

## Virtual Reality in Education Special Effects with Grid Node in Houdini Sidefx

Chaman Lal Sabharwal

*Missouri University of Science and Technology  
Rolla, Missouri – 63128, USA*

---

**Abstract:** Virtual Reality is informally used in many applications where visualization and animation are pragmatic. Special effects in Houdini sideFX are created by generating the illusions of realworld objects. We explore the hidden gems to show how Houdini SideFX can leverage simple node to construct complex objects, related natural phenomena and amazing realistic animations. We start with a grid node to systematically develop and explore complex 3D model views in virtual reality. With the advent of COVID-19, Virtual Reality has revolutionized academic and industrial world for remote interaction, by individuals, group meetings and training. In addition, it leads to innovation and development of new concepts, eBooks replacing the printed books. Virtual Reality was mainly in the realm of film and entertainment industry. It is changing the education culture completely all together and is becoming the mainstay in every aspect of visualization. With this paper we show how immersion results making complex theories as easy-go concepts to retain them in long term memory.

**Keywords:** Virtual Reality, augmented reality, 3D modeling, Visualization, Immersion, Interaction

---

### I. Introduction

Virtual Reality is informally used in many applications where visualization and animation are pragmatic. Virtual Reality has attracted a little attention in computing education. Manuals, internet tutorials concentrate on a specific aspect of the problem where a beginner gets lost after a few minutes into the tutorial. The tutorials lack the cohesive fundamental aspect of Virtual Reality for beginners. These dive into the tutorial before the complex concepts specific to either software or a discipline are understood. To fulfil this interlude, it is beneficial for the user to obtain an in-depth knowledge in an area before delving in applications for natural sciences or physical sciences.

Herein we begin with a simple mathematical data structure, matrix, labeled a grid in Houdini sidefx and explore several operations on it to create 3D models for Virtual Reality [Sidefx]. Visualization is fundamental to understanding complex concepts as easy-going data exploration and exploitation in many applications.

It is not well-known as to (1) what should be the content and structure of an introductory and advanced course in Virtual Reality that is suitable for computer science, (2) what direction emerges from the content and structure. The Virtual Reality is about three things: imagery, interaction, and immersion involved in modeling, animation, and visualization. We explore it in two tutorials to determine the issues in implementation and the approaches to resolve them in the process. We use Houdini sidefx for modeling and visualization. Houdini sidefx software is free for educators and students [McGee]. We also explore fundamentals of Augmented Reality which requires no hardware and lab [Ye J.].

Virtual Reality is used in natural and physical sciences. Human beings have five senses: haptic, hearing, sight, smell, and taste. In addition to information collected through sensory organs, the theories/laws, facts about the way things in the world play a part in understanding. Consequently, a person's perception of reality is likely to change with the introduction of theories and facts about the reality of things [Weidlich].

Movies are packed with special effects of incredible destruction shots and Houdini is used to create most of them. To refine the outcome of modeling, we will use rigid body tools: noise, collision, dopnet, popnet, pyronet, voronoi Fracture, rigid Material Fracture, rigid Fracture etc. Next we will dive a little deeper into the parameters of these nodes, in order to create more and more realistic special effects. For example, we will learn how to smash a grid-wall by a sphere-astroid. The examples that we construct are not just grid, but the applications built on grid that contain more challenging information to make you feel like a seasoned professional [Yilmaz].

Houdini is procedural in multiple ways: generate and deploy. For example: landscape can be created from grid node and grid is controlled where the houses, trees and walkways are located. Houdini sideFX is named after the Hungarian Magician, Harry Houdini. The increasing adoption of Houdini in Virtual

Reality applications opens up new possibilities for tools which can be adapted for focused workflows. It is based on the use of scientific algorithms in generative design and the application of Machine learning and AI in creative fields. We are more in line with a learning platform focused on immersive generative design with SideFX Houdini. With the advent of COVID-19, Virtual Reality has revolutionized academic world whether it is in-class teaching, online teaching, participating in the conferences in-person or remotely, personalized face to face meetings remotely and much more. In addition, it has pointed to innovation and development of new immersive course material in terms of eBooks and videoBooks replacing printed books. Virtual reality had its roots in the realm of film and entertainment industry. Virtual Reality is changing the education culture altogether, and is becoming the norm in software development.

In addition, the diverse modality continues to improve communication amongst a large variety of areas, engineering, industry, natural and physical disciplines. The advantages include learning curve reduction in complex subjects, learning material retention in long term memory and learner confidence building. Virtual Reality creates pretty pictures for visualization and immersive learning environment. Sometimes, Virtual Reality images are superior to real images. For example, in the case of natural disasters, it helps in motivating the audience to be proactively ready against hazards and how to be resilient to destruction resulting from tornados and hurricanes. Virtual Reality leads to innovation in many research areas [Sherman]. For basics of Houdini Sidefx, see the appendix A.

The paper is organized as follows: section II illustrates Houdini Grid node contribution to 3D modeling in Virtual Reality; section III is about Grid node and basic structures, Section IV furniture category modeling chair, table, tennis table etc, with grid node only, section V is about Construction, Brick Wall, House, Colony of Houses, Playground, section VI is nature; grass, animation wind effect disasters, section VII is Noise for the better, section VIII is grid with cloth, fire, ocean waves, section IX is rain, fire, fragmentation, section X is volcanos, hurricanes, section XI about Houdini Digital assets (HDAs), section XII is conclusions, section XIII is references and section XIV is an appendix.

## **II. Houdini Grid Node Contribution to 3d Modeling VR**

In Houdini, everything is a node, whether it is data or operation or even a file. A node can be a class, an attribute, a data object, or an operation function. Houdini programming is GUI based, drag and drop programming replaces formal code based programming. GUIs make it easy to create and maintain projects visually. One can design and develop a complete project with Houdini sideFX software without the need to write a single line of text-code. Houdini provides a digital version of real environment, it allows you to explore and exploit new concepts, one has never envisioned. Houdini features immersion: usability, manageability, efficiency, effectiveness, knowledge retention. It makes updates easy and transparent. Moreover, it is extensible allowing the user to add user created libraries by creating Houdini Digital Assets, thus enabling the user to easily create, edit, scale, embed, track across the deployment [McGee], [Ye J.].

A node can be created in several different ways: Two ways are from tool shelf toolbar by left mouse button (LMB) click for placing the node at a custom location in scene view or control+LMB click for aligning the node center with the scene view origin. Three ways from network pane, right mouse button (RMB) click drops down the nodes list (DDL), then selection from DDL or typing in the top textfield or selecting from history activates the selected node. The resulting node is displayed in scene view, its tab appears in the network view, editing node properties occur in the parameter pane, for scene view see Fig.1.

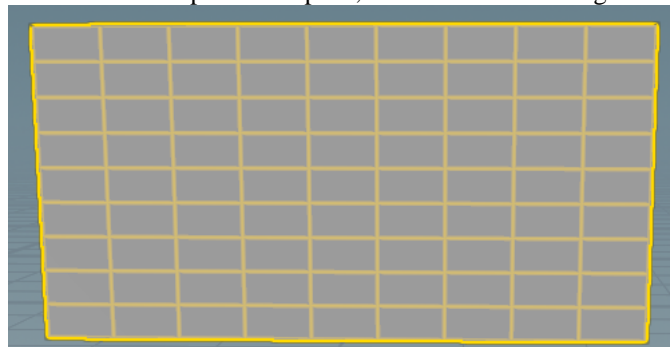


Figure 1. Grid to be used as basis for all special effects

## 2.1 Most Versatile Nodes

In each of the examples in this paper, grid node is the base for modeling any custom object. From a Simple Grid Node. All applications are built around data nodes in modeling. Data nodes are processed by several relevant operator nodes. In particular, most data nodes use a transform node explicitly or implicitly to update display properties. Thus, most of the time, transform node is used, and as a consequence, transform node is implicitly built in some nodes in order to avoid creating a separate node. The attribute values are edited several ways: (1) one way is to “Type in” values, (2) “copy and paste relative reference parameters” is used to align the values of relevant parameters, (3) using explicitly visible thumb slider, (4) using implicit slider (ladder slider), or (5) dynamic values using a function involving frame number  $F$  that corresponds to animation playbar.

## 2.2 Frequently Used Operation Nodes

The Grid node is the simplest and most adaptable node. The *transform*, *color*, and *copy* nodes are frequently used for various reasons. For documentation *font* node and *file* node are also used. Font node is used to display self-documenting text in the scene view. To place it at the right place, one uses implicit transform (rotate, translate, scale) to move it to desired location. For example, it may be used to identify the resource developer. The color node is used to color displayed object-components for esthetic reasons. Color parameters type *constant*, and class *primitive* are sufficient. The *file* node stores info data or images. The file node is used to include a file with the description and image, logo, pertaining the organization of scene view. For any file location reference, a click on arrow on the right of textfield in parameter pane brings drop down description list (DDL) of node options, scroll the directory path to filter the path to where the file is. Like font, operator transform is applied to place the image information at the desired location. To paste an image on an object, it is as easy as creating a uv Quick shade node for the image. For example, a globe is created by uv Quick shade pasting a worldmap onto a sphere very easily.

For custom visualization, one can transform the object display depending on scene view layout. It is preferable to stay in the current desktop view during the whole modeling process. Normally, any changes may be made in parameter pane rather than scene view transformations.

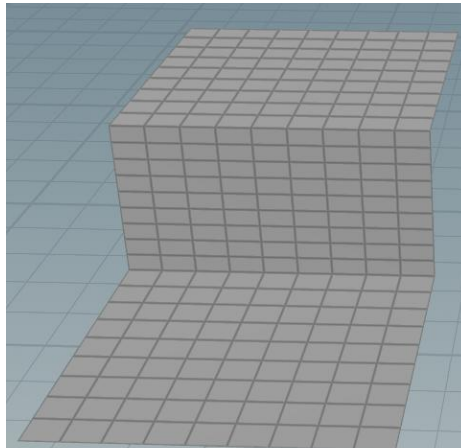
## 2.3 Art of Grid Node

In Houdini, some support nodes are automatically created by the software to free the user from micromanaging the nodes. It usually happens implicitly when solver nodes are created. In this paper, grid node and essential additional nodes, box, circle, sphere, tube, curve, line are used to illustrate the Houdini grid node applications. They are used to create hybrid objects: tennis court, building, table, plates, pipes, stairs, and operator nodes (OPS) for growing grass, coloring, creating trees, wind effects, cloud, fracturing, rain, smoke, fire, fracturing effects. Fundamental OPs nodes, commonly used nodes are: merge, transform, polyextrude, copy, revolve, sweep, bend, merge-boolean-switch, popnet, pyronet, rigid body dynamics. Pasting an image on an object is accomplished by uv Quick shade node. Boolean node parameter Shatter breaks objects without the need to go to rigid body object fracture [Sidefx]. In the next section we apply the basic operation nodes to grid node concepts [Grenfell].

# III. Grid Node and Basic Structures

## 3.1 Grid Node and Staggard Steps

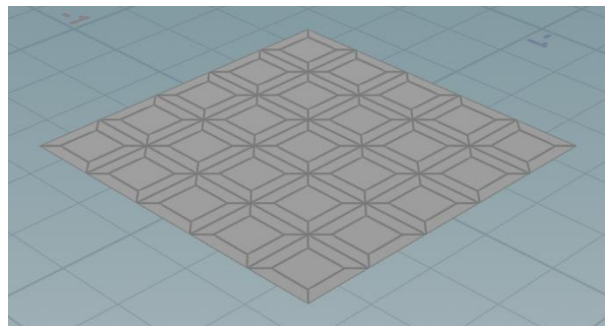
Here a simple  $10 \times 10$  grid node of physical size  $1 \times 1$  is created. Three versions of the same grid are created by rotation, scaling, and translation. Upper part is the result of translation of lower, base, in z-direction by  $-\frac{1}{2}$  and y-direction by 1. The middle part results from lower part by: 1. Scale along z-axis by  $\frac{1}{2}$ , 2. rotation about x-axis by 90 degrees and translated in y-direction by  $\frac{1}{2}$ , and along z-axis by  $-\frac{1}{2}$ , see Fig. 2 it looks like a single piece of grid node



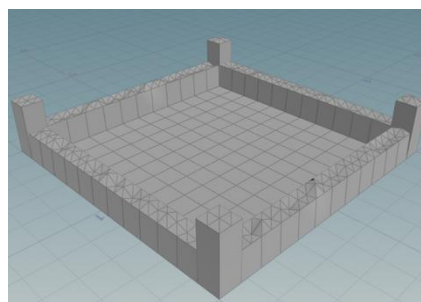
**Figure 2.** Three versions of same grid stacked into step

### 3.2 Grid and Patterns

This section describes generation of artistic patterns from the grid node. To create *Patterns* from grid node we use *polyextrude* node and its *inset* parameter to create artistic windows effect, see Fig.3. *Tower* like structure uses *polyextrude* parameters *distance* and *twist* to create twister tower. Grid *playground* with walls and lighting-pillars at the corners is created by *polyextrude* on grid node, Fig. 4. The  $m \times n$  grid primitives are indexed,  $0 - (mn-1)$ , primitives and grouped for efficient use. The tops of walls are created with glass like *fractals*, that is accomplished with using *indexing* primitives and *grouping* to *polyextrude* with parameter labeled *distance*.



**Figure 3.** Grid pattern with polyextrude parameter *inset* only



**Figure 4.** Grid pattern create by polyextrude *distance* parameter only on a select group of nodes

### 3.3 Artistic Buildings Grid and Highrise Structure

A grid node is created and then a *polyextrude* node is applied to grid to pull the grid upwards, in the first case, *polyextrude*'s *inset* option creates the effect of windows in the highrise building, in the second case *polyextrude* *twist* option creates the effect of twisted towers, see Fig.5-8.

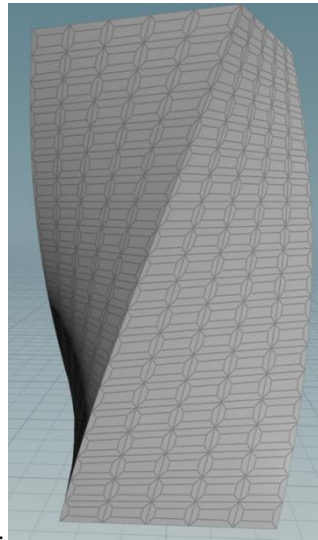


Figure 5. Grid pattern created by polyextrude with *twist* and *inset* parameters

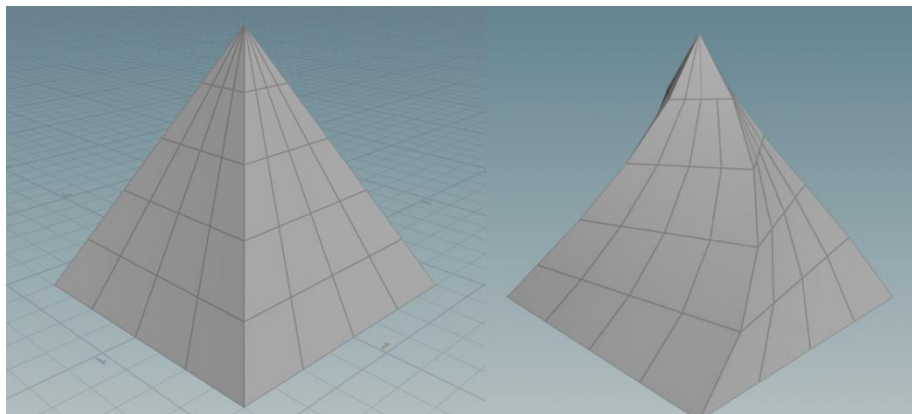


Figure 6. Grid node is used to create 3D pyramids by polyextrude with *distance*, *inset*, *twist* parameters

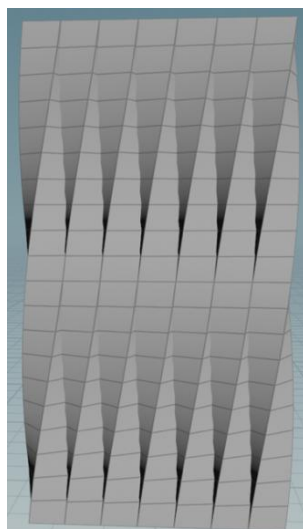


Figure7. grid node is polyextruded with *distance* parameter using *typeindividual elements*

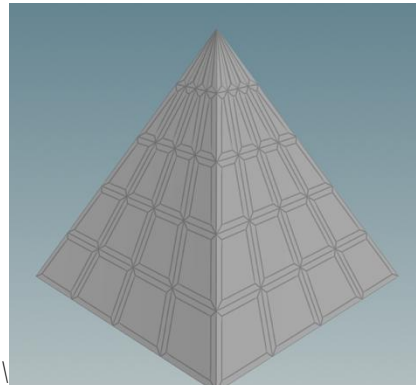


Figure 8. Grid is polyextruded with *distance*, and *inset* parameters, and inset parameter is applied separately to create primitive in the pattern.

## IV. Furniture

### 4.1 Grid and Chair

Grid node is used to create the entire the chair in three steps: the seat for sitting, back support to lean on, and legs for the chair. The seat and back are polyextrude of the grid node. Color node is used for seat and back. For legs, grid is polyextruded in two ways: one way to create the bottom four legs and the other way to create the back two connectors to link the back support and sitting seat, see Fig.9. Copy To Points node is used to accomplish this task.



Figure 9. It uses only one grid node to construct all parts of a chair for sitting. Polyextrude creates legs, color is used to color the legs.. Transform node changes the seat to back support, back support for comfort cushion.

### 4.2 Grid and Table

The grid, copy-to-points and polyextrude nodes are used to create fancy tables. Grid node is used here to create a rectangular table. The grid node is polyextruded to create tabletop, the grid is scaled and polyextruded to create a leg. Then the leg is copied to selected positions with the table top to make it look like a custom table, see Fig.10. Copy To Points node is used to copy the leg to a group of indexes on the grid.

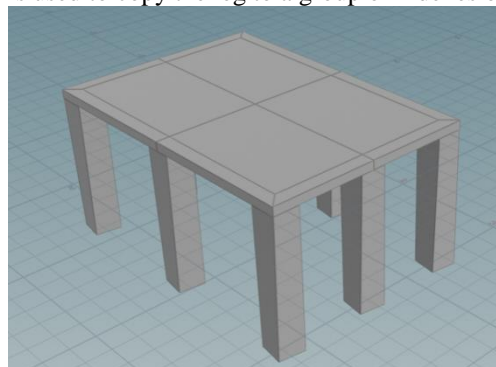


Figure 10. 3x3 grid table top, with a leg copied to eight grid vertices excluding the center grid point.



#### 4.3 Grid and Tennis court

The whole tennis court is composed one grid node. Flat plain grid is scaled to create the ground, courtwith boundary and the net. Color node is used to shade the ground and the court. Grid node on the ground is scaled, rotated and translated to become the net. Same net grid is scaled to make a post and is attached to the ends of the net. This completes the tennis court, see Fig. 11.

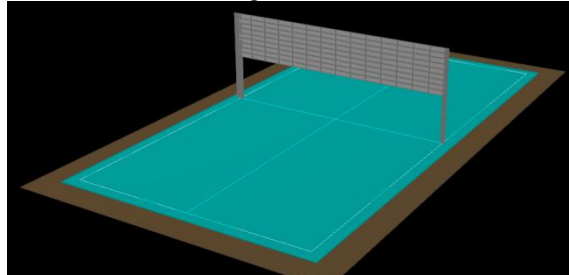


Figure 11. Tennis table is based on one grid node and transform node is used to form a net, polyextrude is used to create net poles

### V. CONSTRUCTION

#### 5.1 Grid and Brick: Design and Construction

For construction of brick objects with grid, the polyextrude node -*distance* - parameter is mostly used. For example, bricks, house structure are created with polyextrude distance parameter only[Poly]. Here brick is constructed from grid, using random color on grid primitives, polyextrude-*distance*, color-*primitive*, *constant*, values of nodes, see Fig.12.

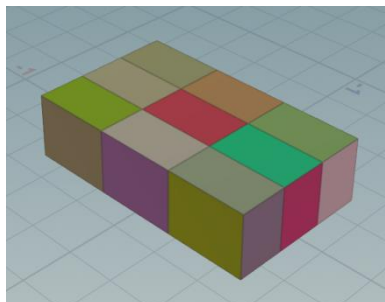


Figure 12. Brick from grid, polyextrude: *distance*, color: *primitive*,

#### 5.2 Grid and Brick Wall

To construct a brick wall, a linear grid *g* of the size of *n* bricks is created, another grid of the size of 1/2 brick is created. The grid *g* is shifted by half brick to become *g1*, Now *g* is append with half brick and *g1* is pre-pended with half brick and translated. The *g* and *g1* are merged so that the half brick is merged at the opposite end of the two layers. It makes a pattern of two rows of bricks aligned in such a way that it make a pattern of bricks. Now this pattern is copied several times and aligned so that it becomes a wall of bricks. See Fig.13. Add color in the grid lines as desired.

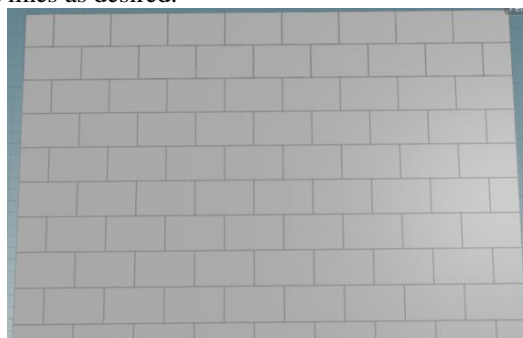


Figure13. Brick wall created from one grid support by transform and copy node

### 5.3 Grid and House Construction

The house is created entirely from grid node and ops. For house, in general, with grid, polyextrude parameters distance, creates the body. The group node selects top of body to create roof with special effects, parameter transform Extrude Front from polyextrude is used to create triangular type roof. Colors are used for esthetics. useful. The steps are: first grid is created. Then (1) Polyextrude is used to create a pattern with inset on individual primitive of polygonal grid, (2) polyextrude is used to create doors from grid, (3) Polyextrude is used on grid to create building, another polyextrude creates the roof. In the second house door is created differently from grid and colored, see Fig.14-15

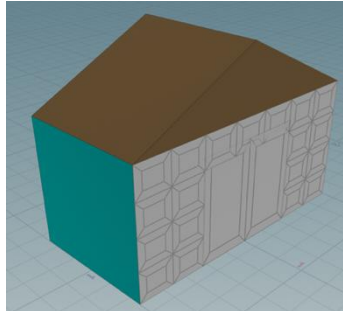


Figure 14. House from grid, polyextrude: *distance*, *inset*, transform Extrude Front, entry doors

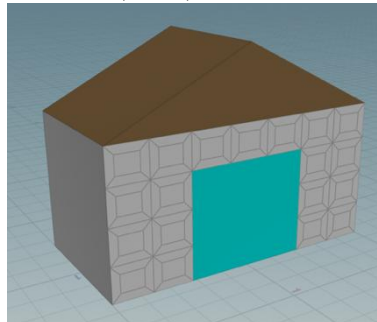


Figure 15. House from grid, polyextrude: *distance*, *inset*, transform Extrude Front, entry colored

### 5.4 Grid and Polyextrude Wear house

This is a flat roof interesting building like a ware house made up from grid, polyextrude, and color. Grid index is used to group extruded legs and roof objects, see Fig. 16.

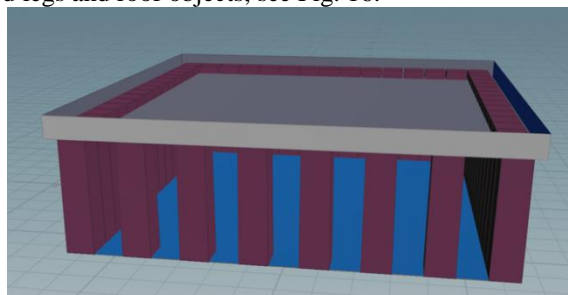


Figure 16. Grid, transform, and polyextrude: *distance*

### 5.5 Grid and Colony of Houses

From grid we create a house as above in the paper, we use copyToPoints to copy the house in the form of rectangular grid or circular shape by using rotation parameter of transform and copy node. *Rectangular copy to Points* copies, see Fig.17. *Circular*, copy to Points copies a house on the grid vertices. Here we first create a first floor, then copy each floor to make it multistory building. *Multistory*- makes house, copy, and transform relocates at position and orientation described by the transform using translation, see Fig.18. *Spiral* – To make a spiral of buildings, we make a house, copy it, and transform it to relocate using both position and orientation described by the transform using translation (offset z) and rotation, see Fig.19. It uses



transform with copy node to create spiral. Finally to make *Stairs*, staircase, we make brick step from grid, use transform and copy node again, with added translate in the y-direction (offset y). grid and copy to points on a rectangular grid . grid, first floor, and copy to pointe verticallyFig.20. Then add tube in the center.

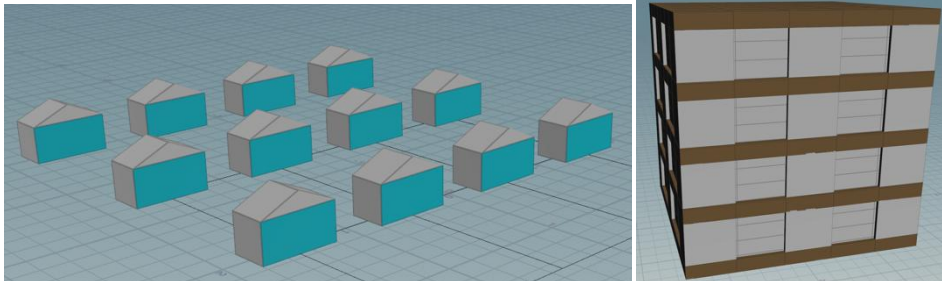


Figure17 grid and copy to points on a rectangular gridFigure18 grid, first floor, and copy to pointes vertically

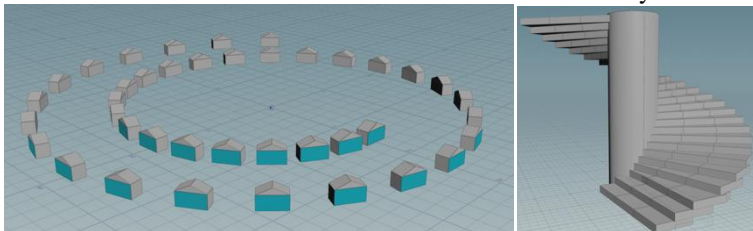


Figure 19.Spiral from grid, transform and copy (offset z)Figure20 Stairs from grid, transform and copy (offset y)

## 5.6 Grid and Playground

Playground construction begins with an  $m \times n$  grid. The  $mn$  vertices and  $(m-1)(n-1)$  primitives are indexed. Then we create groups using indexes of grid primitives, groups are used to represent walls, corners etc as needed. Polyextrude is used to extrude selected primitive groups. Now the groups of indexes are for walls, corners, additional pillars at the middle points of walls and the center of the grid. Grid is multipurpose, Grid is used to create simple playground from grid, grid corners are like light posts at night, walls form an enclosure, and fractals on top of walls form cosmetic fractured glass security, see Fig.21.

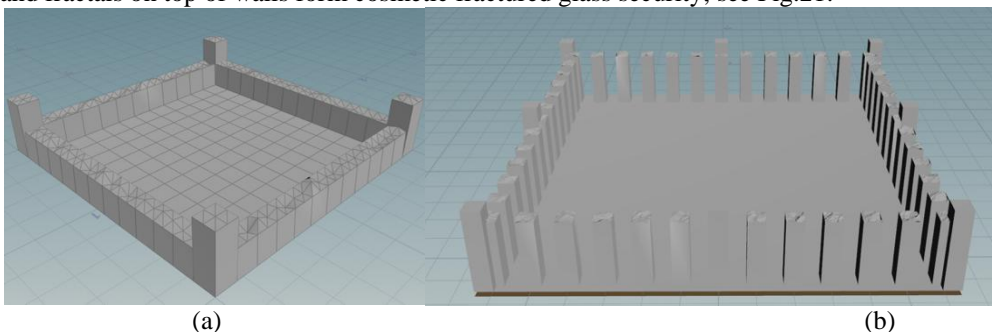


Figure 21.(a)Playground with close entry walls (b) Playground with mid pillars and open entry walls

## VI. Nature

### 6.1 Grid and Grass animation with wind effect.

Here grid is used as base where grass grows at the grid points. Grass is simply grass blades made up of copies of a line or a curve node. Sometimes line is not visible, we use polywire node's radius parameter to make it thick for user friendly visibility.

Grass is created from line node and curve node mixture to look like natural grass blades. Green color adds greenery to grass. Once one set of initial grass blads is created, it is copied with copy To Pints node on the grid. Lastly play ground enclosure is added. Further to simulated wind effect, transform node, sweep node and bend nodes in conjunction with special function on \$F yield wind effect.

For special effects: To make the grass blades visible, polywire node adds thickness effect. Resample node adds more points to make them equally spaced and creates smoothness in the curves[Curve]. Wind effect is created with controlled rotation about x, y, z- axes. Grid and grass effects follow in Fig. 22-25.

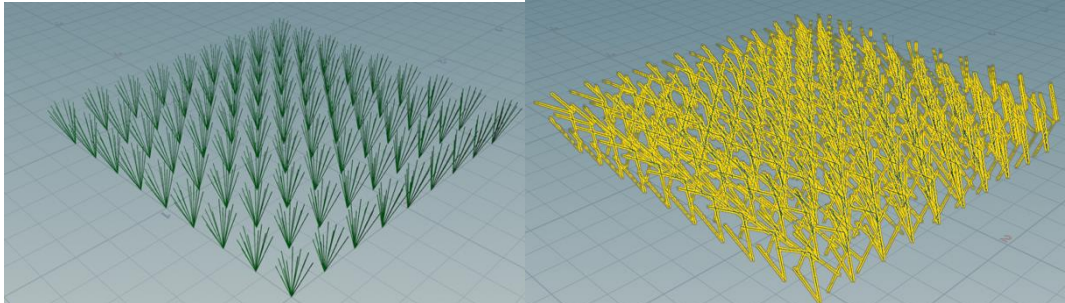


Figure 22. Green grass with distinct grass blades Figure 23. Yellow grass with distinct with visible wind effect, grass blades with visible sun heat effect.

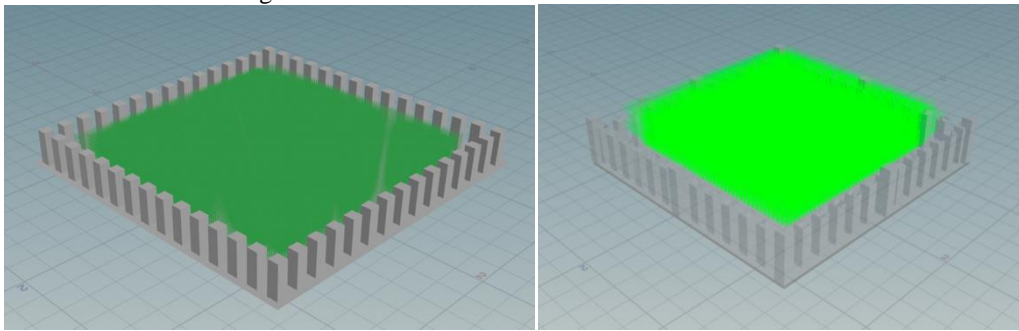


Figure 24. Green grass dense in the playground

Figure 25. Dense Green grass in playground with middle Pillars grid and dun-dried grass

## 6.2 Grid and Animation

Animation is the trademark of virtual reality. In Houdini, animation is built in Solver nodes, solvers provide the animation on all kinds of actions. Custom animation is created with frame variable, \$F, and associated functions for attribute values to create special effects. We use frame number for \$F on color values for animating colors.

Animation is displayed with the use of animation playbar. Here we create custom animation using simple grid node. Simple animation of colors on the grid primitives pattern cycles the color between red and green giving the animation special effect. Such animation can be accomplished with other nodes also. Here we create two grids: One is a polyextrude with *individual elements* and *inset* only to create the appearance of windows. Second is same size grid colors alternating with even or odd index. The first grid is overlaid on the second grid and playbar does the animation. The grid one color reflects through the windows to create color animation effect through holes in the second grid, see Fig. 26-27.

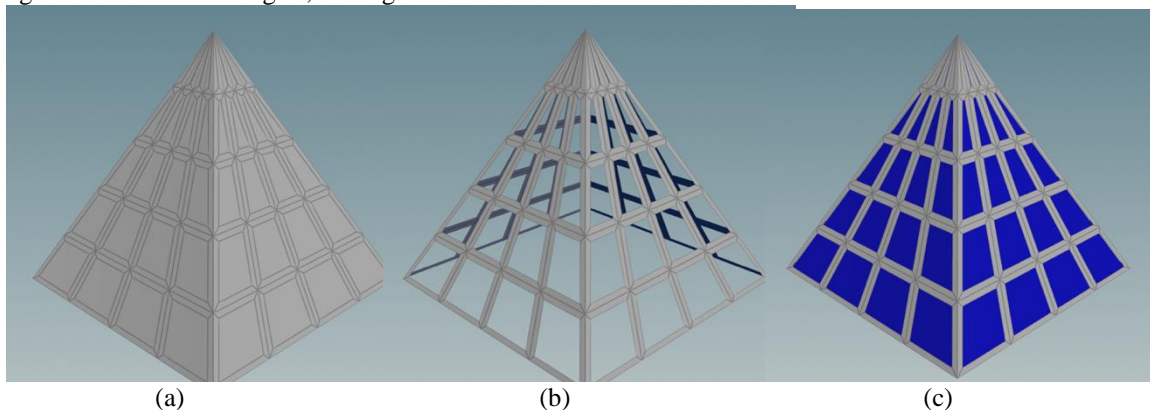


Figure 26. (a) Grid is polyextruded and pattern created by inset node on sides, (b) polyextrude front face off created windows, (c). the two displays are overlaid. Back pyramid is colored to animate between red and blue.

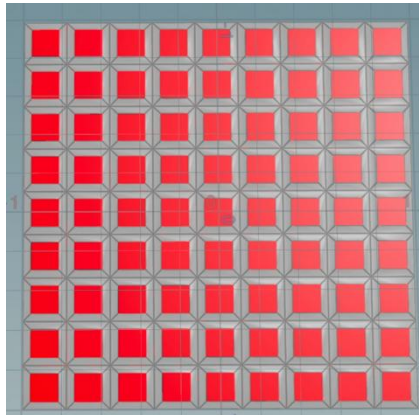


Figure 27 Polyextrude with only *individual elements* and *inset* creates appearance animation.

## VII. Noise for the Better, Special Effects

In this section, we will show how the grid node leverages other nodes to create special effect, the nodes we mean are: noise (mountain, fractal, pointjitter, attribnoise, scatter), popnet, pyronet, to create special effects. Noise is added to grid to make more useful, unlike less useful in data mining problems. In some applications like data mining, machine learning, noise is a source of errors in data. The noise is removed before any mining algorithm is used to make reliable and accurate predictions. On the contrary, noise is introduced to make geometric environment more natural and interesting. There are diverse types of noise to create special effects.

### 7.1 Noise and Visualization special effects

There are several ways to create special effects. The Noise category includes *Mountain* and *Fractal* nodes to convert the grid into surface of mountain. Scatter node also used to create noise. The grid uses *isooffset* node to map grid to cloud, fog type surface. Mountain and isooffset together create *snow* effect. *Bend* is used to grid structures into fancy shapes, see Fig. 28. The taper-squish parameter allows it to scale the grid structure artistically at any point. Grid fractal noise creates unevenness in a plane surface similar to water waves in the ocean, see Fig.29-33.

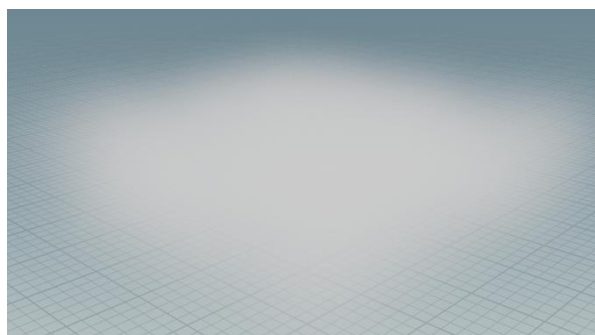


Figure28.Grid and Isooffset node creates fog effect.  
It is useful to create smoke, sky and clouds effects also.

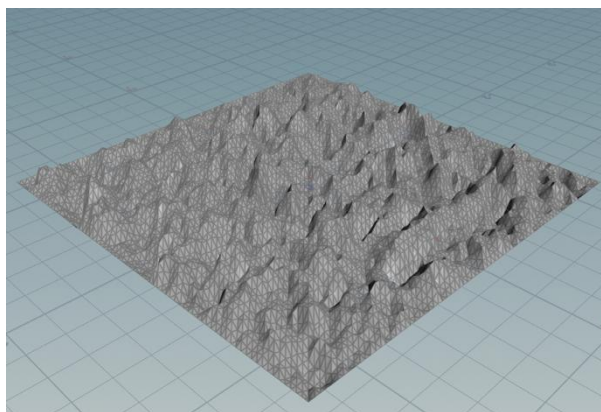


Figure29.Mountain and Fractal Noise effect on grid

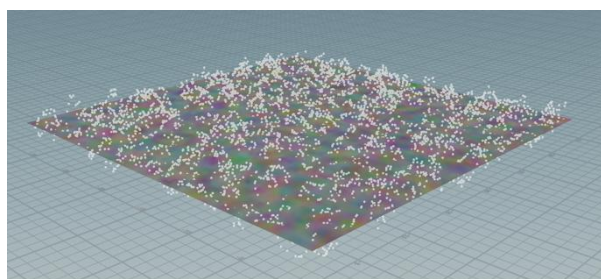


Figure30.Grid and Attribnoise effect on grid

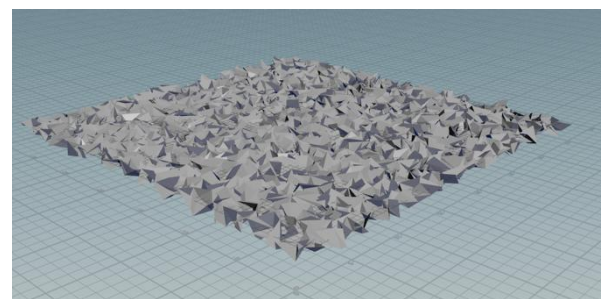


Figure 31.Grid and Point Jitter effect on grid

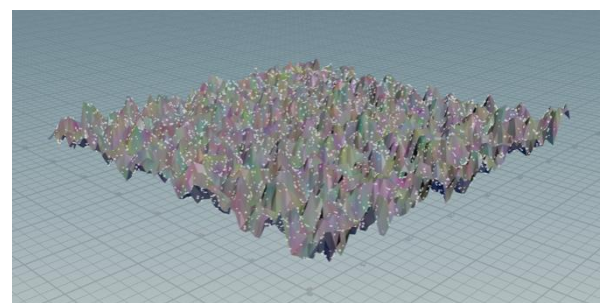


Figure32.Grid, Mountain and attribnoise effect on grid



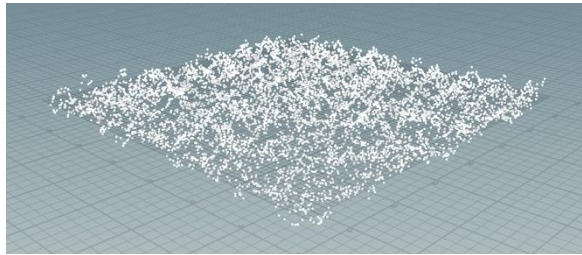


Figure33.Grid and Snowy mountain effect on grid

## VIII. Grid with Cloth File Ocean Waves

### 8.1 Grid and Fountain

To start with, here two uses of grid with popnet node are shown. Popnet node is used to create fountain like effect over the whole grids or at the selected primitives, see Fig. 34-35.

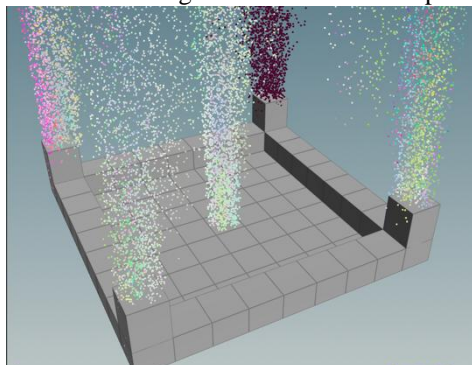


Figure 34.PopnetFountainsLighting Effect

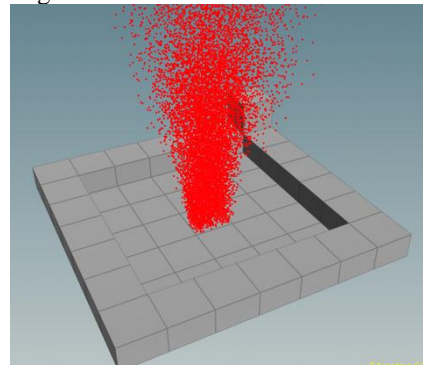


Figure 35.PopnetFountainFire Effect

### 8.2 Grid and Rain

Rain with popnet, popnet node creates particle out of an object and display them in view port. If particles are not visible, they can be simulated for view with little spheres, Or from scene view, type, d and go to geometry and increase the pixel points size to maximum, e.g. .d→geometry→pixel point, , see Fig. 36.



Figure 36. Grid and Popnet for special Rain effect

### 8.4 Grid and Cloth

Cloth vellum cloth node makes grid look like a piece of cloth. It adds a waving effect. Cloth does not stretch or sheer, but can bend to deform the grid. Cloth node parameter type is quadrilateral or triangle for display, even if it is quadrilaterals, Houdini uses triangles internally. When the cloth moves and animates, the “pin” parameter is used to hold cloth, to prevent cloth from flying away due to wind. The “pin” keeps the cloth unaffected by wind motion on cloth grid and holds it fixed at the selected “pin” points. A very good example of the use of pin is flag simulation. In this example, flag image is posted on the cloth grid object, see Fig. 37-38.

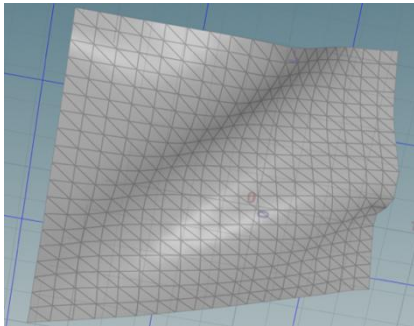


Figure 37 grid displaying the cloth effect,

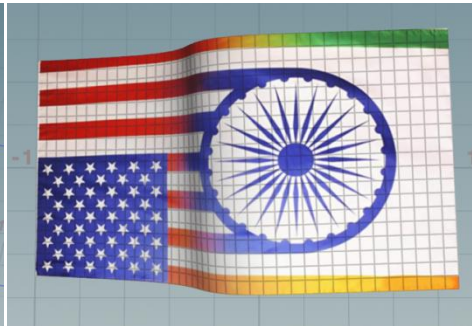


Figure 38 grid with uvquickshade map posted on cloth effect

### 8.5 Grid and OceanWaves

Grid can be made into water effect. In order to create water effects, particularly, ocean waves grid is supported with oceanSpectrum to create wave effects, and oceanEvaluate to create animation of the waves. Additional nodes oceanwaves, oceanfoam, may also be used for special effects. Oceanspectrum is a collection of several waves and parameters. We use the default parameter values to begin with. Ocean evaluate is a solver for animation of various forms of waves generated by varying the parameters in depth for waves fall off. Points increases/decreases the height of waves. Downsample maintains the number of waves, smaller the value, more the waves, see Fig. 39-40. Other parameters include time- speed.

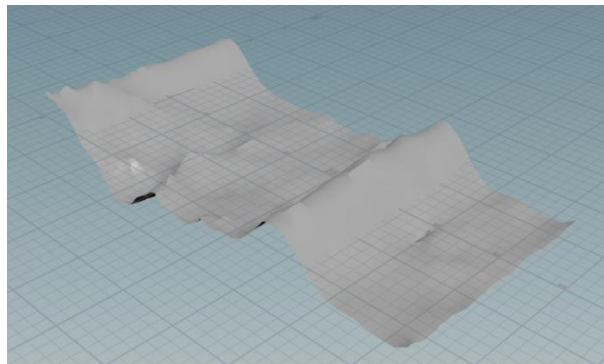


Figure 39. High intensity wave with more points, high intensity waves with larger down sample

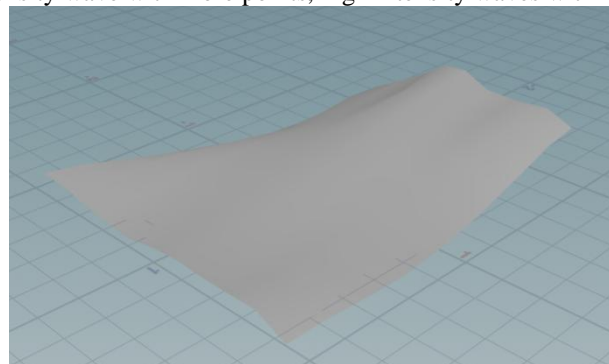


Figure 40. High intensity wave with less points, low intensity waves with smaller down sample.

### 8.6 Grid and Files

Before an application is deployed for use, it may also be desirable to identify the source of application designer. Houdini can be easily adapted to include developer identification source and logo, see Fig. 41-42.



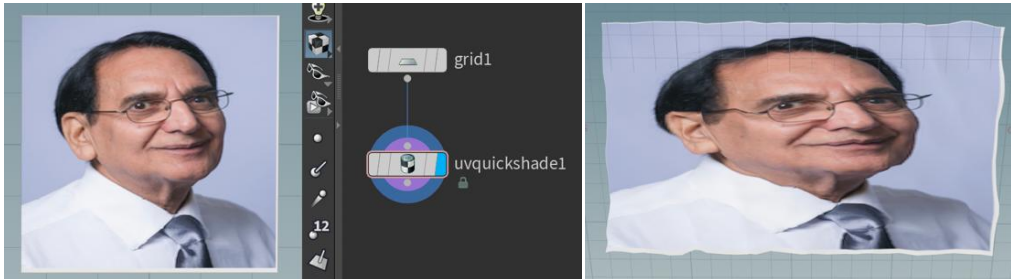


Figure41.OceanWave Effect on an image. Figure 42. Grid node displaying Author's depiction and title

### 8.7 Grid and Bend

Grid-polyextrude creates a vertical column. Bend node bends the top, The *bend-taper-squish* parameter is used to create special effect of scaling about the squish-pivot, see Fig. 43.

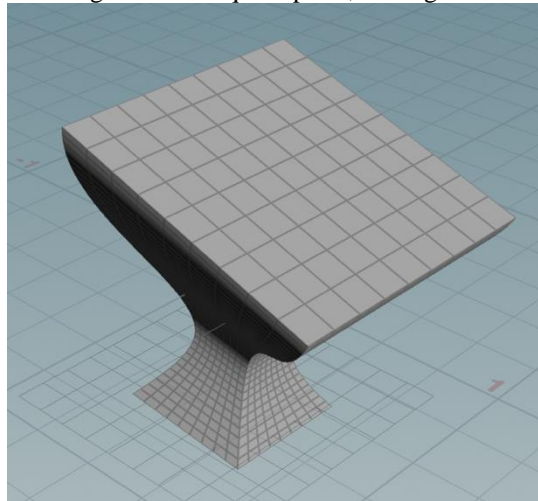


Figure 43.Grid-polyextrudeand Bend parameters *bend-taper-squish*

## IX. Rain, Fire, Storm, Shattering, Fragmentation, Destruction

Grid can be a foundation to create dust, cloud, smoke, rain, fire, fracturing, destruction, hurricanes. We have seen that Grid corners and center depict the effects of fire using popnet. We will soon see how easy it to create volcano effect from grid and popnet nodes. A simple example follows.

### 9.1 Grid Cracks

For a grid, several nodes such as shatter, voronoi Fracture, rbd Material Fracture create cracks in an object, but they stay glued together. In order to make the cracks visible or fragment the cracked object, exploded view may be used, see Fig. 44-45.

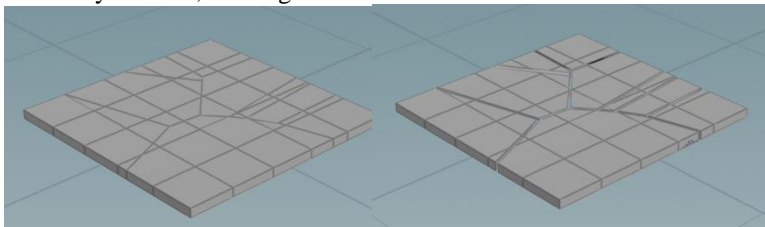


Figure 44.VoronoiFracturegrid,

Figure 45.Fracture cracks

### 9.2 Grid Shatter-Boolean

Here grid node is created and for visibility, it is made a little thick by polyextrude node. Two identical copies, A and B, are placed next to each other. Normally Boolean node has rigid body operations union,

intersection, A minus B and vice versa, B-A. Houdini has another operation called “shatter”. Shatter option is built in the Boolean node. Shatter node creates Voronoifracture where shattering occurs [Pixel]. If we use that operation, then it separates the union of A and B into three parts: A-B, B-A, and  $A \cap B$ , see Fig. 46-47.

Voronoi fracture parameter cut Plane Offset can be used to display the grid cracks at the fracture points. Shatter is also built in the Boolean node, see Fig. 46.

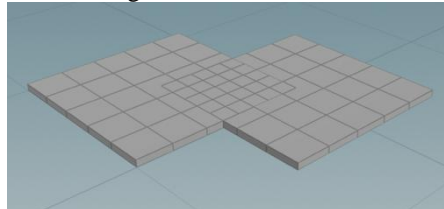


Figure 46. Boolean node with shatter and intersection both display the fragments, A-B, B-A, and  $A \cap B$ .

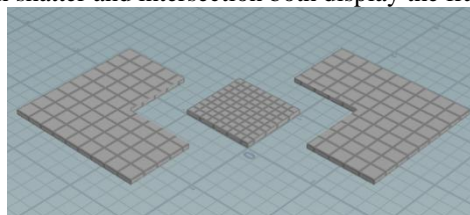


Figure 47. Boolean node with shatter intersection displaying fragmented pieces

### 9.3 Grid and Fragmentation

Easiest way to fracture and view the fractured grid is to shatter it into primitives and use exploded view to view the fractured pieces. If we use shatter node directly along with exploded view, the grid object is fragmented, see Fig. 48.

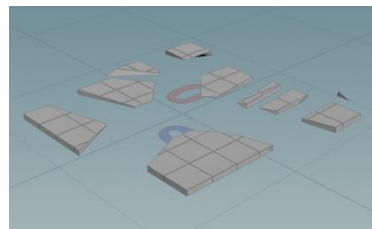


Figure 48. Shattered grid, Fragments at Fracture Cracks.

### X. Grid, Volcano and Hurricane

It is incredibly easy to create volcano effect from popnet. In this paper we are creating the Grid effects, so grid is used to create the volcano opening. We used grid for generating particles to be blown away with velocity parameter in the popsource attribute tab. The attribute tab “add to velocity” has velocity parameter used to set the direction flow of particles, spread of particles is via variance parameter, and force of wind blowing. Here grid is extruded with desired *distance* and *inset* parameter values to create volcano mouth. Popsolver animates the volcano eruption for visualization node. Windstorm effect spreading the particles is created similarly by changing the variance for velocity directions sideways. Extra spheres of varying size are floated to give the effect of storm, see Fig. 49.

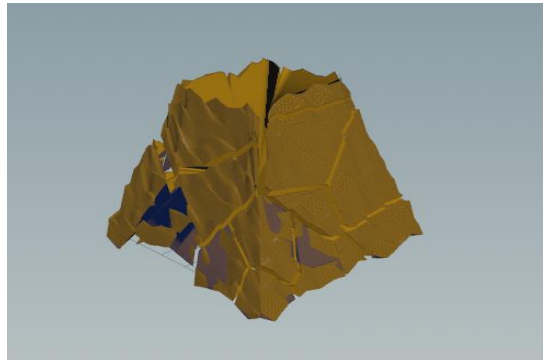


Figure 49. Volcano Opening based on grid node

### 10.1 RainStorm

A grid node is created and a copy of the same grid is created. The second grid is offset vertically so that the two are located on the top and bottom of the screen. Four copies are made for depicting four operations on the grid pair: (a) Two grids with scatter points, (b) popnet on the bottom grid is emitting fire particles, (c) popnet is applied to top grid to emit particles to create rain effect, (d) rain is trying to subside the fire. Fire emission starts from the ground and moves towards the sky. Rain falls from the sky to counter the fire, see Fig. 50. It will ultimately turn into smoke.

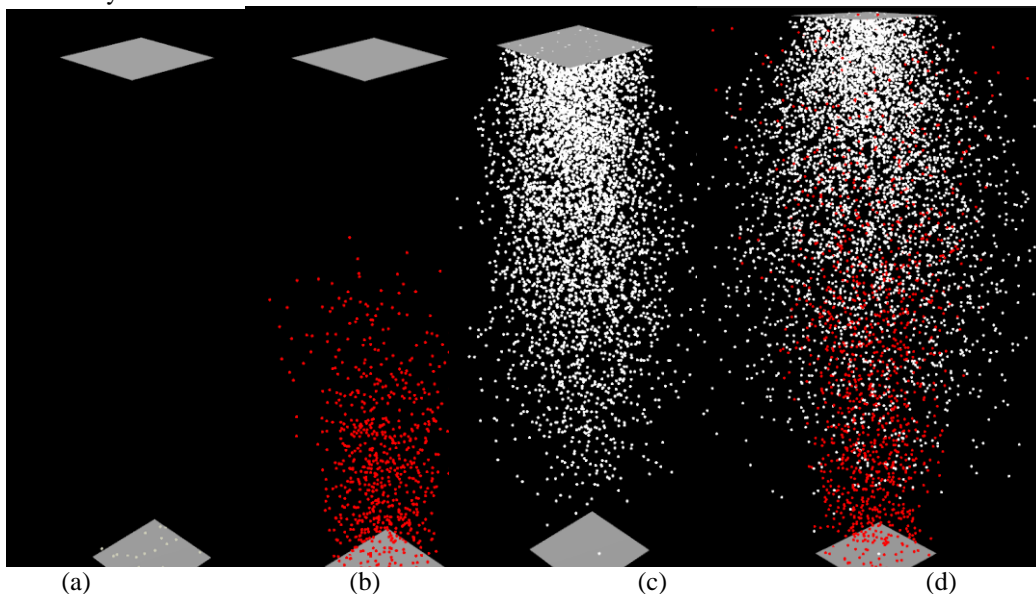
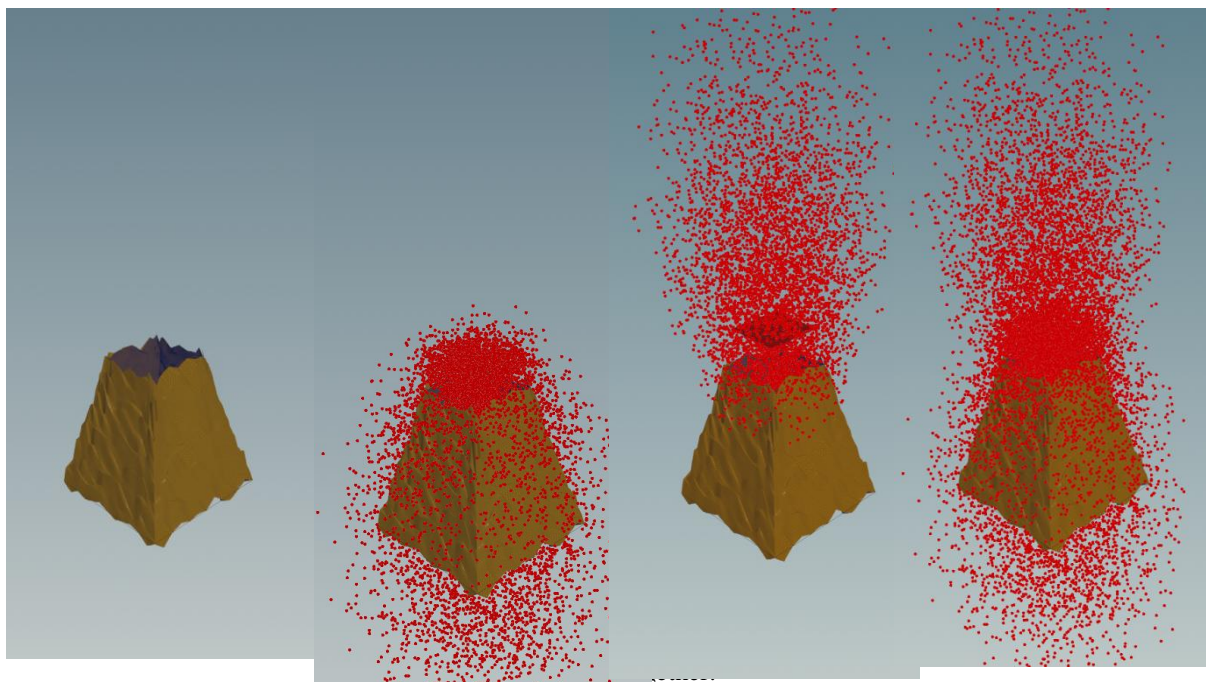


Figure 50. (a) Two grids with scatter points, (b) Bottom grid is emitting fire particles, (c) One grid is emitting particles to create rain effect, (d) rain is trying to subside the fire.

### 10.2 Volcanos

In this section, grid node issued to create volcano eruption effect. Three stages of volcano eruption are described here. The grid node is supplement with popnet node. Using the velocity parameter of pop source node in popnet node, these special effects are created [Sidefx]. It is the same as individual cases along with a volcano's opening. Thus, volcano opening is a grid supported with polyextrude *distance* and *inset*, *fractal* noise, popnet emits particles and blows the particles: (a) volcano opening is depicted, (b) we see that lava is overflowing from the mouth of Volcano overflowing, (c) lava is shooting up from the mouth of volcano, (d) lava is shooting up and down overflowing together covering the volcano source.



### 10.3 HurricanStorm

This is a very simple form of storm. For this, ponet creates particles from grid node for rain effect and objects are little spheres of vary sizes. They are copied on the blowing particles to give special effect of storm blowing away objects during rain, see Fig. 52.

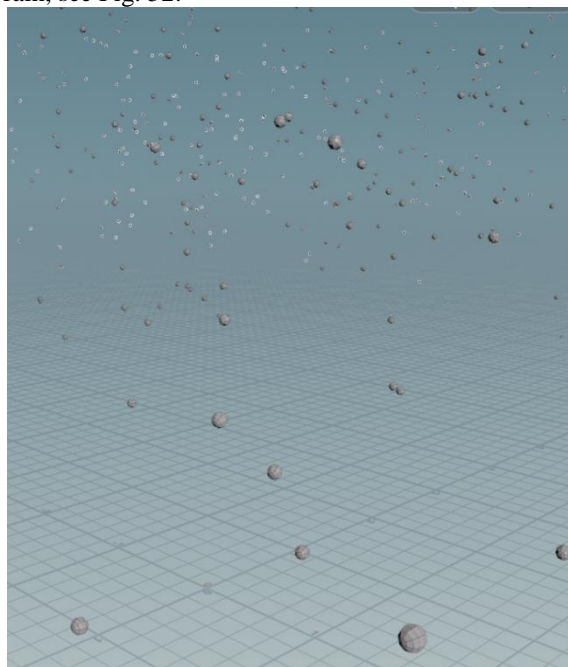


Figure 52.Grid and Storm special effect (Popnet)

### 10.4 Grid and Fire

There are various ways to create fire and smoke effect from the Grid node. They are Grid and Pyro source, Grid and Volume from polygon, Grid and Volume Rasterize, Grid and cloud, Grid and sky.

Here Pyro FXbonfire node is one among the several noded for destruction with fire. Thebonfirenode operates on the grid object and burns the grid object with fire, see Fig. 53.

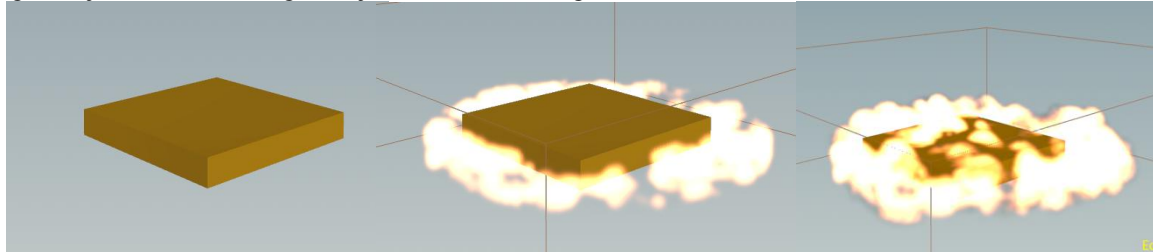


Figure53.Grid and PyroFx bonfire node to create fire.

### 10.5Grid and Rigidbody Dynamics

Besides the exploded view to see fracture, there are other tools for fracturing 3D objects. Exploded view is static, there is no animation in it. There are other nodes that have animation along with fragmentation. Two of the geometry based nodes are rbd Material Fracture and rbd Bullet Solver. Two of the obj level nodes are rbd Fractured, and auto Dop Network. Here again we have a grid wall and a grid brick, colliding with the wall, see Fig. 54. Three effects of the wall are (a) wall and brick break on collision, (b) wall does not break, brick goes through,(c) wall does not break, but breaks the brick on collision, see Fig. 55.

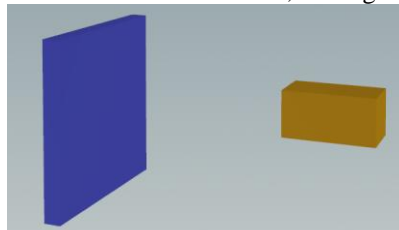


Figure 54. Grid node wall and a grid node brick for collision.

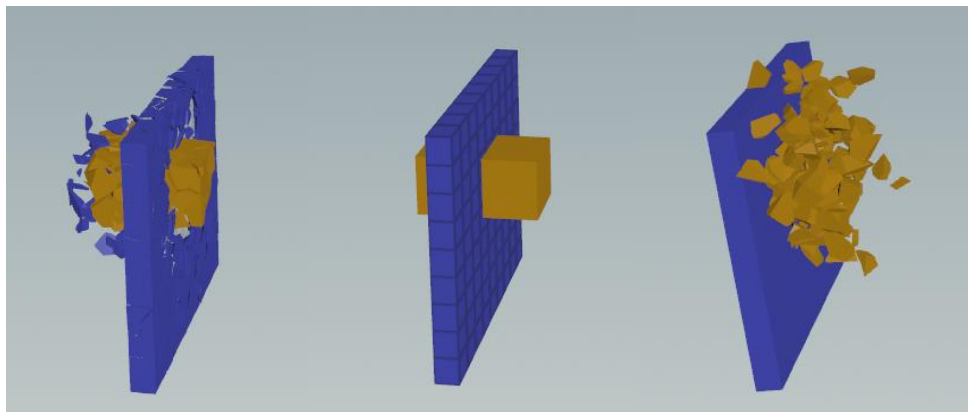


Figure 55.(a)Ball hits the grid wall and both break,(b) Ball hits the grid wall and goes through it; (c)Ball hits the grid wall but cannot go through and breaks itself.

## XI. Procedural Design in Houdini Digital Assets (HDAS) and Object Oriented Design

In object, oriented development, objects are created similar to libraries for reuse instead of recreating the same entity over and over again. Houdini provides a procedural mechanism to create Houdini Digital Assets at object level as well as geometry level. In the sequence of nodes, any node can be activated or deactivated/bypassed to explore different features. The HDA is installed by double click the HDA. To instantiate it, we consider it as any other node, it appears in the Digital Assets tab of dropdown list generated by right mouse click in the networkpane, by following a simple sequences of self documenting dialog boxes. The HDAs can also be used as the other shelf tools in Houdini. For example, somethings are just for information display only, e.g, developer's title and depiction [Simon], [Xiaojun].



## XII. Conclusion and Future Work

In this paper we used Houdini SideFX, node based procedural modeling, to generate various interesting shapes and patterns. We have given enormous examples illustrating grid node contribution. We have shown how versatility of the Grid node in Houdini sidefx. In particular, we have used only one grid node and supporting operators to explore modeling of chair, table, tennis table, brick wall, house, colony of houses, highrise buildings, playground, grass, cloth, oceanwaves, rain, fire, destruction in natural disasters, volcanos, hurricanes. All these special effects are based on only one Grid node along with relevant operators.

We hope that reader will enjoy watching the screen shots and get incentive to envision challenging models for education and entertainment [Xiaodong]. These models can be created very easily using simple data nodes and operation nodes.

## XIII. References

- [1]. [Curve] Houdini curve node <https://www.sidefx.com/docs/houdini/nodes/sop/curve.html>
- [2]. [Grenfell] Grenfell J., (2013). Immersive Interfaces for Art Education Teaching and Learning in Virtual and Real World Learning Environments, *Procedia-Social and Behavioural Sciences*, 93, 1198-1211.
- [3]. [Haf] Häf er P., Häf er V., O t haro a J., , Teaching methodology for Virtual Reality Practical Course in Engineering Education *Procedia Computer Science*, 25, 251-260.
- [4]. [McGee] Robert McGee, Houdini Foundations book, <https://www.sidefx.com/tutorials/houdini-foundations-book/>
- [5]. [Pixel] Pixel Fondu, <https://www.youtube.com/watch?v=xBt7egf0kIU>
- [6]. [Poly] Houdini polyextrude node <https://www.sidefx.com/docs/houdini/nodes/sop/polyextrude>
- [7]. [Sherman] Sherman W.R., & Craig A.B., (2003). Understanding virtual reality-interface, application and design. The United States of America: Morgan Kaufmann Publishers.
- [8]. [Sidefx] Houdini in SideFX <https://www.sidefx.com/products/houdini-apprentice/>
- [9]. [Simon] Simon Verstraete Houdini Digital Assets, introduction to HDA, December, 2021, <https://www.youtube.com/watch?v=ybk8XWBnO4o>
- [10]. [Weidlich] Weidlich D., Cser L., Polzin T., Cristiano D., & J i k er H., Virtual Reality Approaches for Immersive Design, *CIRP Annals - Manufacturing Technology*, 56(1), 139-142.
- [11]. [Xiaodong] Xiaodong W., Dongdong W., Yue L., & Yogtia W., , Teaching based on augmented reality for a technical creative design course, *Computers & Education*, 81, 221-234.
- [12]. [Xiaojun] Xiaojun Fan a, Xinyu Jiang a, Nianqi Deng b, Immersive technology: A meta-analysis of augmented/virtual reality applications and their impact on tourism experience, <https://doi.org/10.1016/j.tourman.2022.104534>
- [13]. [Ye J.] Ye J., Campbell R.I., Page T., & Bad i K., An investigation into the implementation of virtual reality technologies in support of conceptual design, *Design Studies*, 27(1), 77-97.
- [14]. [Yilmaz] Yilmaz, B., & Goken, M. (2016). Virtual reality (Virtual Reality) technologies in education of industrial design , *Global Journal on Humanities & Social Sciences*. [Online]. 03, pp 498-503. Available from: <http://sproc.org/ojs/index.php/pntsbs>

## XIV. Appendix A Houdini Sidefx Basics

There are several software packages to develop and deploy virtual reality projects [Xiaodong]. Houdini sideFX is a software of choice because it is procedural, flexible, enhancible without developer involvement. Moreover, Houdini Apprentice is free for the educators and students, and there is enormous support for the learner. There is more than enough educational material in terms of tutorials for the solo self-learner. Our interest is in simple algorithms for generative design using the application of Machine learning and AI in the creative fields using Houdini SideFX. In order to use Houdini, we give a brief introduction to Houdini user interface. It includes tips or shortcuts for 1. Houdini sideFX user interface, 2. Navigation of scene view pane, network pane, and parameter pane, 3. Simple frequently used nodes, 4. Conditional use of nodes, color and animation for visual effects, 5. Popnet for rain effect, pyronet for fire effect, and rbd nodes for fracturing effects..

### 14.1 First User Interface

The default Houdini interface consists of three panes. Scene view pane displays the objects, network pane shows the workflow network, and parameter pane is for editing the attribute values. Three pane interface is not rigid, but sufficient for small applications. It can be fine-tuned to accommodate complex workflow network.



In fact, the users may tweak it to replace default interface, then they have custom interface automatically installed every time. Here is compact and comprehensible description of interface for the beginner and experienced.

The first step is to understand how to navigate the interface before developing any application. We start first for visually recognizing the objects in the scene view.

#### 14.2 Scene view Background

Many times the color of objects displayed on the scene view overlap the color of the object making it difficult to isolate the view from the object. The screen background color can be easily improved to make the display crystal clear. To accomplish this in Houdini, type d in the scene view, then from the drop down dialog menu select background tab, then from color scheme tab, change the scene view background to desired color (from lighter to darker), to be able to see the model, see Fig. 14.1 below.

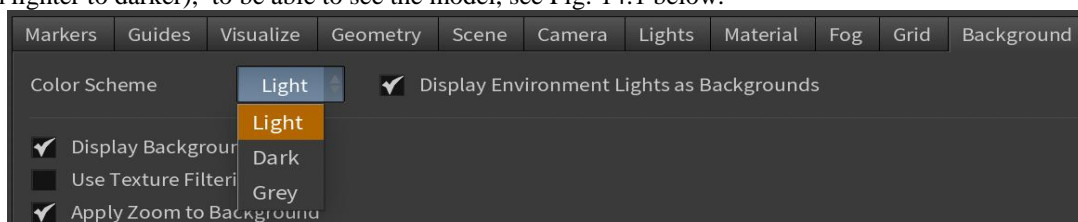


Figure 14.1 How the scene view background color is modified for better visualization of objects.

#### 14.3 Scene view Pixel Point Size

Due to enormous amount of display in scene view, sometimes the images displayed on the scene view are quite tiny in size that makes it difficult for the human eyes to identify them. To enlarge the pixel point size in Houdini, type d in scene view, select geometry tab, then at tab point size, change the pixel point from 3(default) to say 10, see Fig.. Then modelled points become large enhancing the visibility of modelled points. But, many times you want the size to be small, e.g. if you are working with rain drops, hurricanes etc.

#### 14.4 Pane Management

For navigation, it is necessary to manage panes, the size of any of the three panes can be fine-tuned to full window size by clicking on the little white square in the top right corner of a pane. This amounts to expanding and reducing the pane to full screen as desired [2].

A pane can be removed by clicking on the symbol "X" in the toolbars. If you remove the parameter pane, all the space is allocated to network pane. Now in the network pane, one can see the network workflow comprehensibly. Typing "p" in the network pane pops up the parameter pane subordinate to network for editing attribute values. Typing "p" in the network pane activates/deactivates the parameter pane, see Fig 14.2. To customize the desktop, LMB click on desktop in topmost menu bar brings down drop down list (DDL) in which there is desktop manager tab, creates desktop, now by clicking save, accept creates desktop. Now go to newly created desktop, rename it if desired, make changes to desktop, accept. Any desktop view can be saved by using the desktop manager. It can be referenced at will or can be made default current desktop see, see Fig. 14.3.

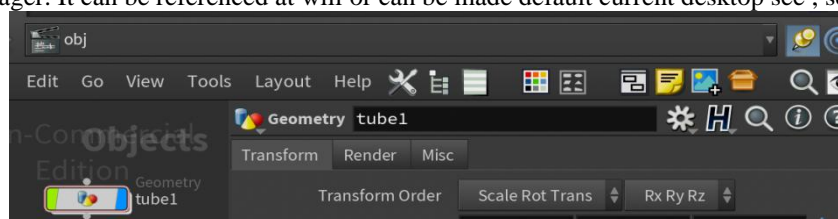


Figure 14.2 Parameter for network tube object is activated or removed or with pressing "p" in the network pane.

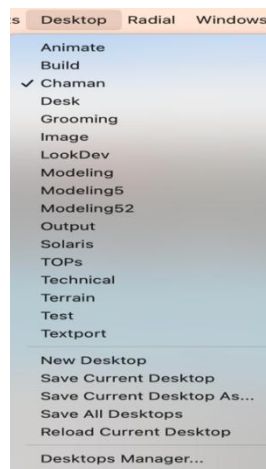


Figure 14.3 Desktop manager in toolbar desktop to create custom desktop, e.g. Chaman appears in the DDL