

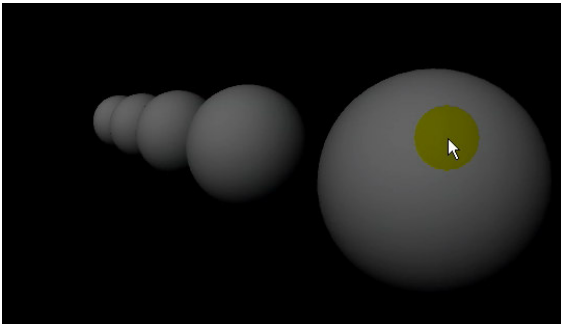
ZDepth and height GrayScale Representation

Nakhle Georges – july 2007

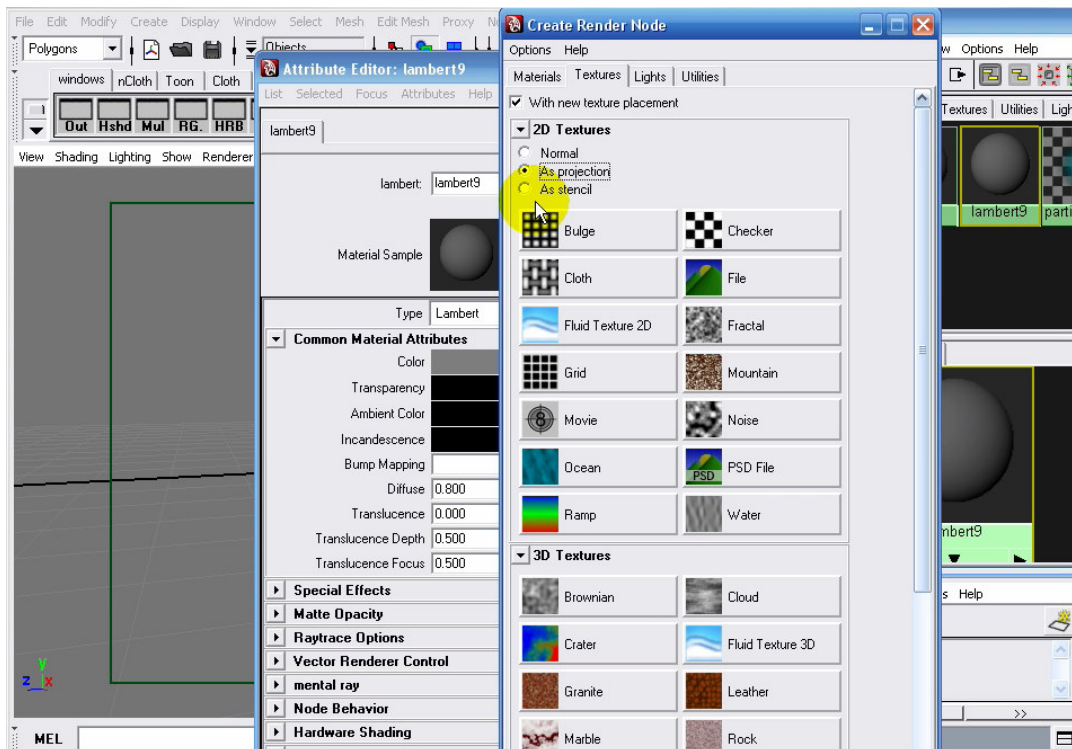
This tutorial shows 3 different methods to create height and depth grayscale representations of a maya scene, which may be useful in many cases such as adding depth of field or fog in post production.

FIRST METHOD

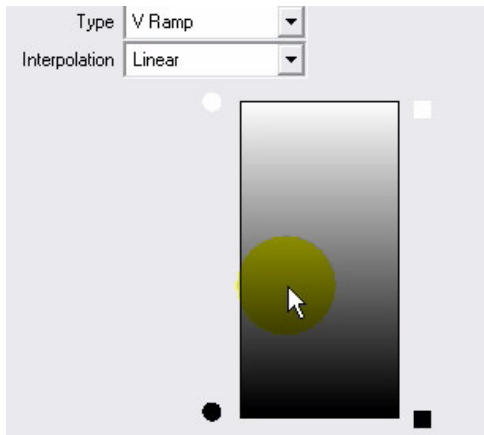
- 1) Create some spheres, and post them on different distances from the camera. Assign a material to the spheres, such as a simple lambert material (**to make the result independent from the lights in your scene, use in all the methods a surface shader**). When rendering at this stage, all the spheres have the same color, independently from the distance from the camera.



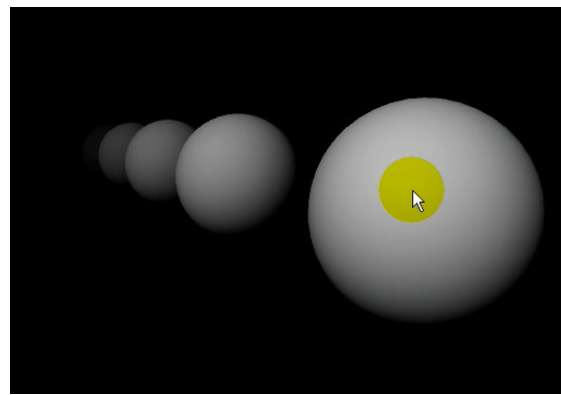
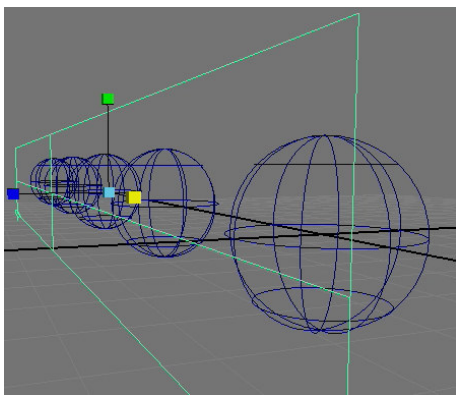
- 2) Go to the Attribute Editor of the material, and map its color with a ramp. In the Create render node dialog, make sure that As Projection is selected, and choose a ramp.



Edit the projection node by choosing planar as the projection type, and make the ramp a black and white 2 colors ramp.



- 3) The projection icon is now visible in the viewport, as a small rectangular shape. Scale that icon to fit your scene entirely. Take a render of the scene. The color of the spheres must now be fading towards black. If it is not the case, you will have to change the ramp type from U ramp to V ramp or vice versa.

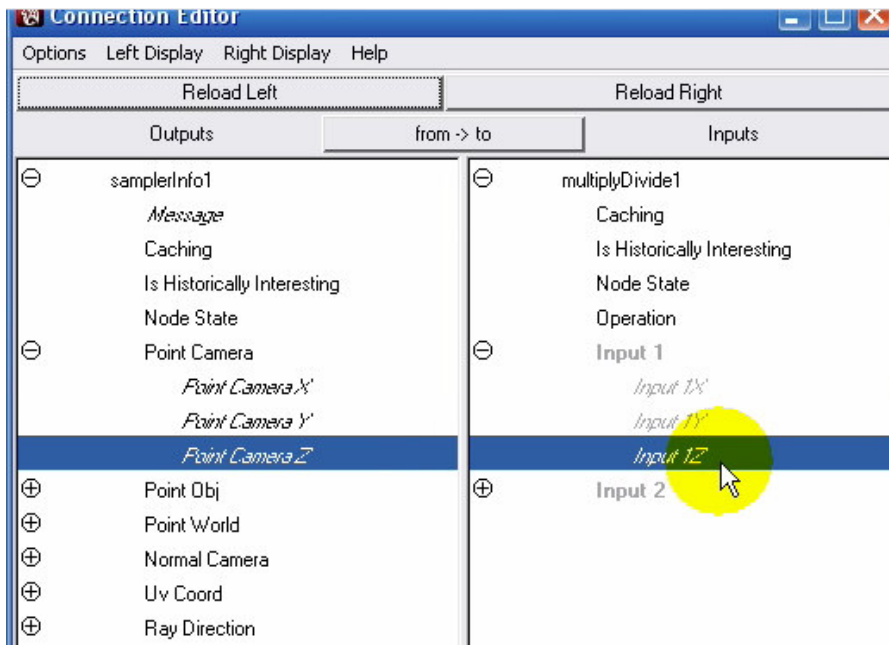


- 4) The 2 main limitations of this method are:
 - a. If you move the camera away from the spheres, their color remains unchanged when rendering.
 - b. If you render the scene from the opposite side, the colors also remain unchanged, so that the closest sphere is dark, while the farthest sphere is clear.

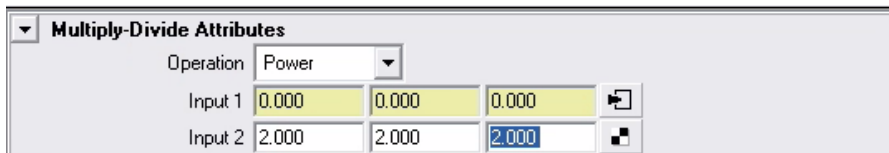
This method is suitable for a static scene, where the place of the camera is already decided, because the result here is static, so it's not suitable for moving cameras. The problem can be solved by parenting the projection icon to the camera, but here another problem arises: the scene might fall out of the range of the projection icon, which will lead to unexpected results.

SECOND METHOD

- 5) Let's check another method to generate the same effect. Here it is calculated based on the distance of the object from the camera, so it's a lot more logical. In the hypershade, assign a lambert material to the spheres, and then create a sampler info node. You will take the point camera attribute, compute the RMS {root – mean – square: $\text{distance} = \sqrt{(x^2 + y^2 + z^2)}$ } value, and then choose a color on a ramp based on that distance.
- 6) For this, your hypershade still open, create a multiplyDivide node. Click on the sampler info node you've created, then shift-click on the multiplyDivide node. Go to the hypershade menu click on window, and choose Connect selected. In the connection editor connect the Point CameraX of the sampler info node to the Input 1X of the multiplyDivide node; do the same for Point CameraY to the Input 1Y and Point CameraZ to the Input 1Z.



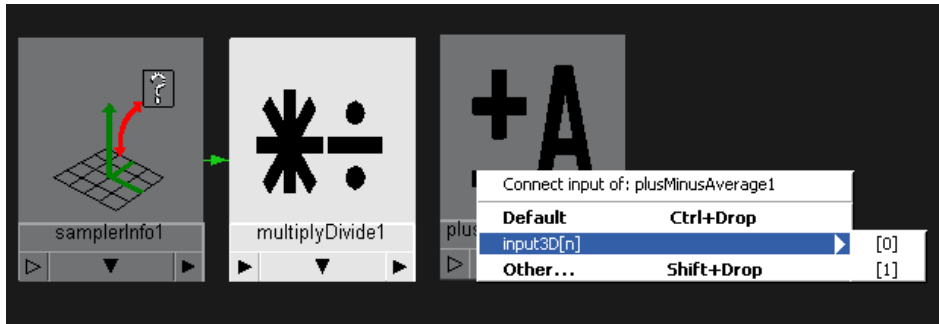
In the multiplyDivide node attributes, choose Power as operation, and set all the 'Input 2' values to 2.



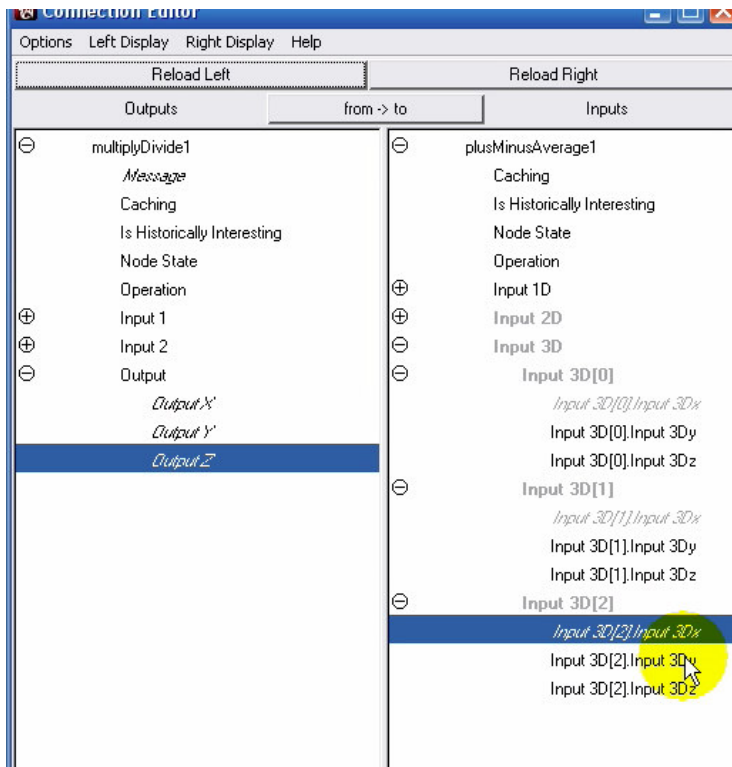
Now to sum up these values, use a plusMinusAverage node. If you load the multiplyDivide into the connection editor, under Input3D section you can't find the inputs to connect to. To make them appear, do the following steps.

Using the middle mouse button, click on the multiplyDivide node and drag it on the plusMinusAverage node, then choose input3D[n] -> 0.

Do this 2 more times to choose input3D[n] -> 1 and input3D[n] -> 2.

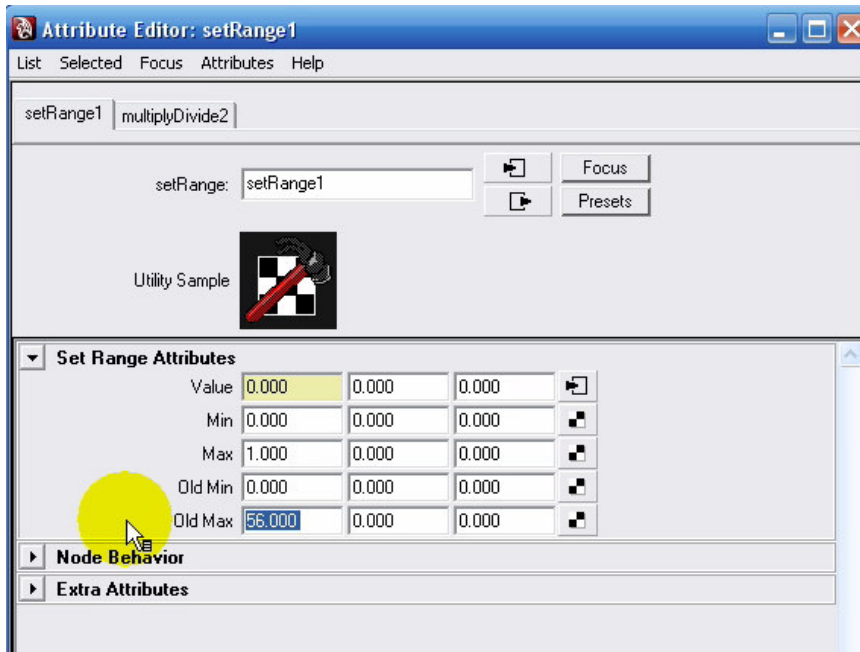


- 7) Now delete the 3 connections you've just created by selecting them and hitting del on your keyboard (remember this trick was done only to make the input[3D] parameters of the plusMinusAverage node appear in the connection editor). Now select them both (the plusMinusAverage and the multiplyDivide), and go to Window -> Connect selected. Connect the OutputX, OutputY and OutputZ respectively to Input3D[0].Input3Dx and Input3D[1]. Input3Dy, Input3D[2]. Input3Dz.

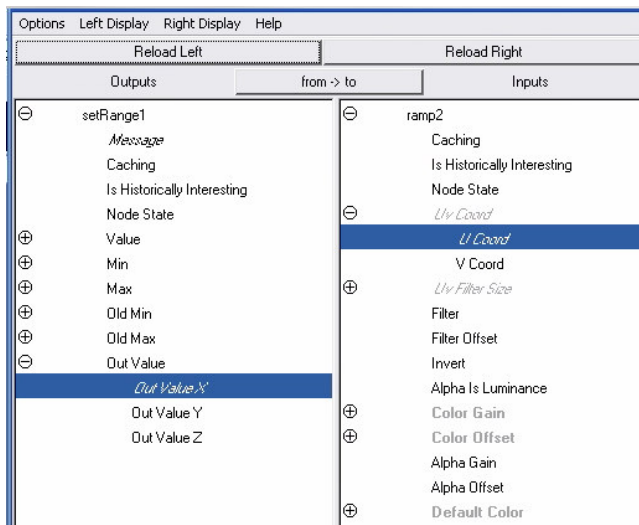


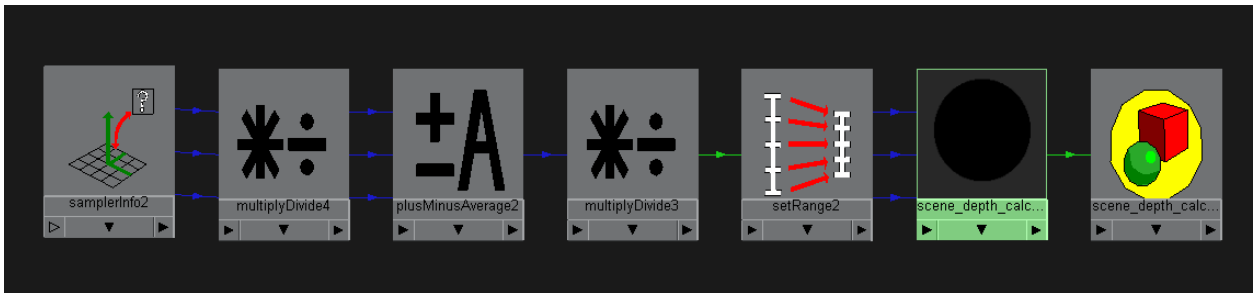
- 8) To calculate the square root of the sum, create another multiplyDivide node, and connect the Output 3Dx of the plusMinus Average node, to the Input 1 of the multiplyDivide node. In its Attributes, choose Power as operation, and set the value of x in 'Input 2' to 0.5.

- 9) The output of the last multiplyDivide node we've created is the distance from the sampled point to the camera. And since the u or v values of the ramp can only take values between 0 and 1. So, create a Set Range node, select the second multiplyDivide node, shift select the Set Range node, and go to Window -> Connect selected. Connect the OutputX of the former, to ValueX of the latter. In the Set Range node Attributes, set the Min to 0, the Max to 1, the Old min to 0 and let the Old max represent the distance beyond which objects become invisible, so in our case, the distance from our camera to the farthest sphere.

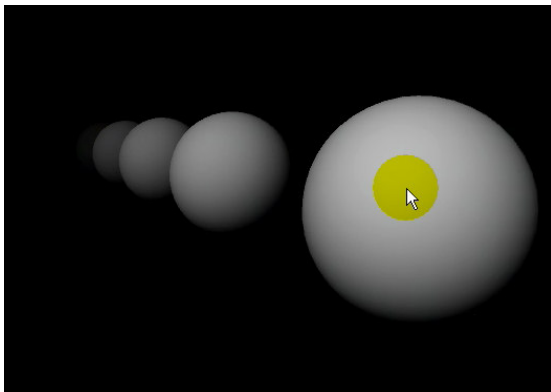


- 10) Now you can use the output of the setRange node to choose between points of a ramp. Map the color of your objects material with a ramp. In the Create render node, make sure that Normal is selected, and choose ramp. Choose black and white for the colors of the ramp. Connect the Output X of the set range node to the UCoord of the ramp node. Make sure that your ramp is a U ramp. (You might need to invert its white and black colors, depends on your case.)





11) This method gets over the problems that we faced with the first one.



THIRD METHOD

12) You can also use scripting to obtain the same result of the second method. For this, you need a sampler info node, and a material. Map the color of the material with a ramp, choose normal, and choose a grayscale ramp as before.

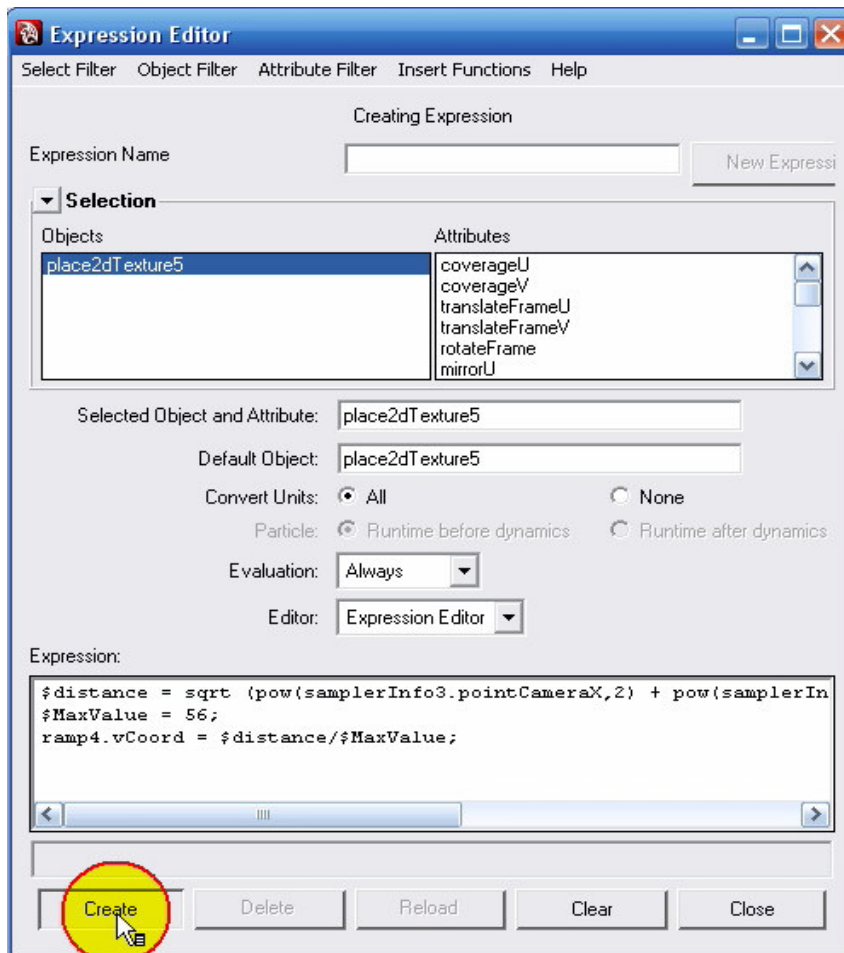
13) In the Expression Editor, write the expression that calculates the RMS, which is the following code line (in my case my sampler info node is samplerInfo1):

```
$distance = sqrt ( pow (samplerInfo1.pointCameraX,2)
                    +( pow(samplerInfo1.pointCameraY,2)
                    +(pow (samplerInfo1.pointCameraZ,2)));
```

Set a maximum value which is equivalent to the old max used in the second method, and divide the distance by this max value, for example let it be 56. Then you will have to link the obtained value the ramp's V coordinate (make sure your ramp is a v ramp in this case).

```
$MaxValue= 56;
ramp4.vCoord = $distance / $MaxValue;
```

Click on create to create your expression, and render the scene. You should obtain the same result as before.



The above methods can serve for many other uses. For example to create a height field, you have to choose the Point world Y attribute instead of the Point camera attribute, inside the sampler info node attributes.

For a video version of this tutorial: http://www.geonak.com/wp-content/uploads/2007/07/ZDepth_and_Height_grayscale_representation_Maya.zip

For any questions or suggestions don't hesitate to contact me at georges.nakhle@geonak.com

Visit my website: www.geonak.com

Regards

NAKHLE Georges