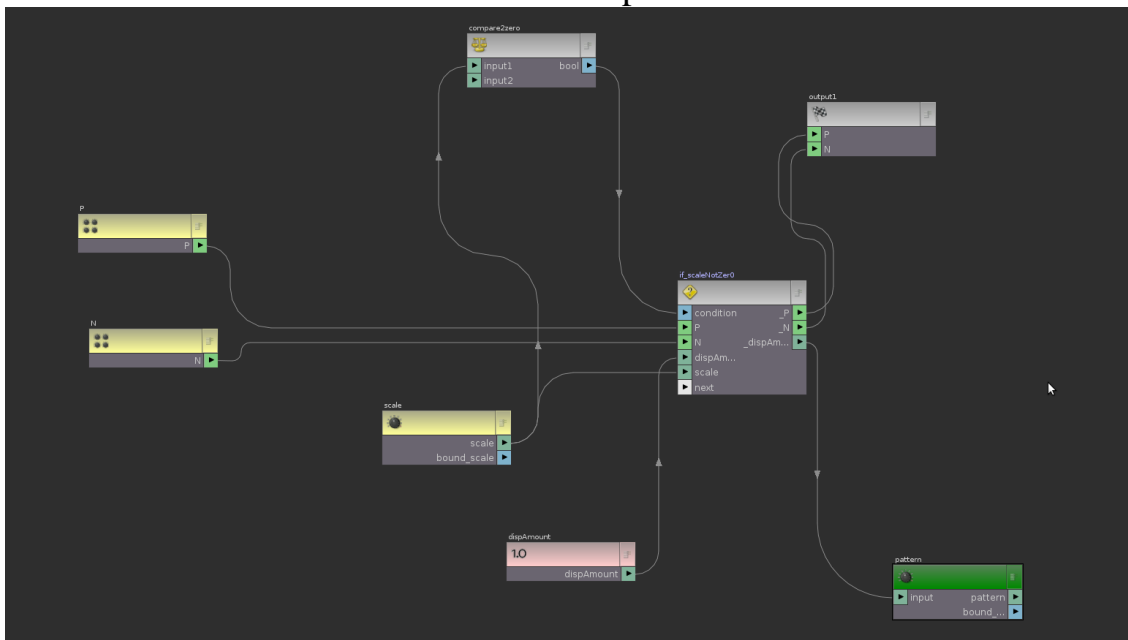
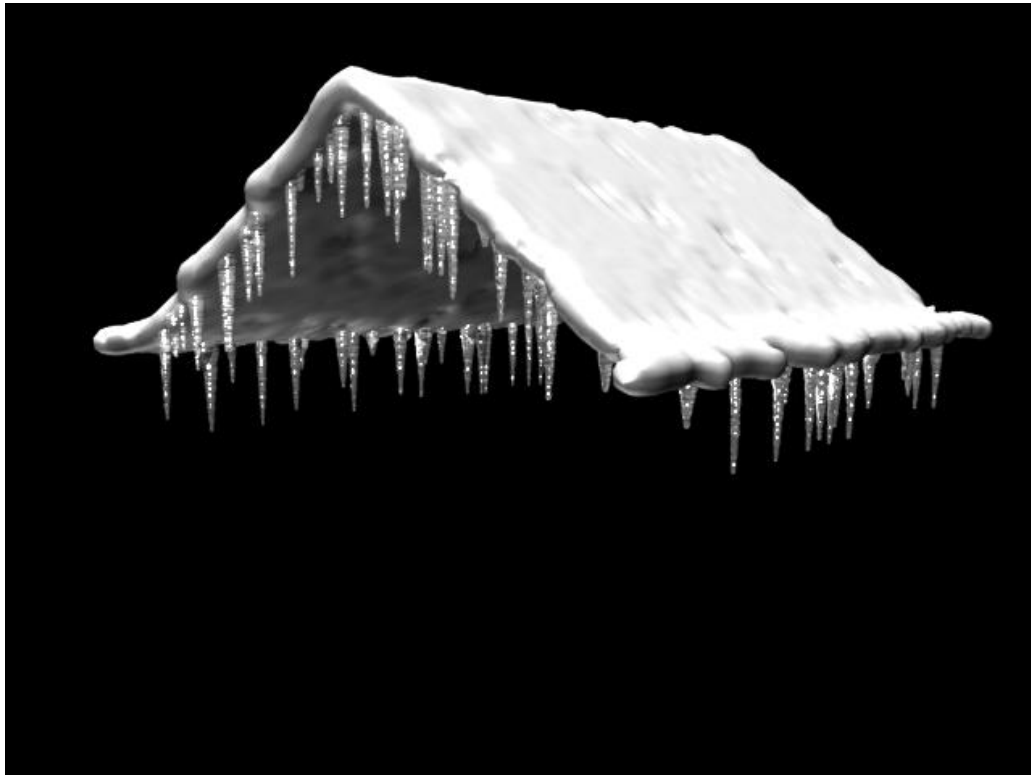


fig 20.The VEX BUILDER displacement shader

8. RENDERING

After generating the snow and icicles they can be rendered using the shader created or the user can give his own shader. I did the rendering thorough Mantra. For the icicles I have used the houdini's default ice cube shader and modified it very much in terms of the displacement, specular, reflection according to the look I needed.

The rendering was performed just by using a basic three point lighting. Mainly to show the accumulation of snow and the formation of icicles I didn't have in any background.



The icicles has got ray tracing on, There are no render passes in this image.If we can render the speculars, ambient occlusion and reflection in different passes we can achieve good results using this technique.

9. CONCLUSION

The Winter Effects tool can be used for small objects to huge environments if the shots and assets are broken down in very proper way. This gives the artist lot more controls to get the effect generated automatically or even if the artist wants to go artistically, this tool can be of great help. For painting on the surfaces function, currently Houdini does not support the Paint SOP for including in a digital asset. I have given a good network for the artist to understand it really well and gain good results out of it. I tried different ways to get it working some how, but resulted in very time consuming calculations. So it is just included as a separate function for the time being.

10. REFERENCES

- [1] Fearing, P, *Computer Modelling of Fallen Snow*, Proceedings of SIGGRAPH 2000
- [2] Nishita T,Iwasaki H,Dobashi Y, Nakamae E., *A modeling and Rendering method for snow by using meatballs.*
- [3] Moeslund T.B, Madsen C.B, Aagaard .M, Lerche D. *Modelling Falling and Accumulating snow*
- [4] Martin Short “*Why Icicles are long and thin*” Mathematical physics explain how Icicles grow
available at http://www.sciencedaily.com/videos/2007/0202-why_icicles_are_long_and_thin.htm
- [5] Snow Business, United Kingdom.
<http://www.snowbusiness.com/gallery/stills.aspx?id=1223&parent=1151>
- [6] Houdini power user tips and techniques, Craig Zerouni.
- [7] Side effects tutorials.
http://www.sidefx.com/index.php?option=com_content&task=blogsection&id=14&Itemid=132
- [8] <http://forums.odforce.net/>
- [9] The Magic of Houdini, William Michael Cunningham
- [10] P.Joyce, Final Project *Image Synthesis, Rendering an ice cube*
Available at <https://graphics.stanford.edu/wikis/cs348b-06/JoycePan/FinalProject>