

Virtual Reality Special Effects with Houdini SideFX Grid

Chaman Lal Sabharwal

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

Abstract: Most of the special effects in Houdini sideFX are created by using one simple planar node. Some of the planar nodes are grid, circle, and heightField. These nodes have interesting properties similar and unlike in terms of geometry and special effects. We explore the hidden gems to show how Houdini SideFX can leverage them to develop “grid” based animations. We describe the progression of systematic development using elementary Houdini grid node and explore how this node can lead to complex multifaceted scene views in virtual reality. 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, individual zoom meetings and all that. In addition, it leads to innovation and development of new courses eBooks replacing printed books. Virtual reality was mainly in the realm of film and entertainment industry. It is changing the education culture altogether, and is becoming the mainstay in every aspect of visualization. With this paper we show how immersion results in innovation of new applications and how complex scenes can be created easily.

I. Introduction

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. VR 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, and natural phenomena. The advantages include learning curve reduction in complex subjects, learning material retention in long term memory and learner confidence building. VR creates pretty pictures for visualization and immersive learning environment. Sometimes, VR 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. VR leads to innovation in many research areas [1].

II. Houdini Sidefx Basics

There are several softwares to develop and deploy virtual reality projects. Houdini sideFX is a software of choice because it is procedural, flexible, enhance able without developer involvement. It is free for the beginner and there is enormous support for the learner. There is more than enough educational material in terms of tutorials for the solo 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 sceneview pane, networkpane, and parameter pane, 3. Simple frequently used nodes, 4. Conditional color and animation for visual effects.

1. First User Interface

The default Houdini interface consists of three panes. Sceneview 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 sceneview.

2. Sceneview Background

Many times the color of objects displayed on the sceneview 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 sceneview, then from the drop down dialog menu select background tab, then from color scheme tab, change the sceneview background to desired color (from lighter to darker), to be able to see the model, see Fig. 1 below.

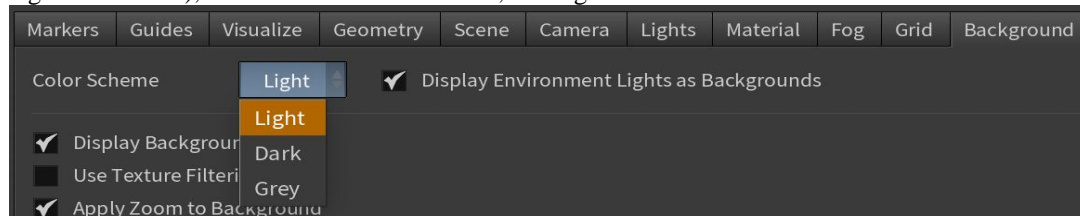


Figure 1How the sceneview background color is modified for better visualization of objects.

3. Sceneview Pixel Point Size

Due to enormous amount of display in sceneview, sometimes the images displayed on the sceneview 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 sceneview, 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.

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 posup the parameter pane subordinate to network for editing attribute values. Typing “p” in the network pane activates/deactivates the parameter pane, see Fig 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 3.

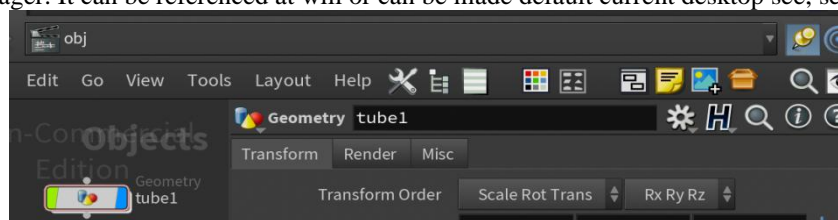


Figure 2Parameter for network tube object is activated or removed or with pressing “p” in the networkpane.

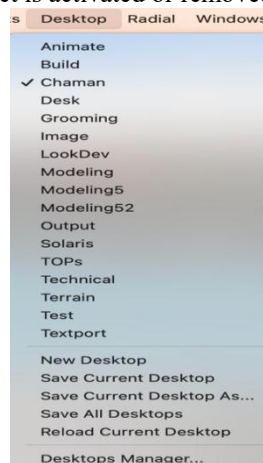


Figure 3Desktop manager in toolbar desktop to create custom desktop, e.g. Chaman appears in the DDL

III. Magic of Houdini Grid Node

In Houdini, everything is a node, whether it is data or operation. A node can be a class, an attribute, a data object, and 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, you have 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.

A node can be created in several different ways: 1. 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, (3) 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 text field 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, see Fig 4.

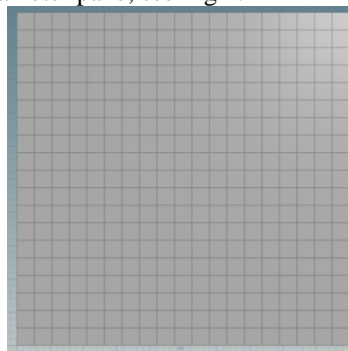


Figure 4 Grid to be used as basis for all special effects

1. Node Processing for creating custom objects

Data nodes use transform node explicitly or implicitly to update display properties. Most of the time, transform node is used, and as a consequence, implicitly built in the other nodes. The attribute values are edited by “Type in” value or “copy and paste relative reference parameters”, explicitly visible thumb slider, implicit slider (ladder slider), reference parameters, or dynamic values using some function involving frame number \$F corresponding to animation playbar.

1. Evergreen nodes

Three nodes *transform*, *font*, *color* and *file* are frequently used for various reason. The font node is used to display 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. It is useful to identify the resource developer. Color is used to color objects. Color parameters type constant, and class *primitive* are sufficient. The file node is used to include the description and logo pertaining the organization in scene view. For any file location reference, a click on arrow on the right of textfield in parameter pane brings DDL of options, scroll the directory path to filter the path to where the file is. Like font, apply transform to place the logo 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 custom visualization, one can transform the camera view depending on scene view layout. It is preferable to stay in the current desktop view during the whole process. Normally, any changes may be made in parameter pane rather than scene view.

2. Art of Grid Node

In Houdini, some support nodes are automatically created implicitly by the software to free the user from micromanaging the nodes. It usually happens when solver nodes are created. In this paper, grid and essential additional nodes, box, circle, sphere, tube, curve, line are used to illustrate the Houdini grid applications. They are used to create hybrid simple objects: table, plates, pipes, stairs, building, OPS: growing grass, coloring, tree creation, wind effects, cloud, fracturing, rain, smoke, fire, fracturing effects. Fundamental OPs nodes: commonly used nodes are: transform, polyextrude, Copy, revolve, sweep, bend, Merge-boolean-switch, popnet, pyro, rigid body dynamics. Pasting images on 3D object is accomplished by uv Quickshade node, and color. Boolean node parameter Shatter breaks objects without the need to go to rigid body object fractures [3].

3. Grid and Patterns

This section describes generation of artistic patterns from the Grid node. The *Patterns* use polyextrude and its *inset* parameter to create artistic patterns and window, see Fig 5,6. *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. The tops of walls are with glass like *fractals*, that is accomplished with using *indexing* primitives and *grouping* to polyextrude with parameter labeled *distance*.

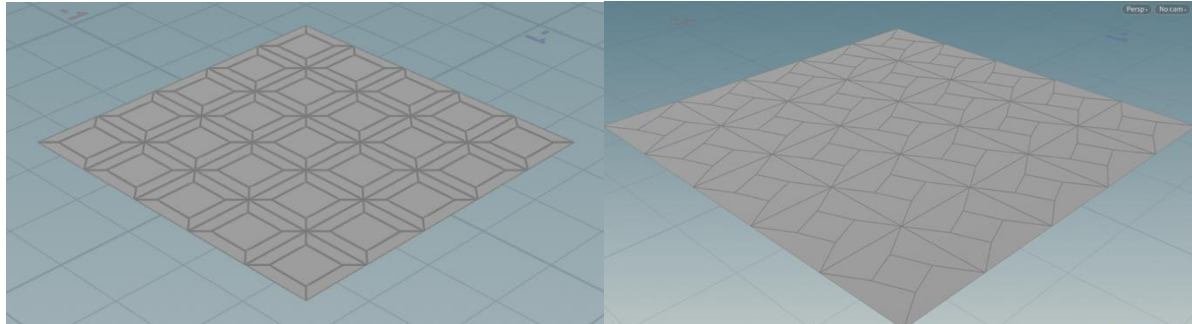


Figure 5. Grid pattern with polyextrude parameter *inset* only Figure 6. Grid pattern with polyextrude parameter *inset* only

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 Fig7,8

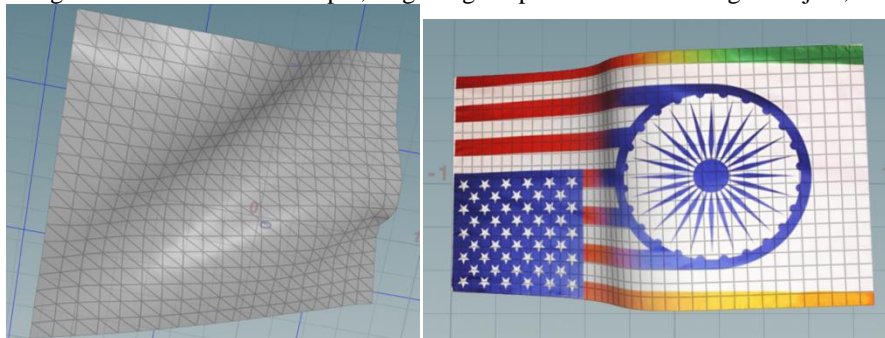


Figure7. grid displaying the cloth effect, Figure8. grid with uvquickshade map posted on the cloth effect

5. Artistic Buildings Grid and Highrise Structure

A grid node is created and then a poly extrude 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 high rise building, in the second case polyextrude *twist* option creates the effect of twisted towers, see Fig.9,10.

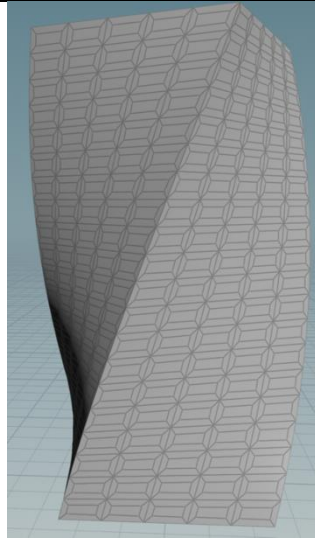


Figure 9 grid is polyextruded with *distance* parameter, used connected component, another polyextrude is used to create a pattern on the walls.

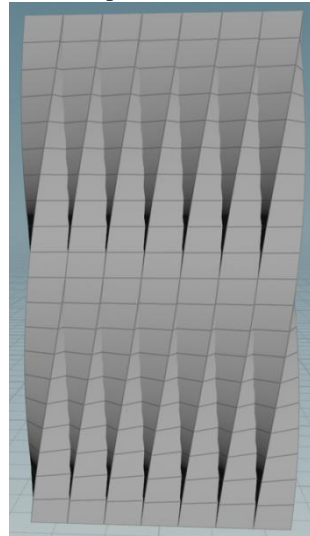


Figure 10 grid is polyextruded with *distance* parameter, used individual elements

6. Furniture- Grid Table

Grid node is used to create a rectangular table. The grid is polyextruded to create table top, a scaled grid is polyextruded to create a leg, leg is copied to select positions with the top to make it look like a custom table, see Fig.11. CopyToPoints node is used to accomplish this task.. The grid and polyextrude node can be used to create other fancy tables.

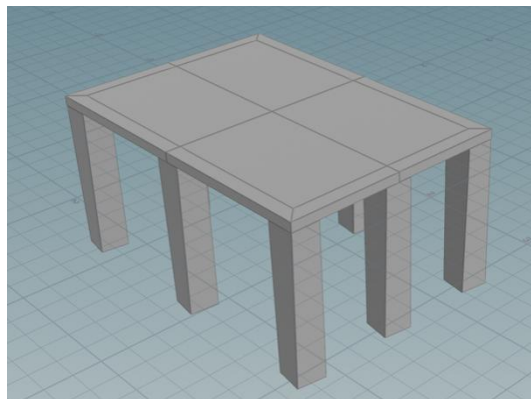


Figure 11 3x3 grid table top, with a leg copied to eight grid vertices excluding the center grid point.

7. Animation with Grid

Animation is the trademark of virtual reality. Mostly, animation is built in Solver nodes. Custom animation is created with \$F and associated functions for some attribute value to create special effects. 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. We use frame number for \$F on color values for animating colors. In Houdini, solvers provide the animation on all kinds of actions. Polyextrude with *individual elements* and *inset* only creates appearance animation. The color reflects through the windows to create color animation effect through holes in the walls, see Fig. 12.

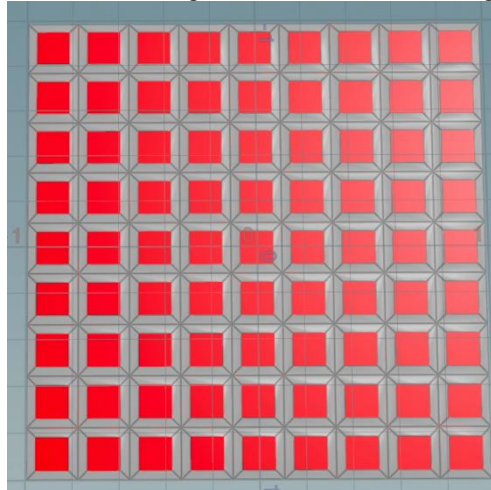


Figure 12 Polyextrude with *individual elements* and *inset* only creates appearance animation.

IV. Nature

1. Brick with Grid: Design and Construction

For construction of Brick objects with grid, polyextrude-*distance* - parameter is mostly used. For example, bricks, house structure are created with polyextrude distance parameter only [4]. Here brick is constructed from grid, polyextrude-*distance*, color-*primitive*, *constant*, values of nodes, see Fig. 13.

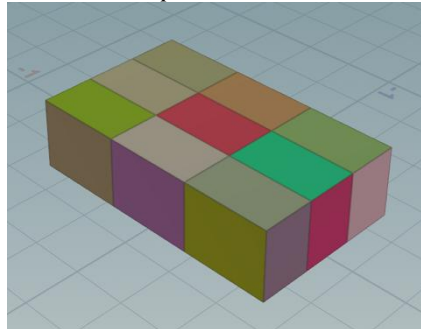


Figure 13 Brick from grid, polyextrude: *distance*, color: *primitive*,

2. GridHouse 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. 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. Poly extrude is used on grid to create building, another poly extrude creates the roof. In the second house door is created differently from grid and colored, see Fig. 14, 15.

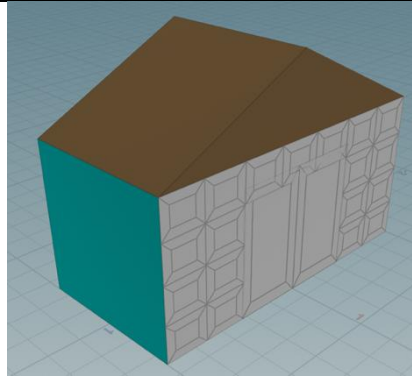


Figure 14 House from grid, polyextrude: distance, inset, transformExtrude Front, entrydoors

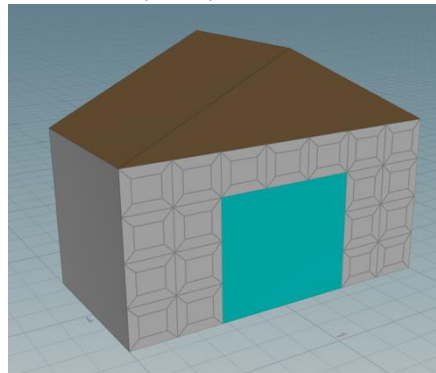


Figure 15 House from grid, polyextrude: distance, inset, transformExtrude Front, entry colored

3. Grid and Polyextrude

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

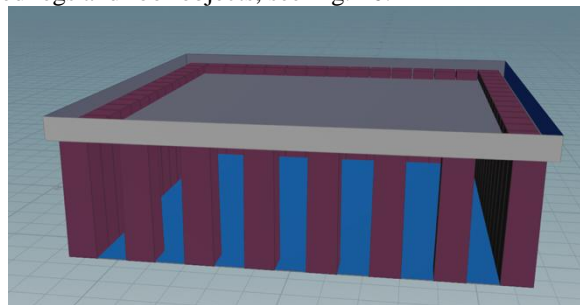


Figure16 Grid, transform, andpolyextrude:distance

4. Grid and Colony of Houses

From grid we create a house as in V.2 in the paper earlier, 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* copytoPoints copies, see Fig.17. *Circular*, copyto 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), see Figure 17. grid and copy to points on a rectangular grid Figure 18. grid, first floor, and copy to pointe vertically Fig.20. Then add tube in the center.

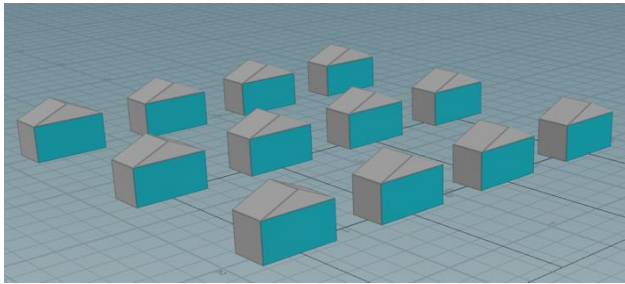


Figure 17 grid and copy to points on a rectangular grid

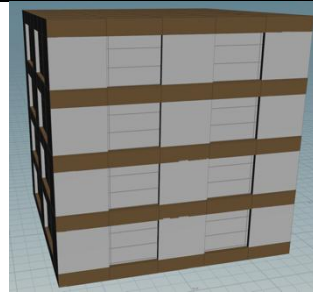


Figure 18 grid, first floor, and copy to points vertically

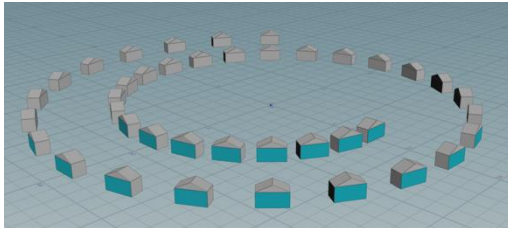


Figure 19 Spiral from grid, transform and copy (offset z)

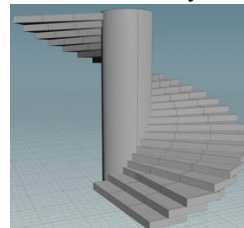


Figure 20. Stairs from grid, transform and copy (offset y)

5. 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, 22..

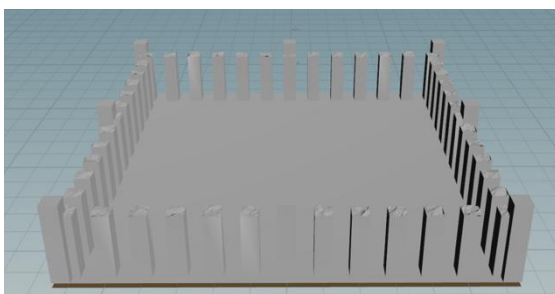


Figure 21. Playground with mid wall pillars and open entry walls

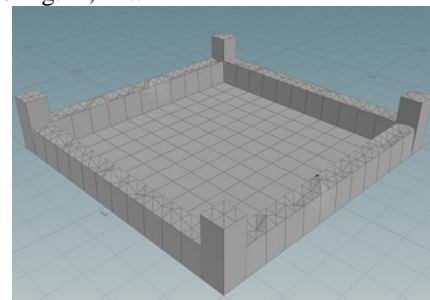


Figure 22. Playground with close entry walls

V. Natural Phenomena

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 userfriendly 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 blades is created, it is copied with copyToPoints 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[5]. Wind effect is created with controlled rotation about x, y, z- axes. Grid and grass effects follow in figures 23-25.

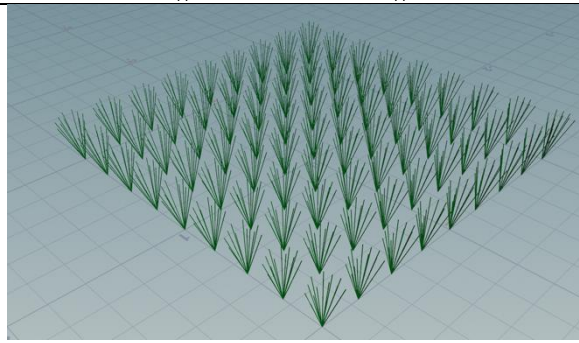


Figure 23 Green grass with distinct grass blades with visible wind effect.

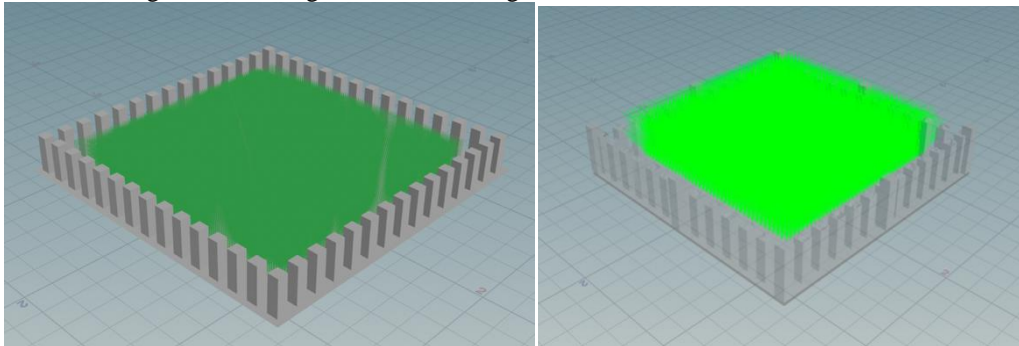


Figure 24 Green grass dense in the playground Figure 25. Dense Green grass in playground with middle pillars

2. Grid on Fire Popnet

Popnet node is used to create fire effect over the whole grid or at the selected primitives, see Fig. 26-27.

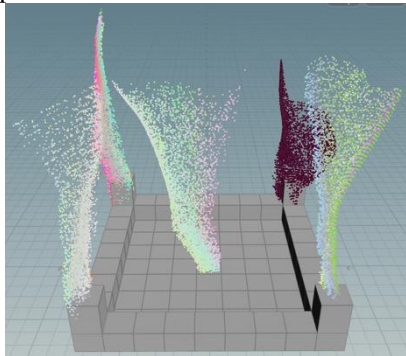
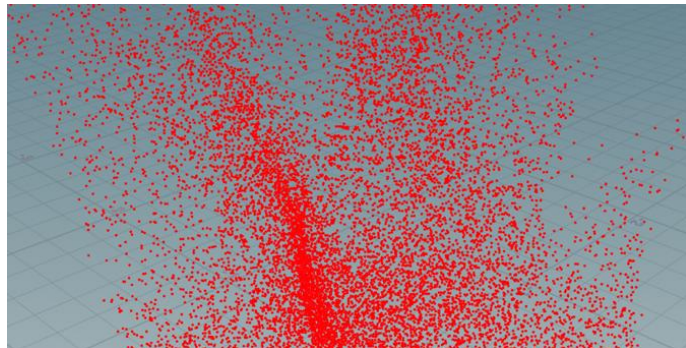


Figure26.Gridand Popnet Lighting Effect Figure



27 Grid and Popnet Fire Effect

3. Grid and Rain Rain with popnet

If particles are not visible, see Fig. 28, they can simulated for view with little spheres. Or from sceneview, type, d and go to geometry and increase the pixel points size to maximum, e.g. .d→geometry→pixelpoint



Figure 28 Grid and Popnet for special Rain effect

4. Grid and Data Noise

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.

5. 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. 29. 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.30-34.

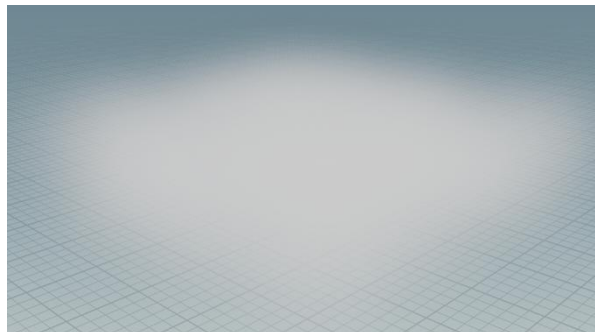


Figure 29 Grid and Isooffset node creates fog effect. It is useful to create smoke, sky and clouds effects also.

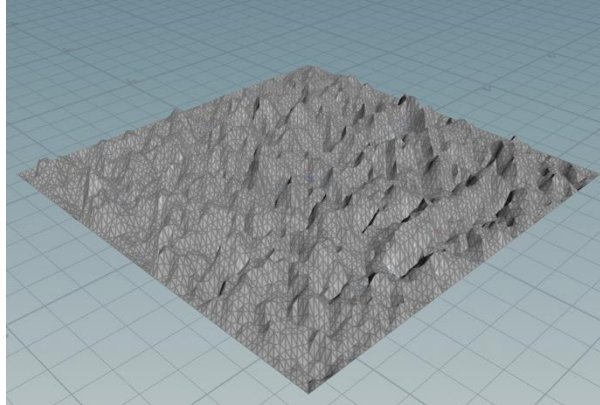


Figure 30 Mountain and FractalNoise effect on grid

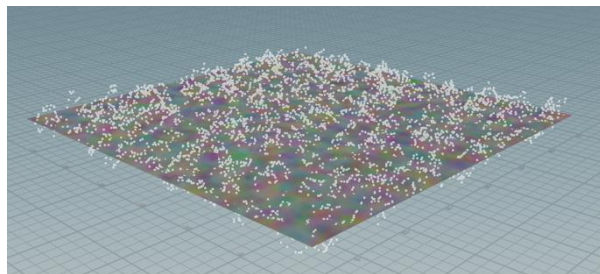


Figure 31 Grid and Attribnoise effect on grid

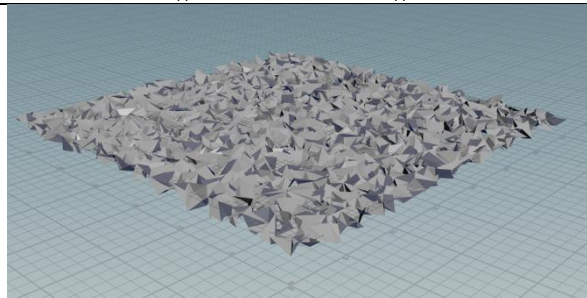


Figure 32 Grid and Point Jitter effect on grid

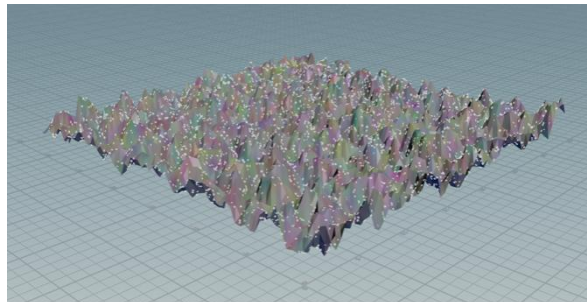


Figure 33 Grid, Mountain and attribnoise effect on grid

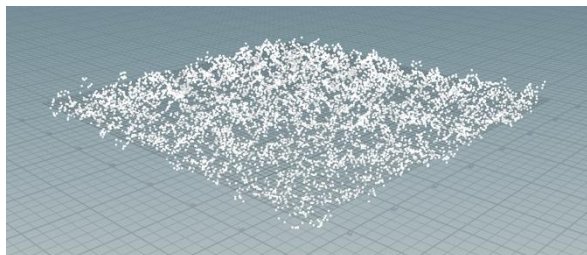


Figure 34 Grid and Snowy mountain effect on grid

6. 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. 35-36. Other parameters include time- speed.

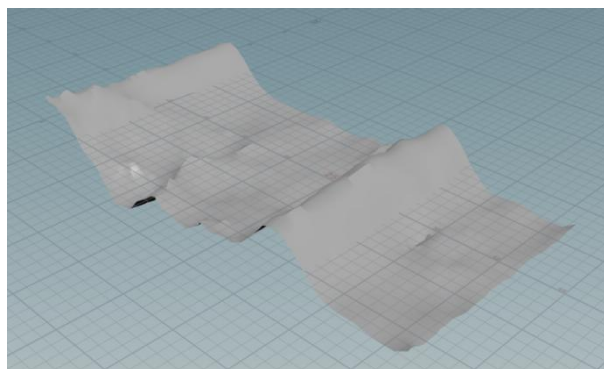


Figure 35 High intensity wave with more points, high intensity waves with larger down sample

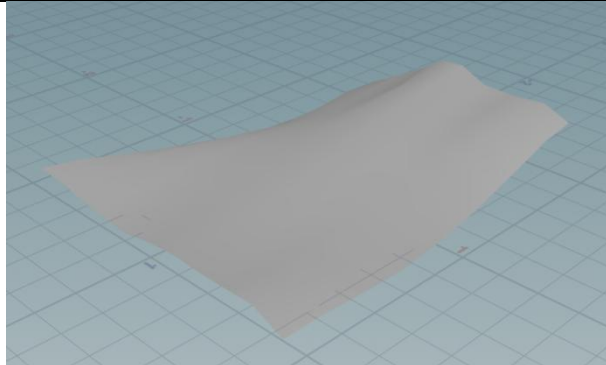


Figure 36 High intensity wave with less points, low intensity waves with smaller down sample.

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 37, 38..



Figure37.OceanWave Effect on an image.

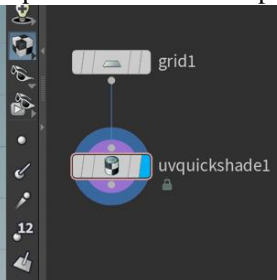
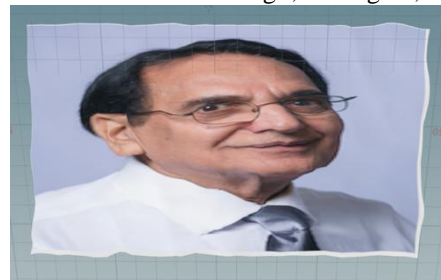


Figure 38. Grid node displaying Author's depiction and title



6.1Grid 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. 39.

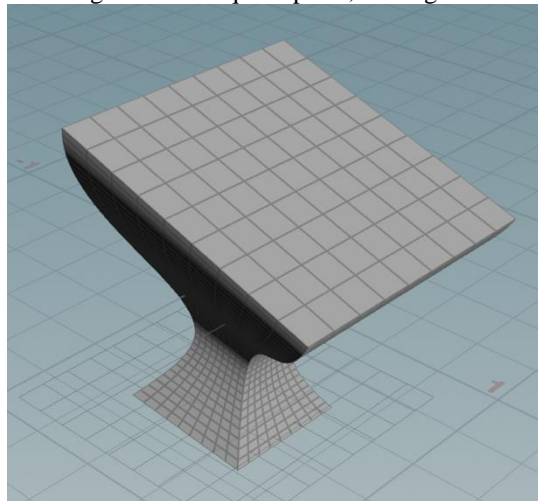


Figure 39 Grid-polyextruded and Bend *bend-taper-squish*

VI. Shattering, Fragmentation, Destruction

1. Fracturing

Grid can be a foundation to create 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 popnetnodes. A simple example follows.

2. Grid Shatter-Boolean

Here grid node is created and 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 is built in the Boolean node. If we use that operation, then it separates them into three parts: A-B, B-A, and $A \cap B$, see Fig. 40-41.

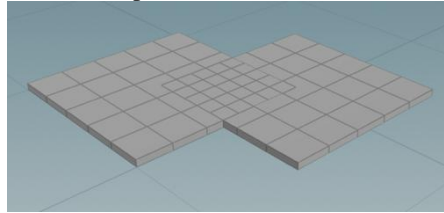


Figure 40 Boolean node with shatter and intersection both display the fragments, A-B, B-A, and $A \cap B$.

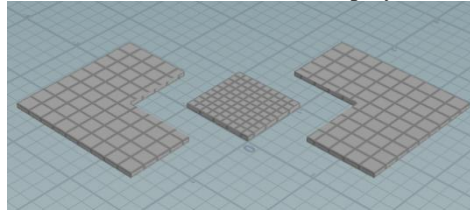


Figure 41 Boolean node with shatter intersection displaying fragmented pieces

3. Grid and Fracturing

Easiest way to fracture and view the fractured grid is to shatter it into primitives and use explodedView to view the fractured pieces. Shatter node creates Voronoifracture where shattering occurs[6]. Voronoifracture 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.42-44.

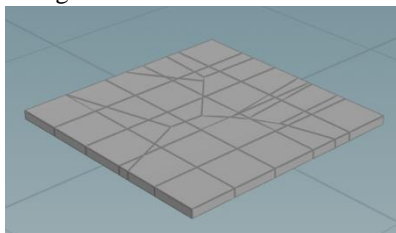


Figure 42 Voronoifracture grid,

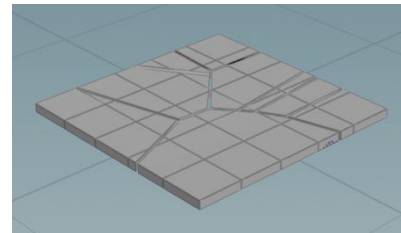


Figure 43 Fracture cracks

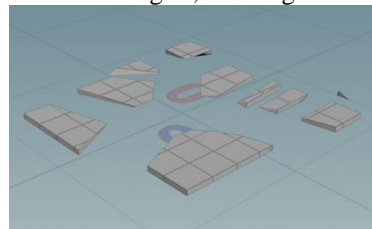


Figure 44 Vornois Fracture grid, Fracture cracks, Fractured pieces.

4. Grid, Volcano and Hurricane

It is incredibly easy to create volcano effect from popnet. We used grid for generating particles to be blown away with windforce. The windforce node is used in popnet to set the direction flow of particles and force of wind blowing. Here grid is extruded with desired *distance* and *inset* parameter values to create volcano mouth. Popsolver animates the volcano for visualization node. Windstorm effect is created similarly by changing the windforce direction sideways. Extra spheres of varying size are floated to give the effect of storm, see Fig.45-46.

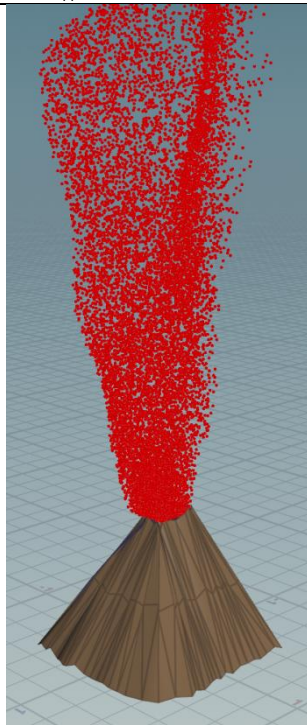


Figure45 Grid and Volcano

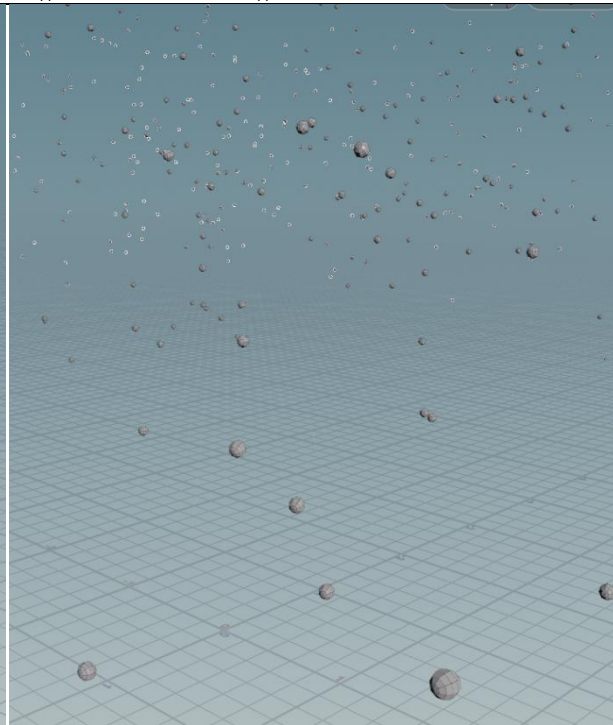


Figure 46 Grid and Storm special effect (Popnet)

5. Grid and Pyro

Pyro node is supported with several destruction nodes. One such nodes is sparse Fireball which give the of ball of fire, see Fig. 47.

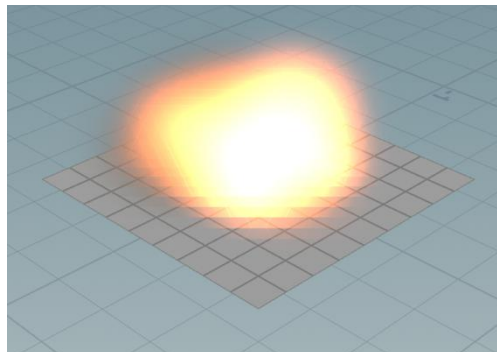


Figure47 Grid and pyro sparse fireball

6. Grid and Rigidbody Dynamics

Beside the exploded view, there are other animation tools for fracturing 3D objects. Two of the geometry based nodes arerbd Material fracture and rbd Bullet Solver. Two of the obj level nodes rbd Fractured, and autoDop Network.

VII. 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 mechanism to create Houdini Digital Assets at object level as well as geometry level[7]. HDA is created by right mouse click on the node under consideration, and following a simple sequences of self documenting dialog boxes. The HDA is installed by double click the HAD[8]. 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. The HDAs can be used as the other shelf tools in Houdini. For example, something are just for display, e.g, developer's title and depiction[9].

VIII. Conclusion and Future Work

In this paper we have shown the how virtual reality Houdini SideFX node based procedural modeling can be used to generate various interesting shapes and patterns.

We have given enormous examples illustrating grid node practices. Sections illustrate 1. *Houdini Sidefx Basics*: a. First User Interface b. Sceneview Background, c. Sceneview Pixel Point Size, d. Pane Management; 2. *Magic of Houdini Grid Node*: a. Node Processing for creating custom objects, b. Evergreen nodes c. Art of Grid Node d. Grid and Patterns, e. Grid and Cloth, f. Artistic Buildings Grid and Highrise Structure, g. Furniture- Grid Table, h. Animation with Grid ; 3. *Nature*: a. Brick with Grid: Design and Construction, b. Grid House Construction, c. Grid and Polyextrude, d. Grid and Colony of Houses , e. Grid and Playground; 4. *Natural Phenomena*: a. Grid and Grass animation with wind effect, b. Grid on Fire Popnet, c. Grid and RainRain with popnet, d. Grid and Data Noise, e. Noise and Visualization special effects, f. Grid and Ocean Waves, g. Grid and Bend; 5. *Shattering, Fragmentation, Destruction*: a. Fracturing, b. Grid Shatter-Boolean, c. Grid and Fracturing, d. Grid, Volcano and Hurricane, e. Grid and Pyro, f. Grid and Rigid body Dynamics.

In the coming paper we will show that same grid can be used to create special effects of rain, hailstorm, smoke, fire, earthquake, natural disaster hazards, and hurricane special effects. We hope that reader will enjoy watching the screen shots and get incentive to envision challenging models for education and entertainment.

IX. References

- [1]. Yilmaz, B., & Goken, M. (2016). Virtual reality (VR) 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>
- [2]. 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]. 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]. Sherman W.R., & Craig A.B., (2003). *Understanding virtual reality-interface, application and design*. The United States of America: Morgan Kaufmann Publishers.
- [5]. Simon Verstraete *Houdini Digital Assets, introduction to HDA*, December, 2021, <https://www.youtube.com/watch?v=ybk8XWBnO4o>
- [6]. Weidlich D., Cser L., Polzin T., Cristiano D., & Jik er H., Virtual Reality Approaches for Immersive Design, *CIRP Annals - Manufacturing Technology*, 56(1), 139-142.
- [7]. Xiaodong W., Dongdong W., Yue L., & Yogtia W., , Teaching based on augmented reality for a technical creative design course, *Computers & Education*, 81, 221-234.
- [8]. 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.
- [9]. Nee A.Y.C., Ong S.K., Chrysosolouris G., & Mourtzis D., , Augmented reality applications in design and manufacturing, *CIRP Annals-Manufacturing Technology*, 61(2), 657-679.