

Cubism with Houdini

by Kader Djehaf

<http://daek.online.fr>

--- I. the Points Cloud ---

1. run Houdini. 1 Geo OBJ node, 4 Light nodes are already there (3 point lights and 1 ambient light). if not, then press <TAB> and chose a Geometry.

A new Geometry node called geo1 is created with a default geometry in it.

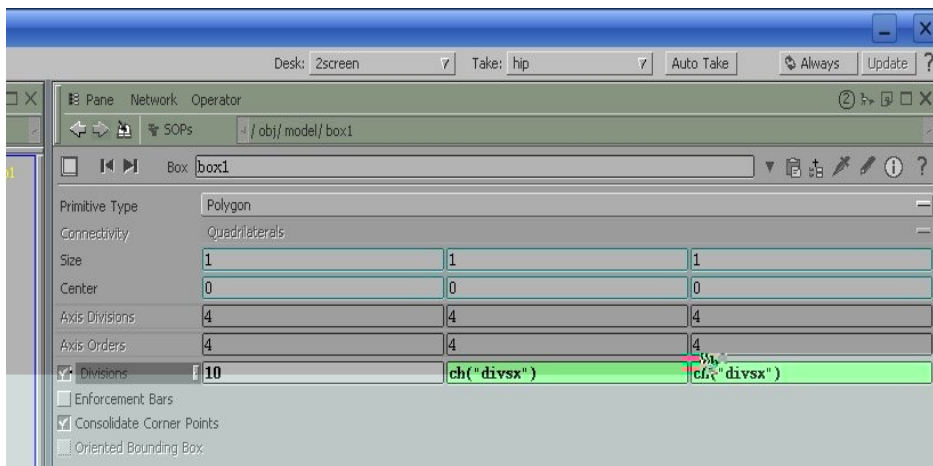
2. select model/geo1 node and rename it "PointsCloud" en press Enter to go in SOP level

3. remove the default geometry if needed.

4. press <TAB> key and lay down a Box SOP. Rename it Cloud_Base. Set Primitive Type to Polygon and size to 1, 1 and 1

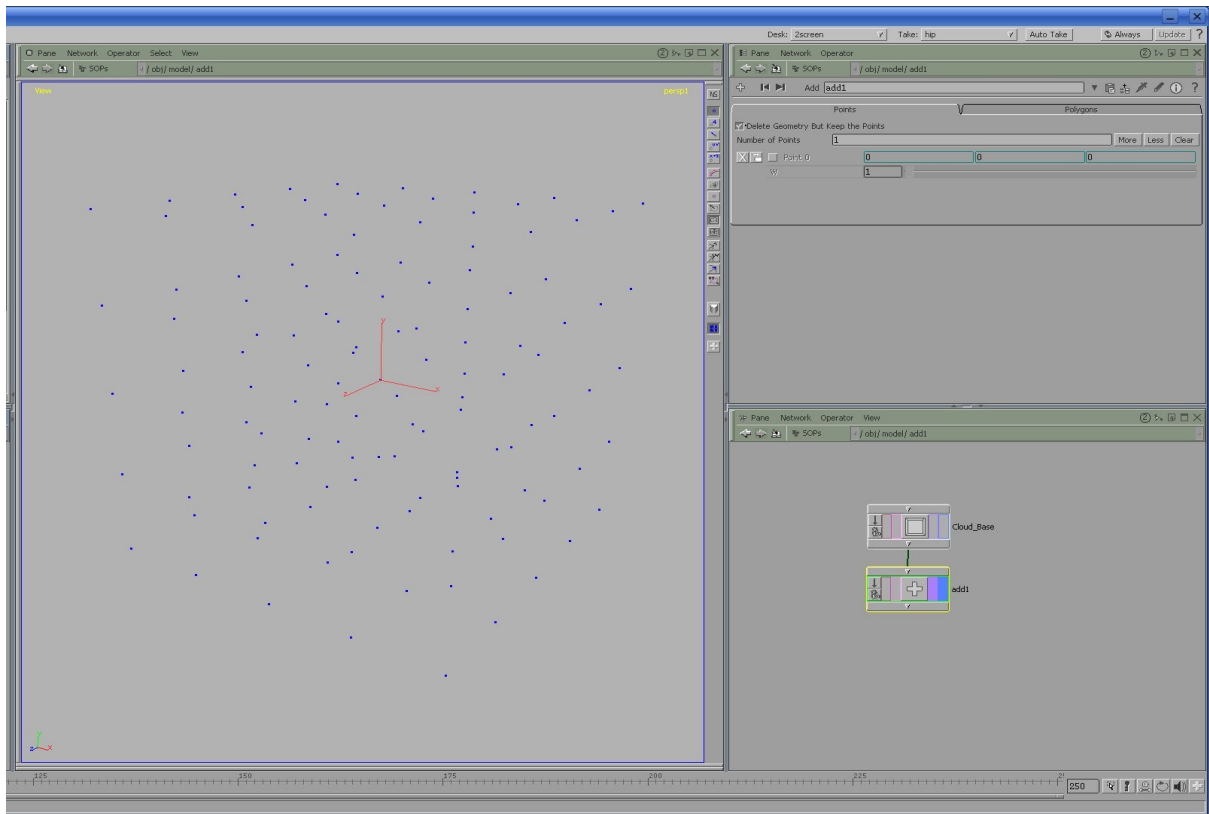
5. turn on Divisions and set Division X (divsx) to 10.

6. in Division Y and Z write the following expression: `ch("divsx")` or right click in the divsx field and chose "Copy Parameter". Right click in divsY and divsZ and choose "Paste Copied relative Refs". It will create the expression for you.



7. Press <TAB> and choose Add SOP

8. Turn on Delete Geometry but keep the points. If you can't see the points, turn on the points button in the viewport.



Now, we going to import a Geometry in BGEO Format. You can Download the model on my website. bunny.bgeo or Armadillo.bgeo These Stanford University scan models. I converted them from they original PLY format to BGEO using the "gply" command
You can use any other model or quickly model a random shape for this tutorial.

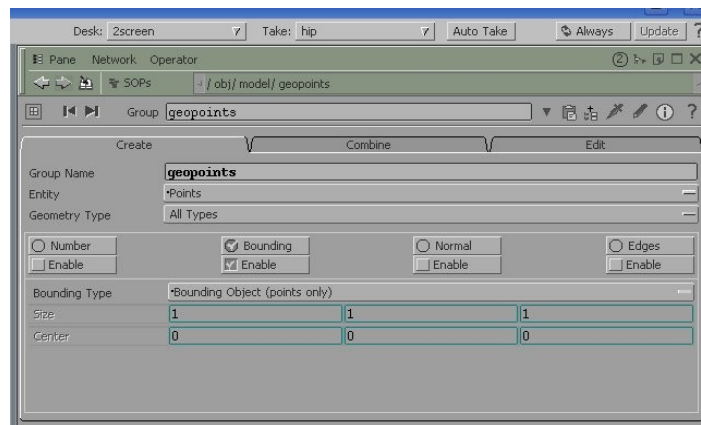
9.Press <TAB> and lay down a File SOP a Group SOP and a Delete SOP.

10. Connect the Add SOP the the first input of the Group SOP and connect the File SOP to the second input of the Group SOP.

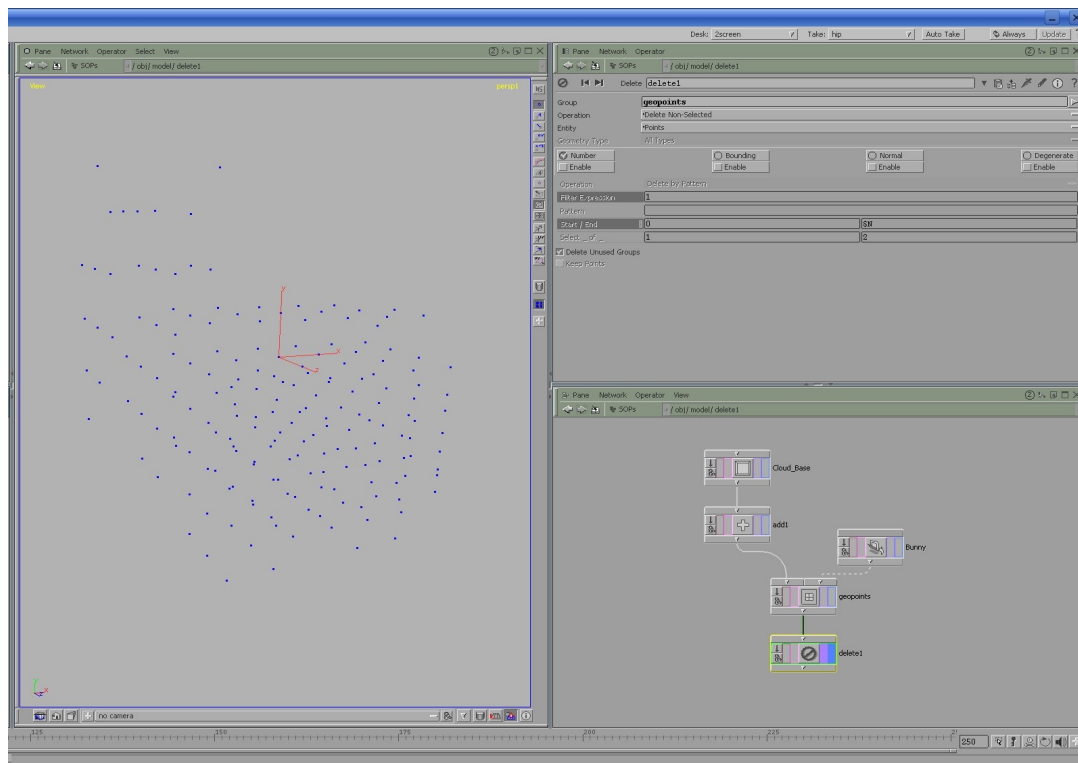
11. In parameter of the Group SOP, change the Group Name to geopoins and Change Entity to Points.

12. turn on Bounding and Enable (just below). In bouding box type, choose "Bounding Objects (points only)".

The points inside the geometry are selected. We will delete the unselected points with the Delete SOP.



13. In the Delete SOP parameter, write "geopoints" in the Group field. Set the Operation to "Delete Non-Selected" and Entity to "Points".



--- II. Points Instancing ---

1. lay down another Box SOP. Rename it "Base_copy".

We want Base_Copy box and Cloud base box to be the same size. To make sure they always the same.

2. In the Cloud_base parameter right click on Size and choose "Copy Parameter". In the Base_copy parameter right click on Size and choose "Paste Copied relative Refs".

3. press <TAB> and lay down a Transform SOP and a Copy SOP. Rename the Transform SOP to resize_Box.

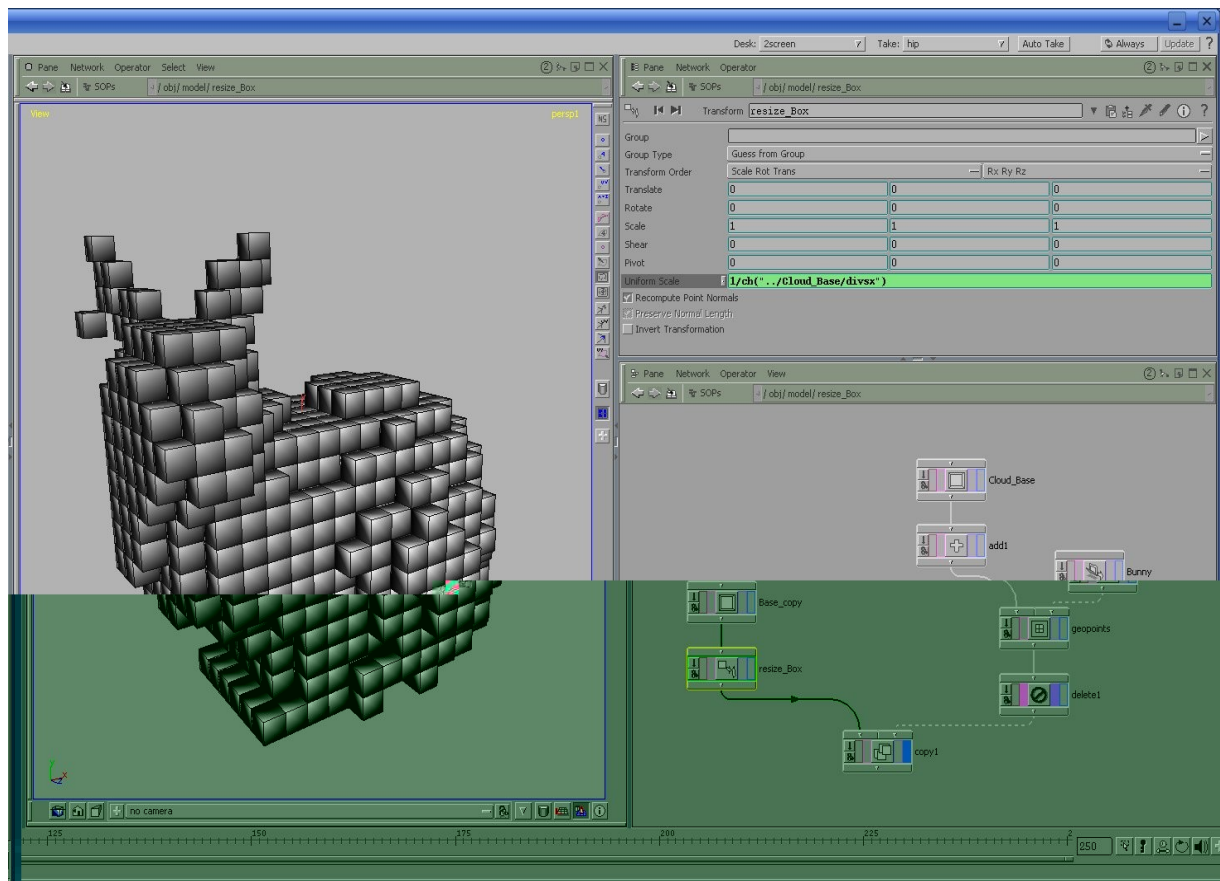
4. Connect Base_copy to resize_Box and the resize_Box to the first input of the Copy SOP.

5. Connect the Delete SOP to second input of the Copy SOP.

We need to resize our Base_Box. We are going to use an expression. No matter the value used in the Cloud_Base Divisions parameter the box will always be the right size.

6. Select resize_Box and enter the following expression in the Uniform Scale parameter: $1 / \text{ch}("../\text{Cloud_Base}/\text{divsx}')$.

If you now change the Division resolution of the Cloud_Base, the copied boxes will be resized.



I is very nice to visualize the copied boxes in the viewport. It is good enough for visualisation but can become redundant if you want to use a more complex object rather than just boxes. Even with Boxes, as soon as we start increasing the Division resolution, and depending on your machine, your computer is going to slow down.

To solve that problem, we going to use a very useful SOP: the Point SOP. The boxes will be visible only at render time.

7. press <TAB> and lay down 3 Null SOP and a Point SOP. Connect `resize_Box` to the first Null SOP and rename it "BOX". Connect the Delete SOP to the second Null SOP and rename it "POINTS" (I use upper-case as a convention with Null SOP). Connect POINTS to the second input of the Copy SOP. Rename the Copy SOP "CopyVIZ".

Connect the output of Bunny (File SOP) to the 3 Null SOP and rename it MODEL_VIZ. Connect MODEL_VIZ to the second output of the `geopoints` Group node. From now on, any manipulation of the Bunny will occur before the MODEL_VIZ Null SOP node.

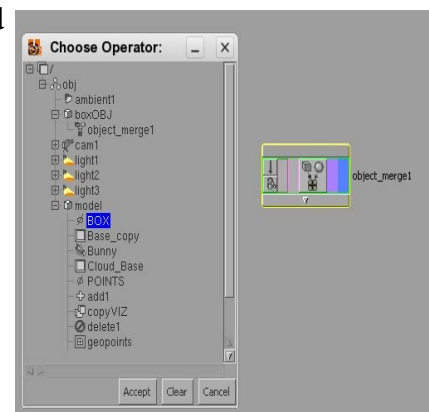
8. Middle click on the output of POINTS and lay down a Point SOP. rename it "instantiate".

9. press "u" to go at Object Level. Lay Down a Geo OBJ Node and rename it "boxOBJ".

Select it and press enter to go into SOP Level. Delete the Default Geometry and lay down an "Object Merge" SOP.

10. Make sure Enable Merge 1 is turn on and click on the "plus" button. navigate into PointsCloud and select the Null SOP named BOX.

11. Press "u" to go back to Object level. Select PointsCloud and press enter to go into SOP Level.

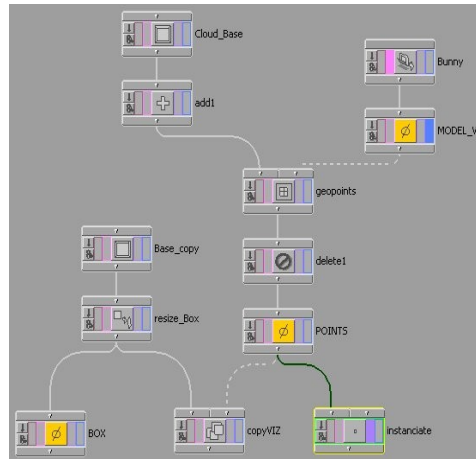


12. Select instantiate Point SOP en choose the Particle parameter Tab.

13. Set instance to "Add instance" and click on the plus button. Navigate and select boxOBJ node. Turn the render flag on of the instantiate node and the display flag of the MODEL_VIZ Null SOP node on

14. Press "u" to go to Object level. Select "pointCloud" Geo OBJ node and chose the Render Parameter Tab. Turn "Point Instancing" on.

You can see the bunny geometry in the viewport and if you render now, you will get the boxes smoothed. But it is not the look we want for our boxes. We want sharp edges. So back into SOP Level, add a Facet SOP between Base_Copy and resize_Box and turn "Unique points" on.



--- III. VEX in action. ---

Now we have a pretty cool effect and we can certainly do a lot with it already.

However I want to give a more organic look to the general shape or maybe rotate the boxes in a random manner. VEX is one way to do that.

In SOP level, lay down 2 VOP Sop Definition. Call the first one VEX KD Random and the second one VEX KD Turb.

-- VEX KD Random --

This Vop will be used to randomise the normals of the points in the cloud. the boxes will be rotated according to the points normal.

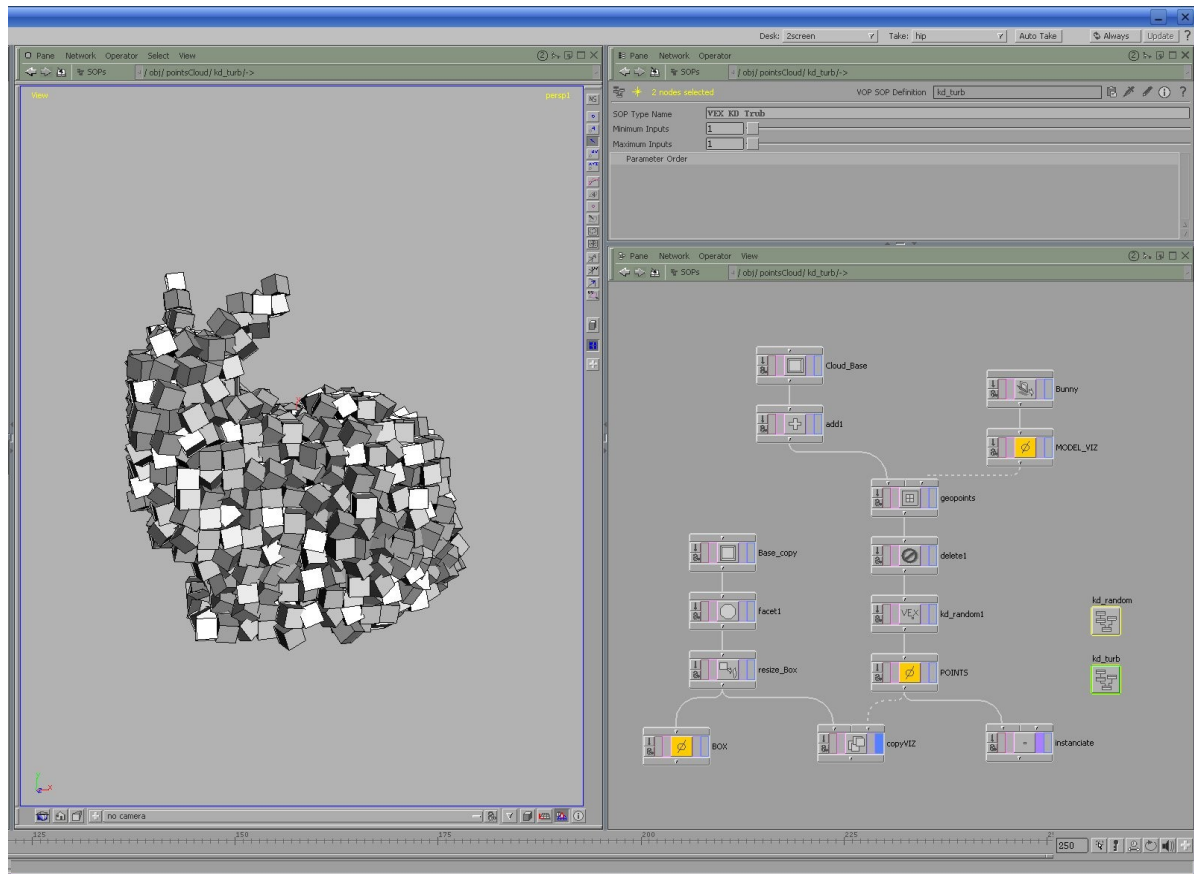
1. select kd_random en press enter. We are now in a VEX Level.

2. press <TAB> and lay down a Global VOP and connect a random VOP to the "ptnum" (point number).

Set the Signature in the Random VOP to "1D Integer Input, 3D Vector".

3. Lay down a Transform VOP and connect its output (xformpos) to the N input of the output VOP.

4. Connect a Constant VOP to the "pos" input of the transform VOP and change its Constant Type to Vector. We are going to set up the initial direction (and intensity but it doesn't matter here) of the



-- VEX KD VORTEX --

... coming soon

-- *Colour and Textures*

... coming soon