Houdini
3D ANIMATION TOOLS

Houdini
Light, Shade, Render

M07: Mantra Render Engines

Ari Danesh
ari@sidefx.com

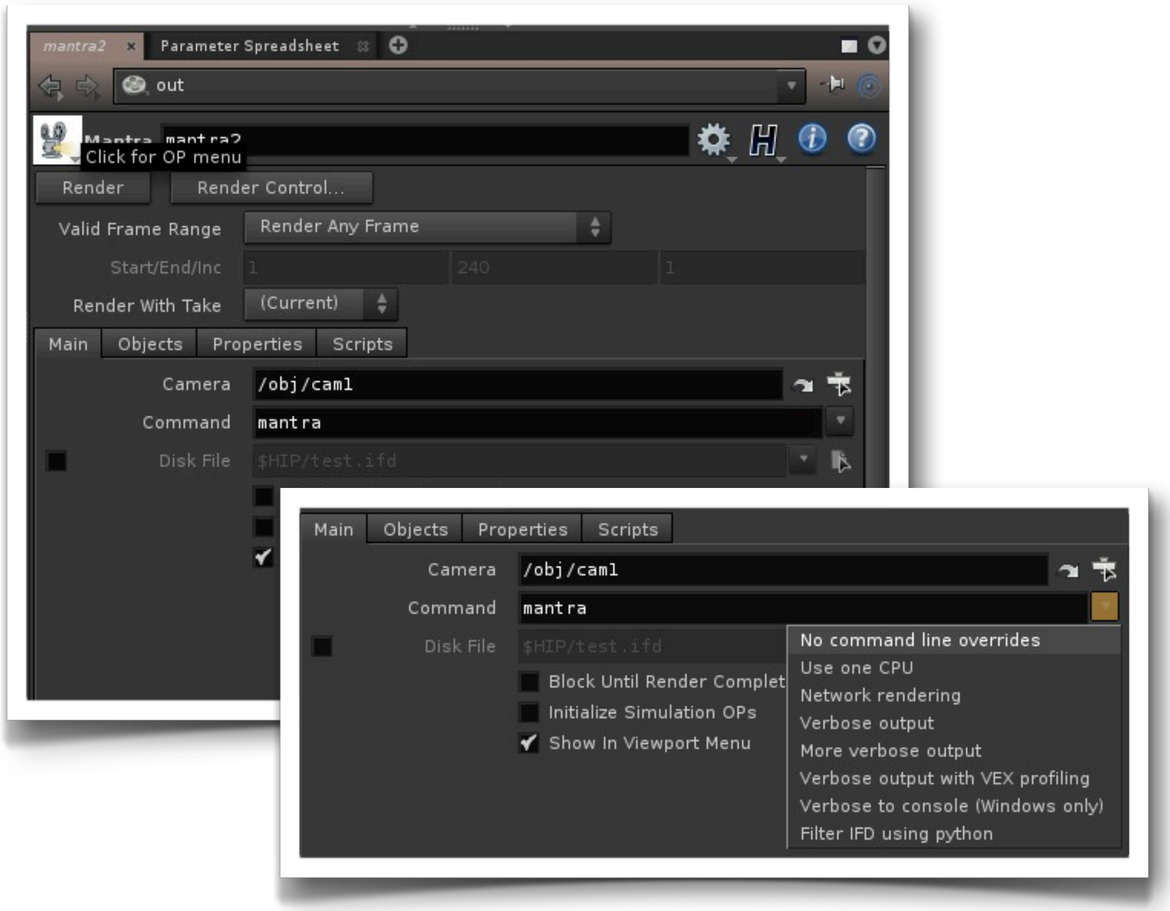
SIDE EFFECTS SOFTWARE

Agenda

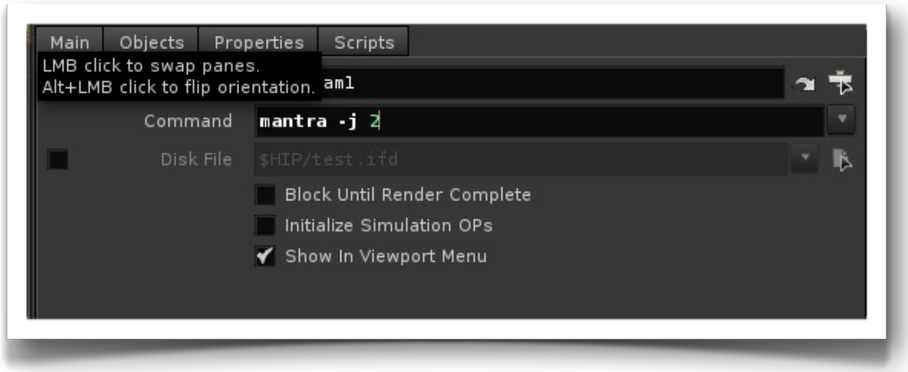
- Limiting the number of CPUs Rendering (A Diversion)
- Saving and Rendering an IFD
- Definitive Resource for Mantra Learning
- Viewport Speed vs Render Speed
- First Step of Rendering - Refinement
- Benefit of Instancing or not
- Bump Maps
- True Displacements
- Buckets
- Coving
- Opacity
- Shading
- Jitter

SIDE EFFECTS SOFTWARE

Limiting the Number of CPUs for Rendering

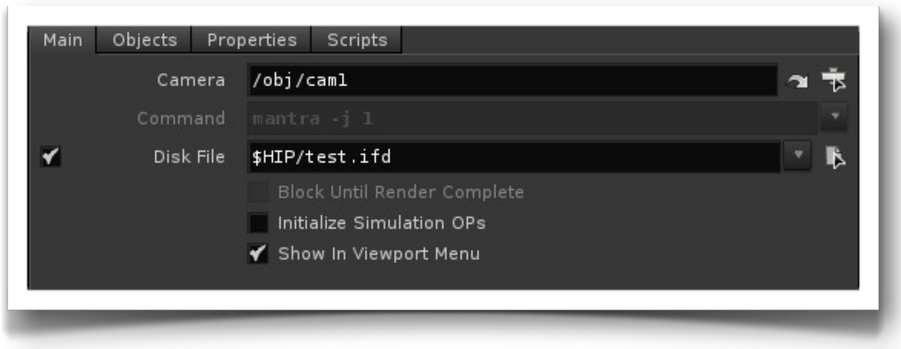


- Drop down a Mantra Node
- In the Main Tab Click the Drop Down Menu for Command
  - Select “Use One CPU”
  - Change the 1 after -j to the number of processors you want



SIDE EFFECTS SOFTWARE

Saving and Rendering a IFD



Without an output file the render is saved to whatever the hip files Mantra node was setup for

- IFD - Instantaneous Frame Description format produced by Houdini and consumed by mantra to produce a rendered image or animation sequence. The IFD file contains a complete description of the scene and how to render it.
- IFD is the Houdini equivelent of the RIB format
- After rendering the IFD to disk. Try typing one of these commands in the shell:
  - mantra < test.ifd
  - cat test.ifd | mantra test.jpg

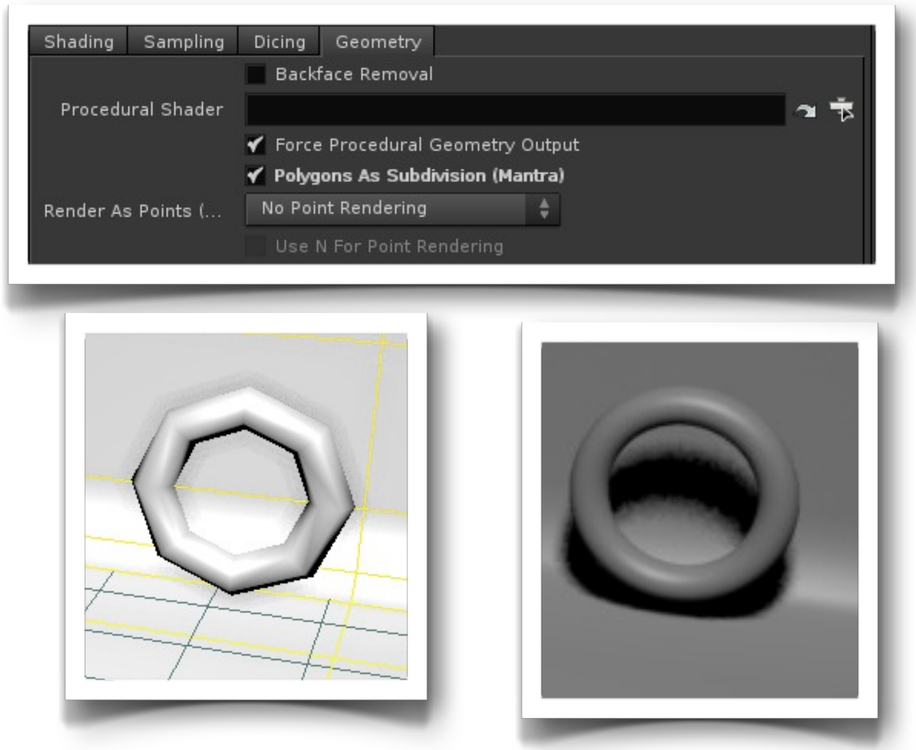
SIDE EFFECTS SOFTWARE

Master in Class in Mantra

- Old but still very good! Masterclass: Rendering With Mantra 9.5
  - http://www.sidefx.com/index.php?option=com_content&task=view&id=1414&Itemid=344
- Masterclass: Mantra & Houdini 11
  - http://www.sidefx.com/index.php?option=com_content&task=view&id=1412&Itemid=344
- Masterclass: Mantra Rendering
  - http://www.sidefx.com/index.php?option=com_content&task=view&id=2160&Itemid=344

SIDE EFFECTS SOFTWARE

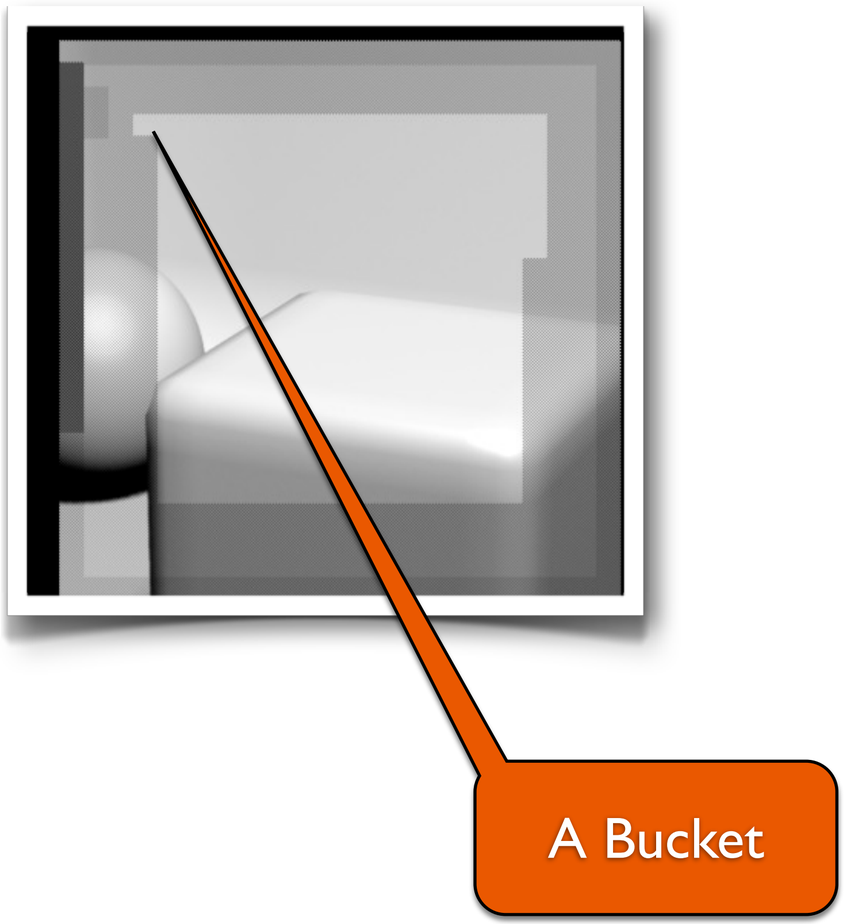
Viewport Speed vs Rendering Speed



- Houdini has to reduce every polygon into smaller polygons until it can render the object smoothly
- Do you want to subdivide upfront or let Houdini subdivide at render time
  - Game Engine Perspective - Let the Renderer do the work
  - Maybe not the right solution - Might be better to optimize yourself
- Try the following make a torus with very view primitives
  - Render
  - At the Object Level go to the geometry tab of the Torus and select “Polygon as Subdivision”

SIDE EFFECTS SOFTWARE

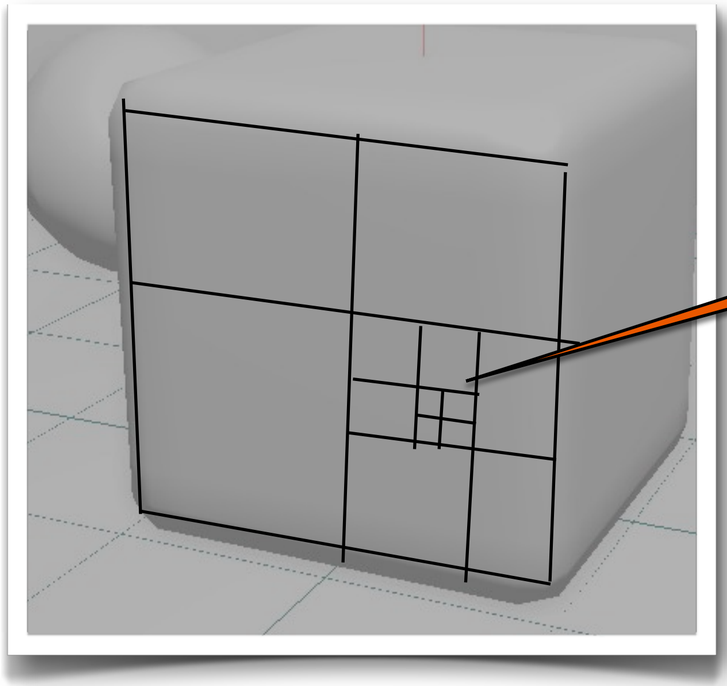
Refinement



- Given the current view of the camera Mantra has to render every primitive in view of the current scene.
- Go to render View
  - Turn off Preview
  - Click render, then stop render
  - See size of bucket
  - Each poly must be reduced into smaller polys to render smoothly
  - Refinement takes your geometry and subdivides your geometry into very small pieces (Very Costly In Time)

SIDE EFFECTS SOFTWARE

Refinement (cont.)

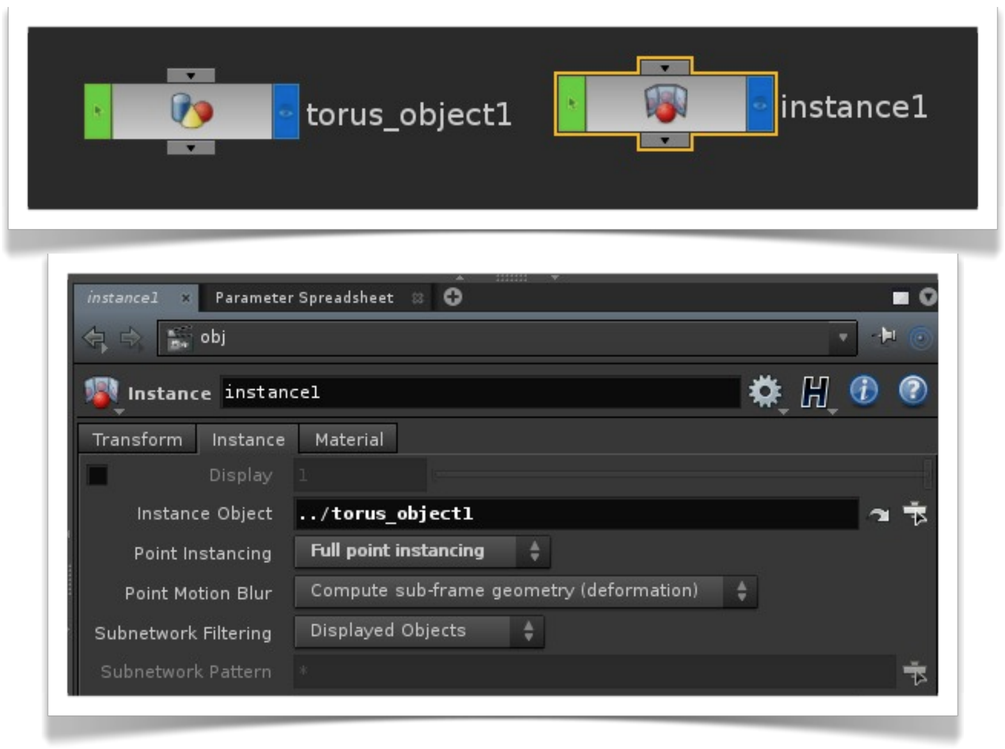


Single Polygon being refined for rendering

- Let's look at a single polygon
- Refinement takes the polygon and cuts it down the middle and say vertically, and then horizontally. Now one poly is four. Mantra then takes one of the four and repeats the process. This is done over and over again in a very long for...loop (actually BSP Tree) for each poly. This is a very time intensive task in Mantra

Benefits of Instancing...

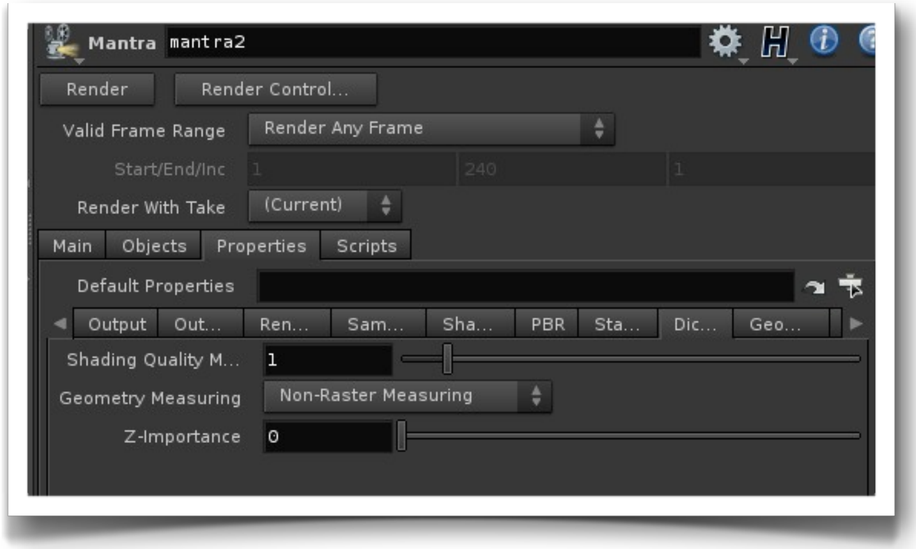
Not really when it comes to refinement



- Instancing helps by only loading in one copy of the geometry to memory. This is great!
- Once the refinement phase starts. Polys must be reduced for each instance.
- As of H12 all rendering engines use the same refinement stage.
- Also, as of H12, the Shading Quality Multiplier is shared among all the renderers.

SIDE EFFECTS SOFTWARE

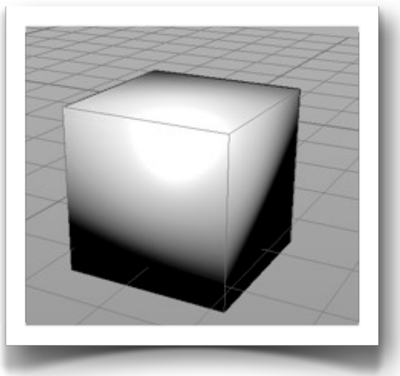
How do I control the Shading Multiplier



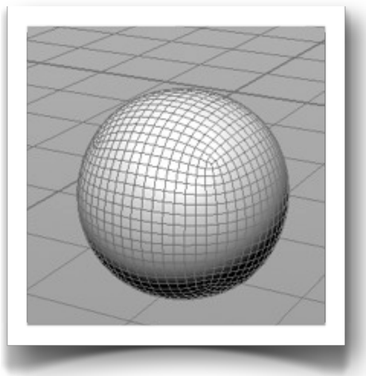
- Increasing the Shadow Quality Multiplier causes more refinement to occur.
- Refinement is a product of Shading Quality Multiplier, bucket size, and view of camera.
- In the Dicing Tab of the Properties tab in Mantra
- Larger is greater quality - Opposite of RSL

SIDE EFFECTS SOFTWARE

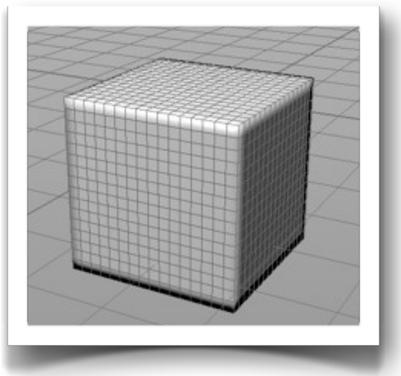
What if I have a Huge Scene?



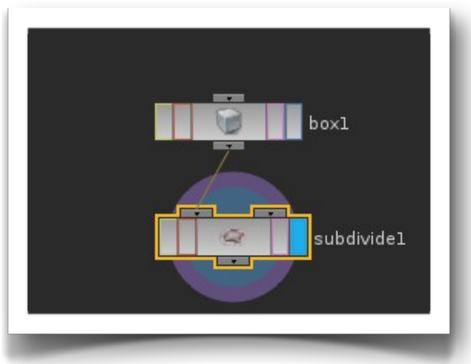
Original Box

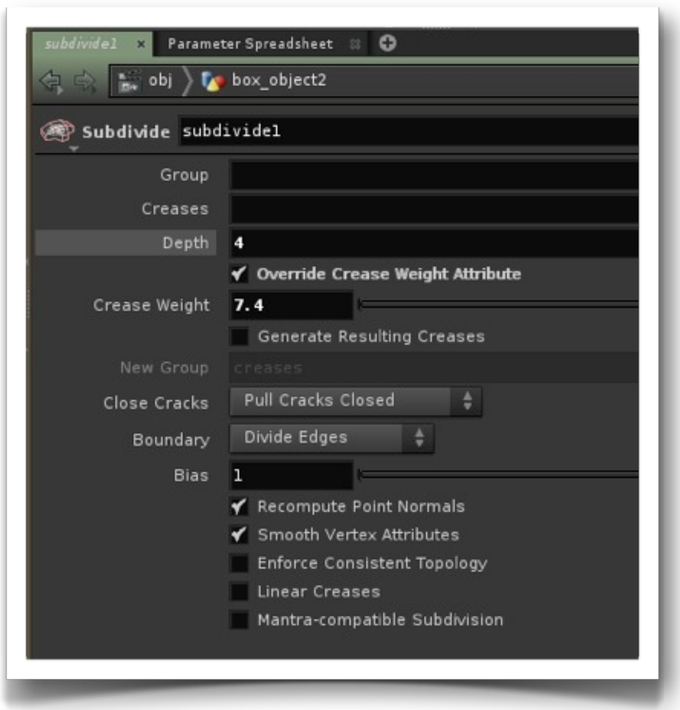


Subdivided, Depth = 4



Subdivided, Depth = 4
Override Crease
Weight Selected
Crease Wight = 4





- I have hundreds of millions of polygons. It might be better for the user to do the refining
- How do I refine on my own?
- Drop down a box
  - Go inside the geometry
  - Add a Subdivide SOP
  - LOOKS AWFUL!
  - Turn on “Override Crease Weight”
  - Increase Crease Weight to 1 or greater
  - Change Depth to say 4
  - Adjust Crease Weight

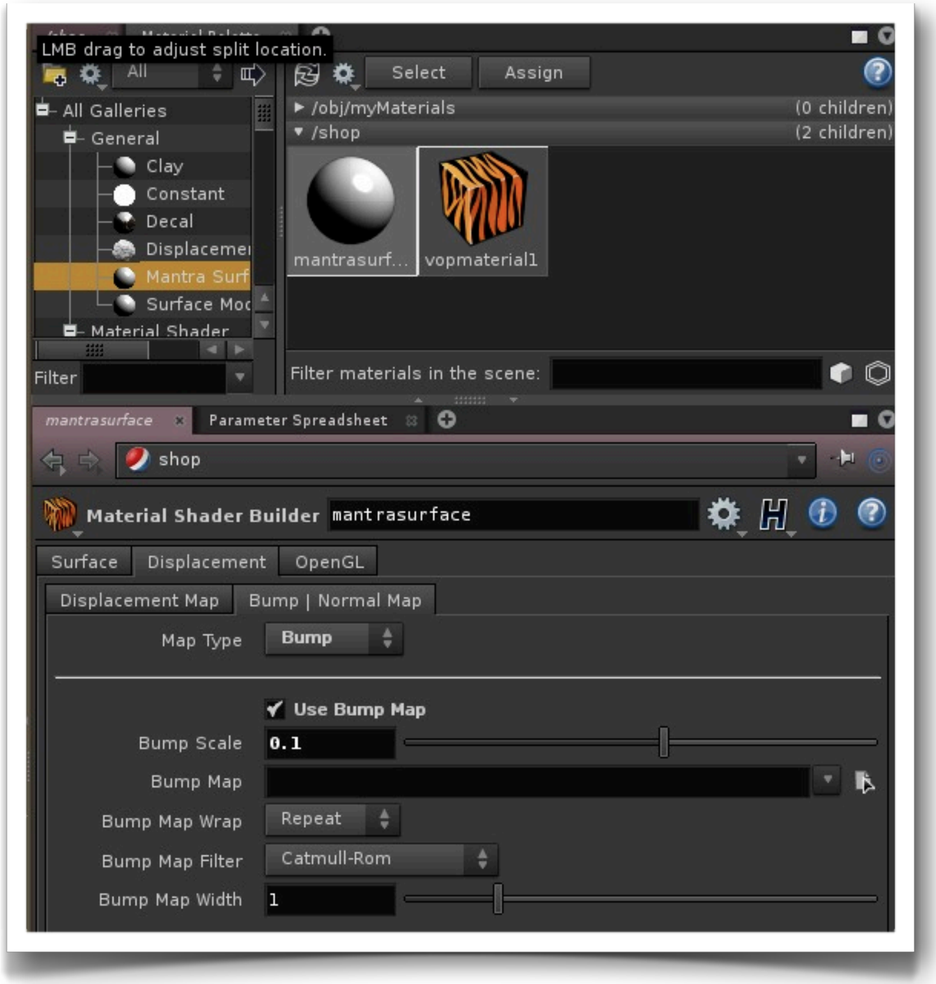
SIDE EFFECTS SOFTWARE

Moral of the Story...

- In large scenes with high poly counts if you do manual refining then instance objects are all the same through the rendering pipeline. Therefore rendering can be much faster.
- Look through the camera.
  - Try to have polys of instances roughly the same size
  - Some instances might be better a separate objects and un-refined.
- What if I am applying Displacement Maps?
  - All Geometry must then be diced?
  - Do you really need Displacements?
  - What about Bump only?

Bump Maps...

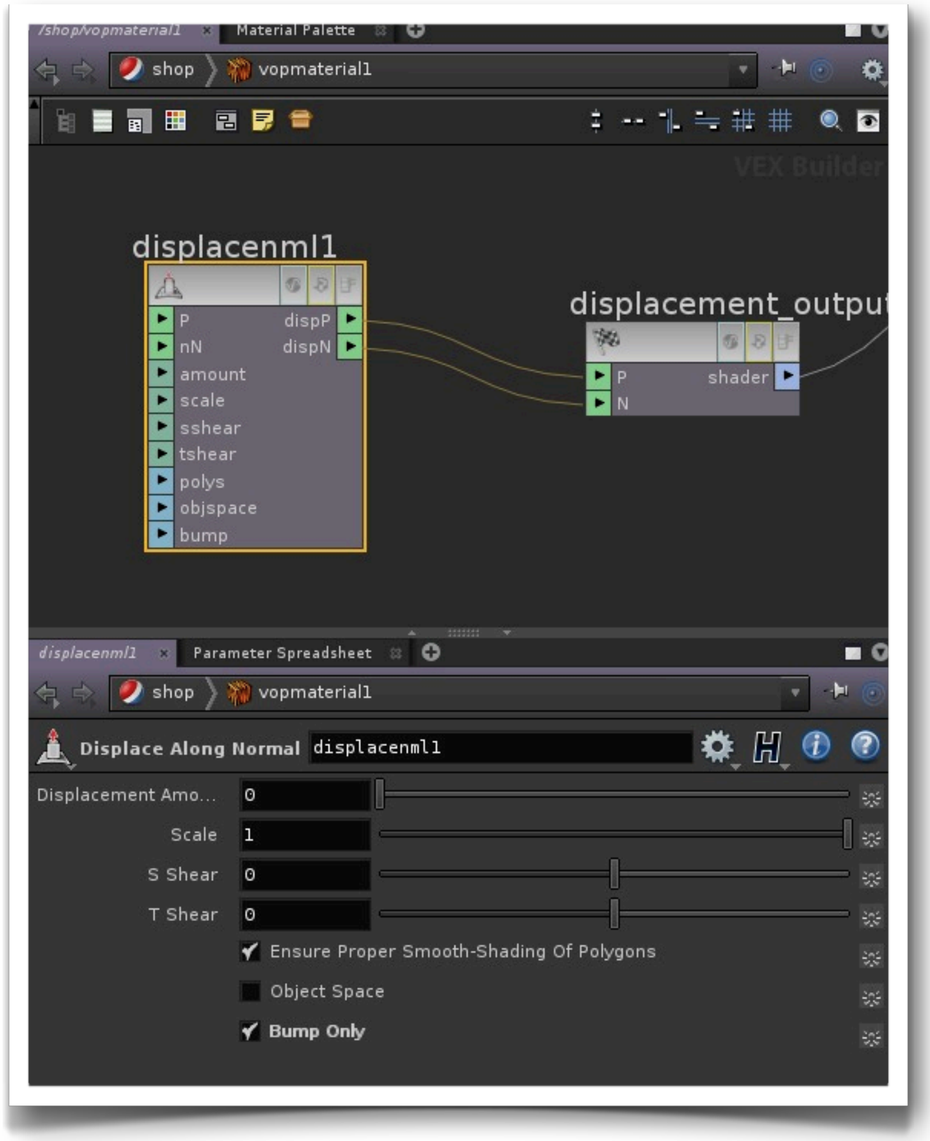
Mantra Surface Shader or Make Your Own



- In the Material Palette
  - Drop down a Mantra Surface Shader
    - Select Displacement Tab
    - Sub Select Bump|Normal Map
    - Select Use Bump Map

SIDE EFFECTS SOFTWARE

Bump Maps... (cont.)



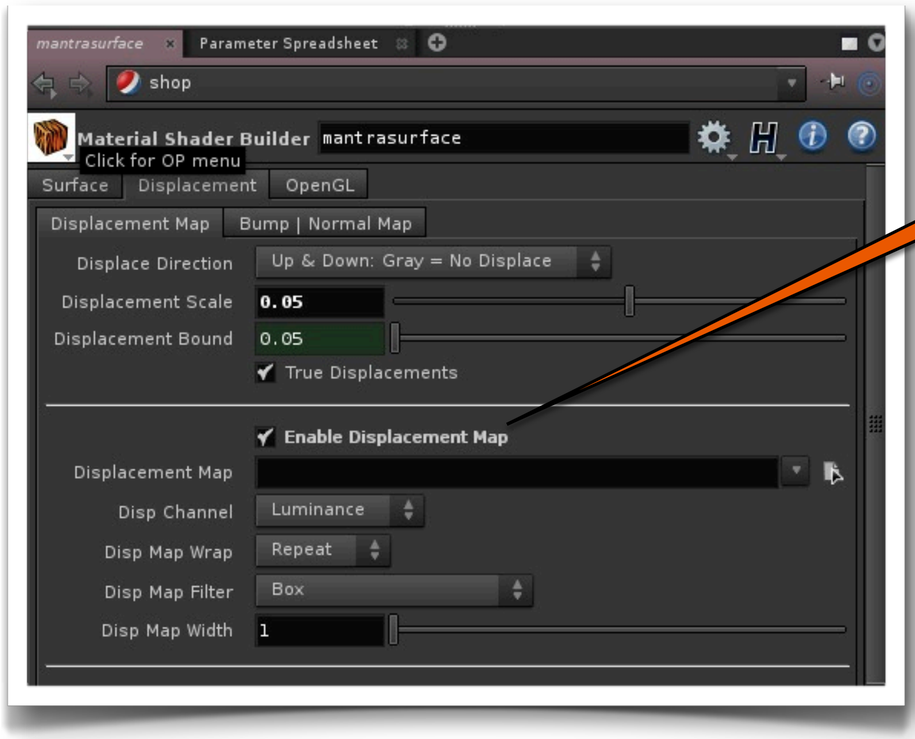
Make Your Own

- In a SHOP Context
  - Drop down a Material Shader Builder
    - Inside the VOP Network
      - Drop Down a “Displace Along Normal”
      - In the parameters choose “Bump Only”

Only use Displacements when Necessary

SIDE EFFECTS SOFTWARE

True Displacements

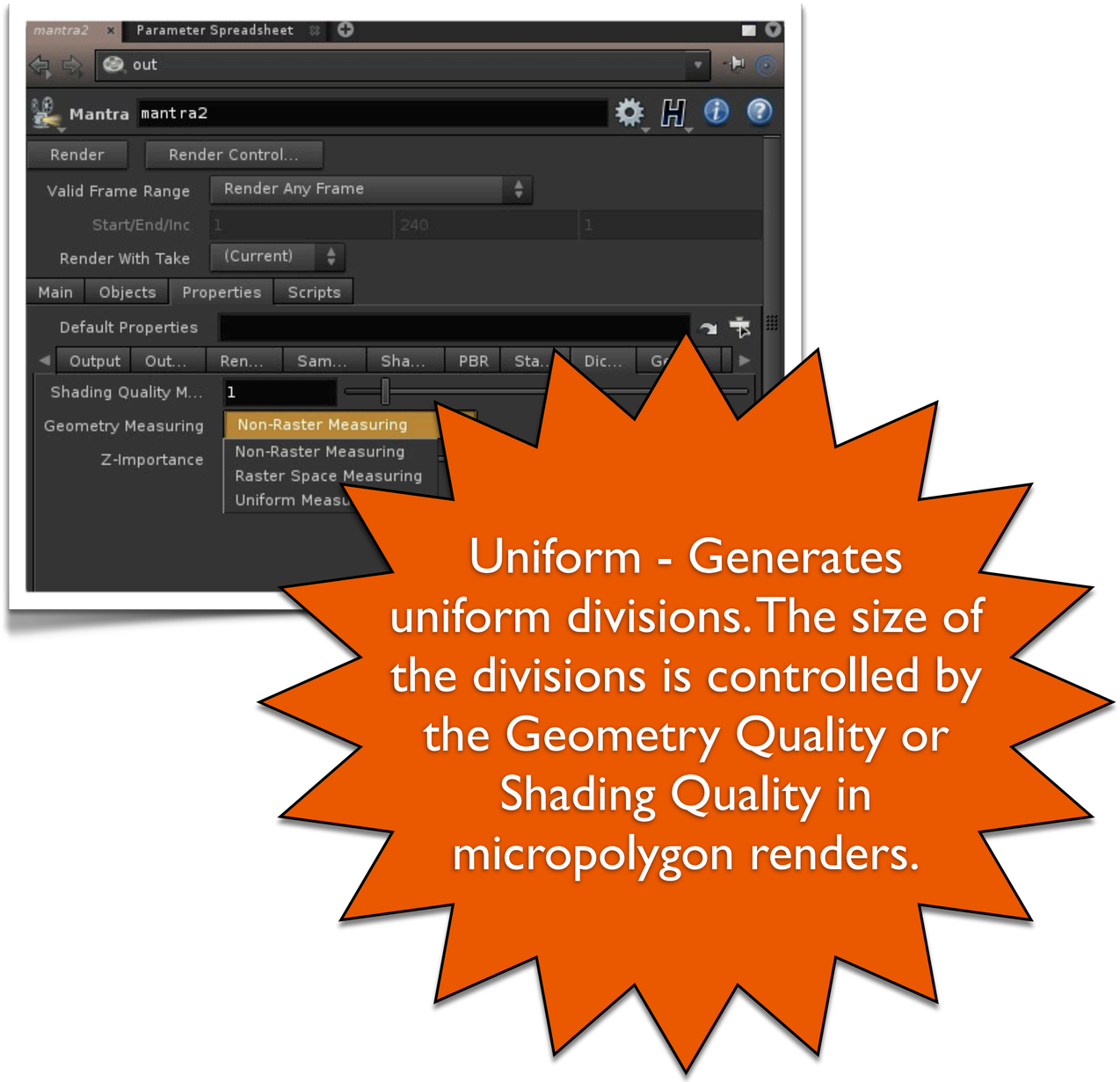


True Displacement in the Mantra Surface Shader

- With True Displacements on Instances become unique even before refinement!
- May cause memory bloat

SIDE EFFECTS SOFTWARE

What Properties Effect Refinement?



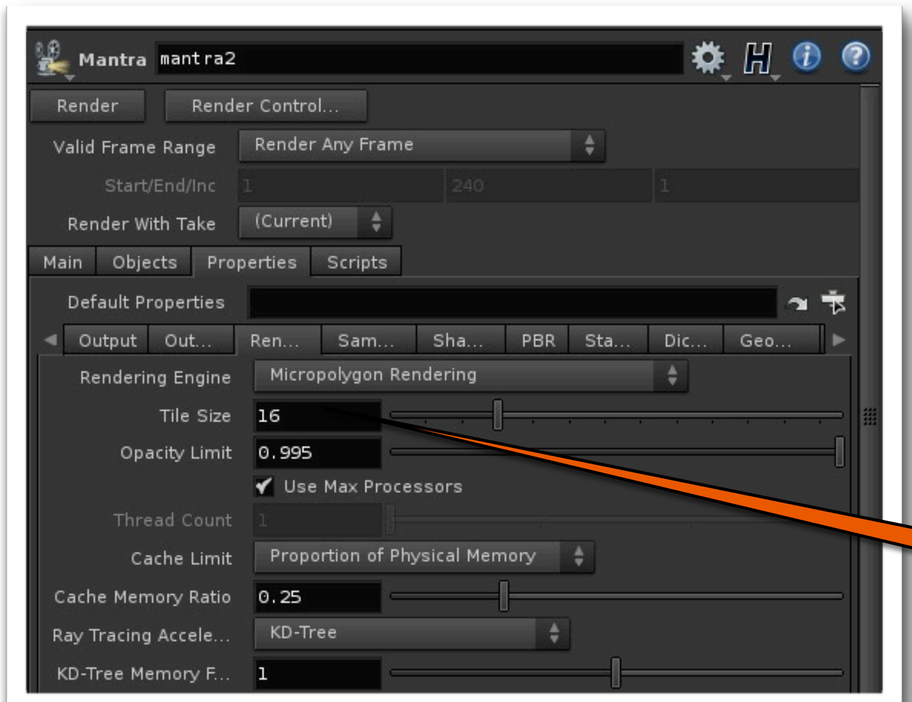
- Shading Quality
- Displacement
- Level of Detail in NURBs or Bezier Attributes
- Dicing
  - Shader Quality
  - Geometry Measuring
- In Geometry Measuring you can use Uniform Measuring to get back Instancing. It measures at Scene View vs Camera View
  - Non-Uniform = Camera Space
  - Raster Space = RasterSpace
  - Uniform = Scene Space

SIDE EFFECTS SOFTWARE

Buckets

16x16 bucket

64x64 bucket

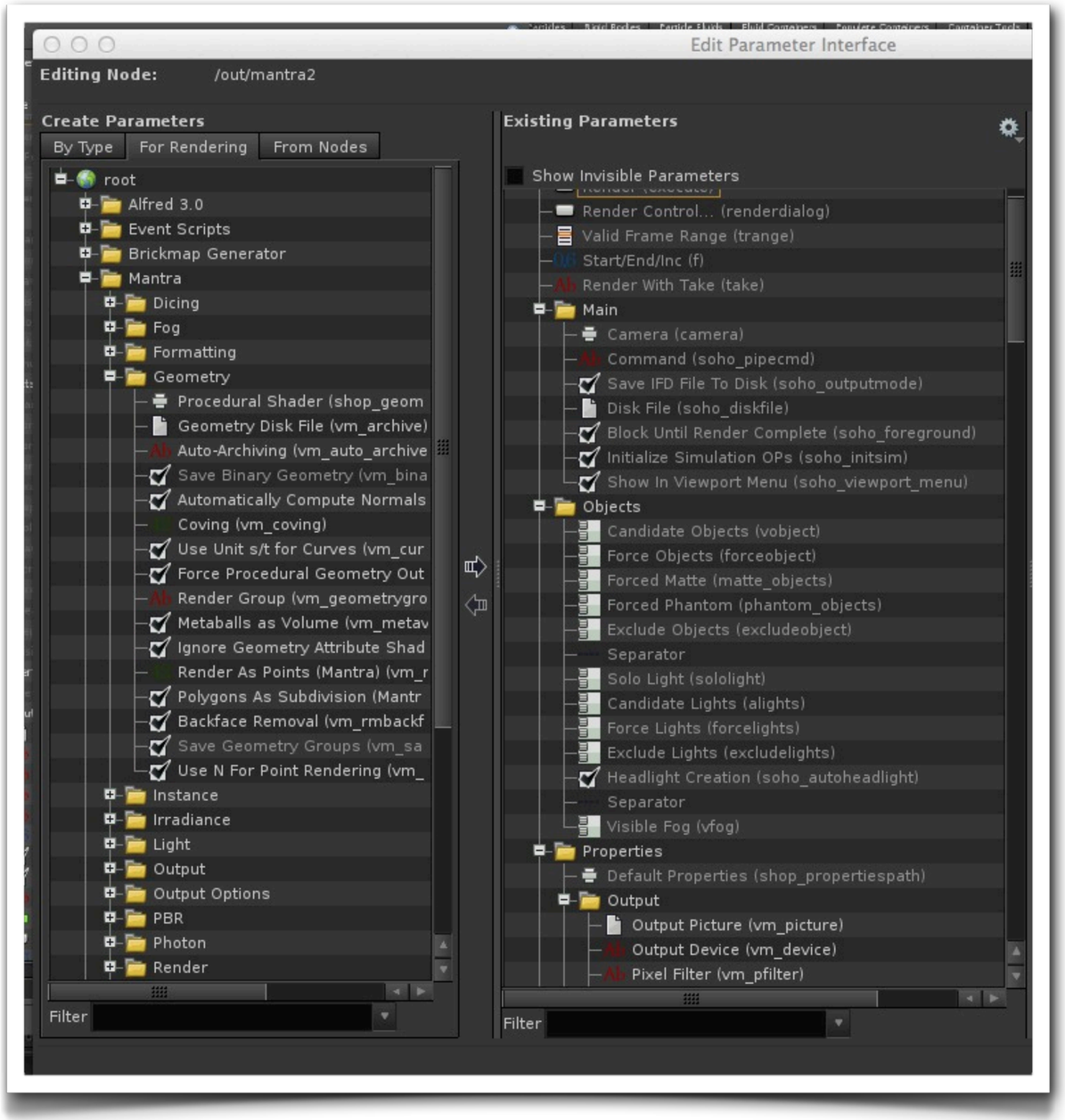


- Mantra by default uses 16x16 pixel bucket size.
- Each bucket is rendered independently of its surroundings
- After a bucket is rendered, the renderer can throw away all the computed micropolygon geometry, saving memory
- If you are having memory issues Try increasing the bucket size.
  - Sounds like inverse logic, but since after a bucket is rendered Mantra releases the memory you might have a more efficient render.

Bucket Size

SIDE EFFECTS SOFTWARE

Quick Diversion - Rendering Properties



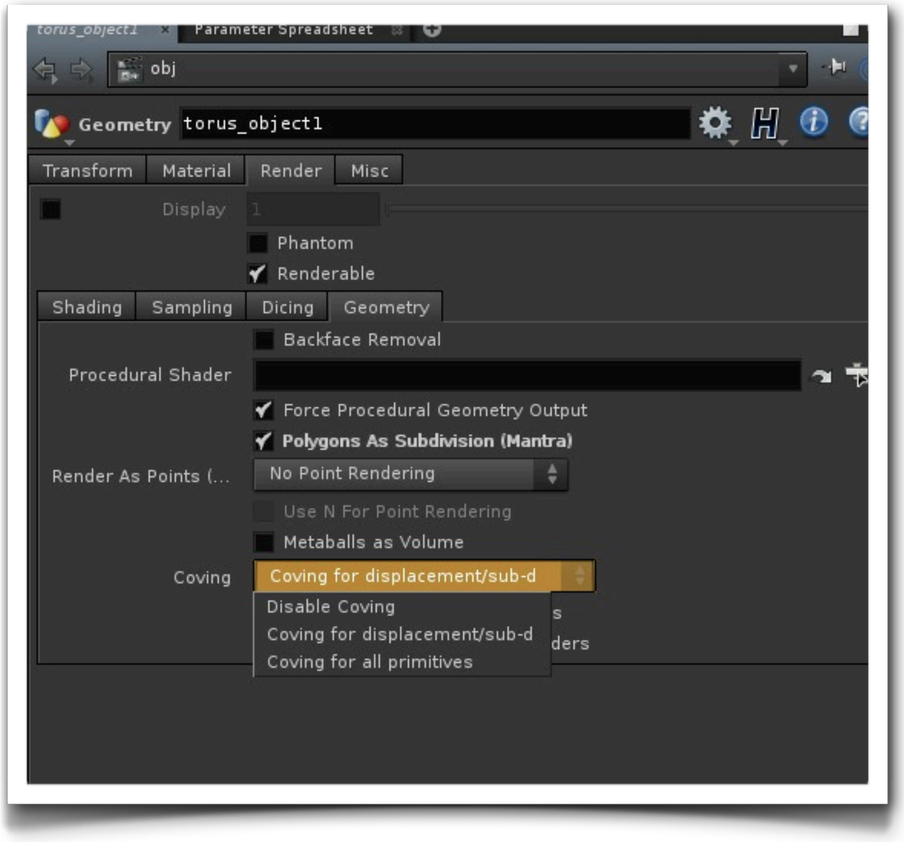
- Let's take a under the hood view of the Mantra Rendering Pipeline
  - Open up Rendering Parameters
  - Mantra ---> Geometry
    - You can see the geo pipeline in the refinement stage
- Now look at Dicing
  - If you look at flatness. It deals with curves and see what value Mantra should determine that the curve is flat
- Re-Dice Displacements under shading (remember from last week)

SIDE EFFECTS SOFTWARE

Note on Re-Dicing

- Make a hook just using displacements.
- You can take a poly, displace along the y. Re-dice and displace in the z, re-dice and displace on the x, finally re-dice and displace on the on z.
- What happened to refinement?
  - Refinement must now be redone after each displace.
  - This will yield huge memory bloat.

What is Coving?



- Coving is the process of filling cracks in diced geometry at render time, where different levels of dicing side-by-side create gaps at T-junctions.
- Do not disable Coving - It can hurt render times and create ugly holes in the rendered geometry
- The default setting, Coving for displacement/sub-d, only does coving for surfaces with a displacement shader and subdivision surfaces, where the displacement of points can potentially create large cracks. This is sufficient for more rendering, however you may want to use Coving for all primitives if you are using a very low shading rate or see cracks in the alpha of the rendered image.

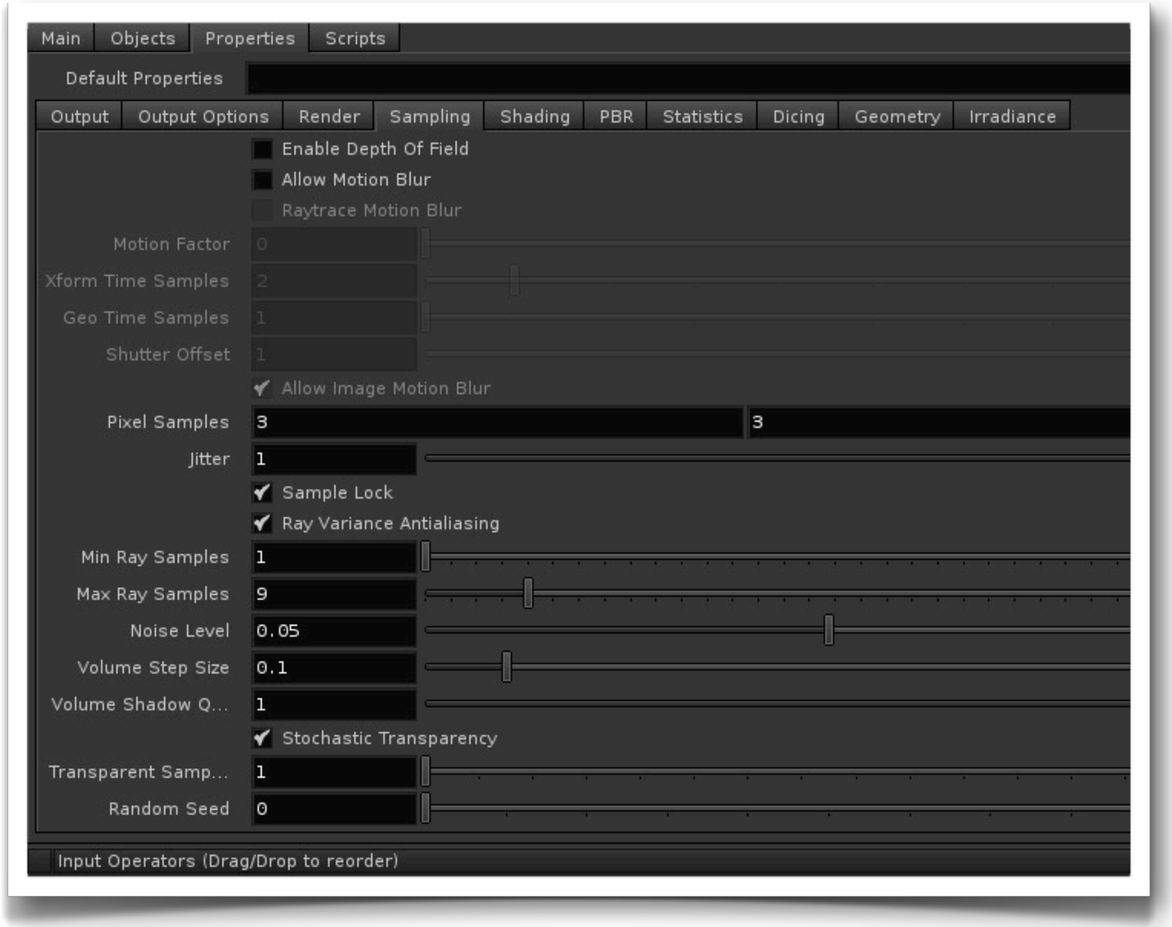
SIDE EFFECTS SOFTWARE

Opacity Stage

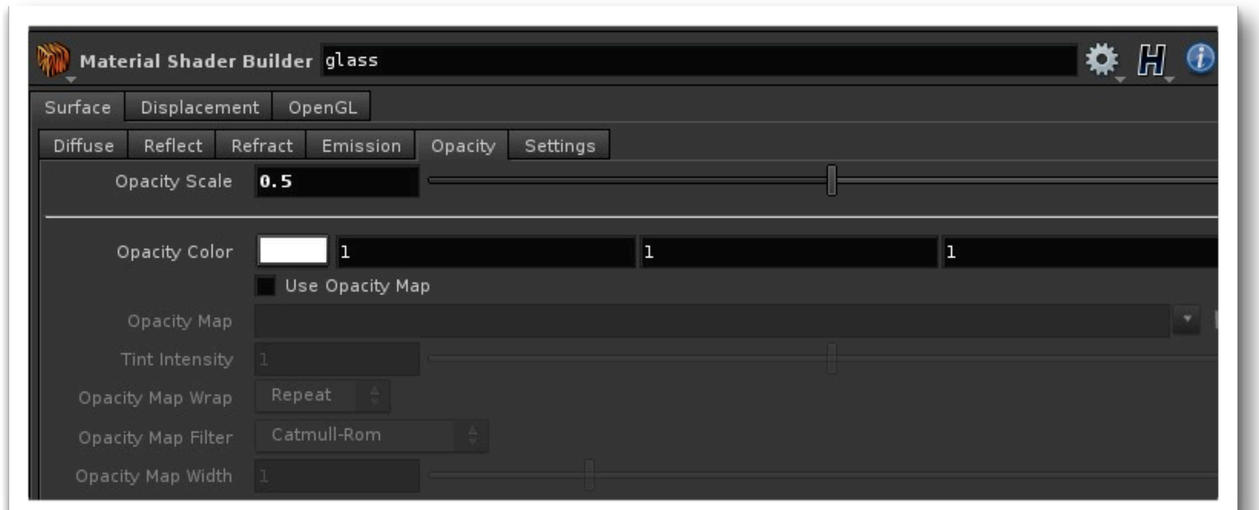
- Add a Sphere at the Object Level
- Partially intersect with Box
- Assign a Glass Shader
- Hit Render using PBR
- After Refinement with have Opacity and Culling Phase.
- Every vertex on every primitive. Only the “Of” part of the geometry is tested.
  - Different then RSL because Mantra optimized for Volumes, Fog.
  - 1 = Opaque, O = Transparent
  - Anything in Between gets tested further in the pipeline

SIDE EFFECTS SOFTWARE

Stochastic Sampling

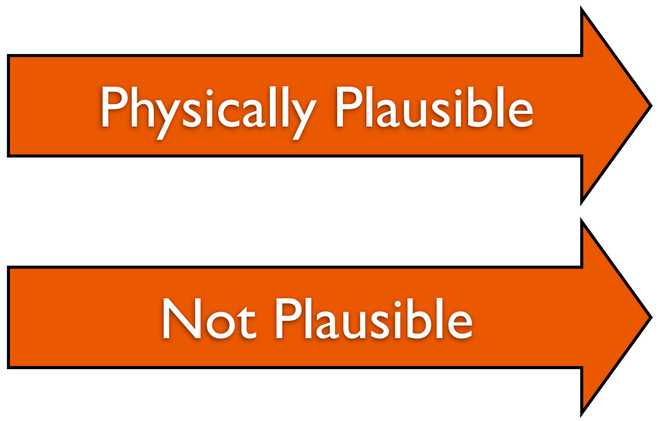


- Glass has an Of of 1 or 0
- We can break the shader by going to our Glass Shader and Set Opacity Scale to 0.5
- If Transparency is detected, Stochastic Transparency is fired off
- Only for Ray Tracing, not Micro Polygons



SIDE EFFECTS SOFTWARE

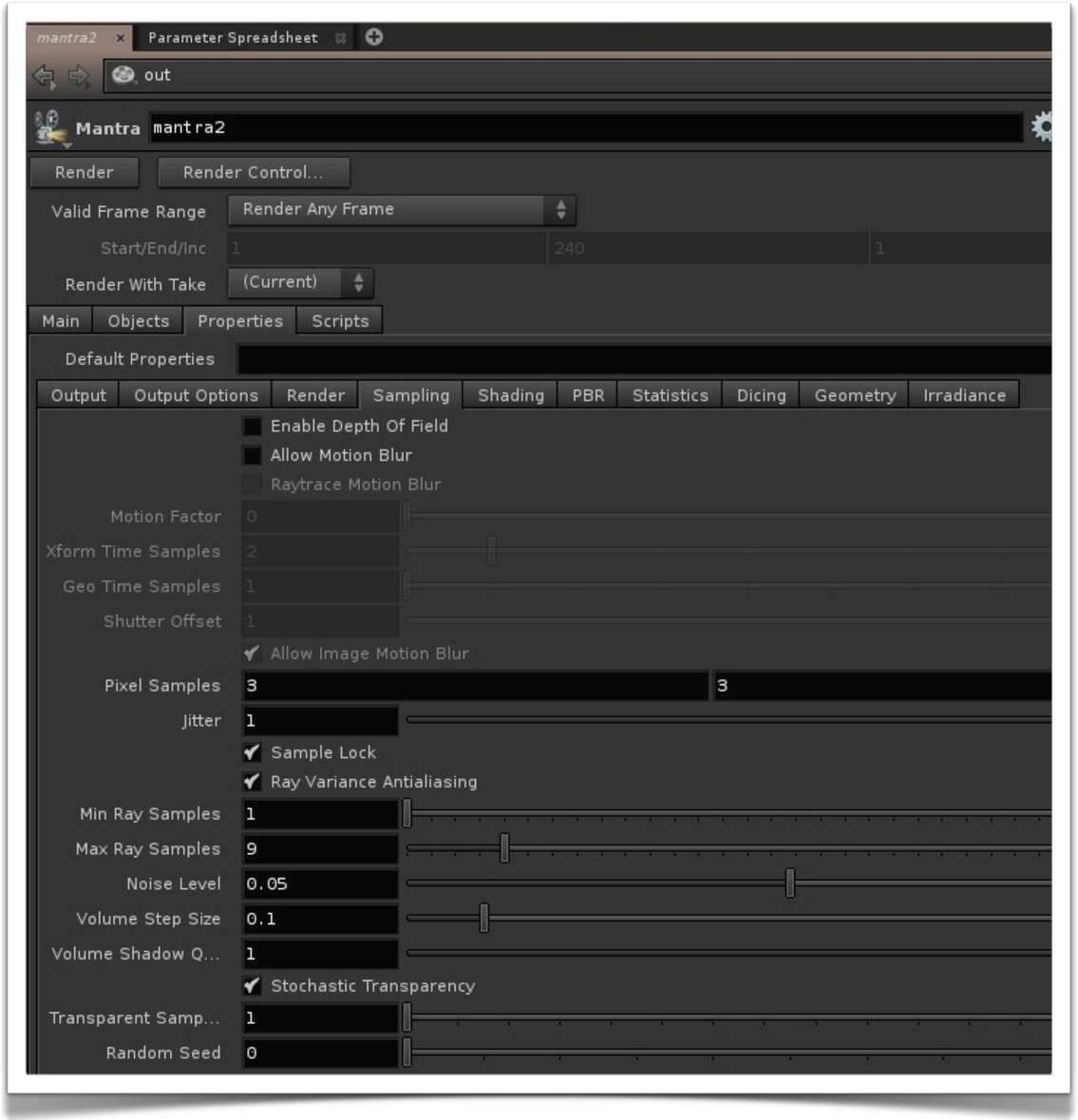
Shading



- Three Different Renderers
  - PBR (Physically Based Renderer)
    - PBR is Non-Biased Rendering
  - Ray Tracing, MicroPolygon Renderer
    - Biased Rendering
- PBR - Area Lights Render faster then Point Lights

SIDE EFFECTS SOFTWARE

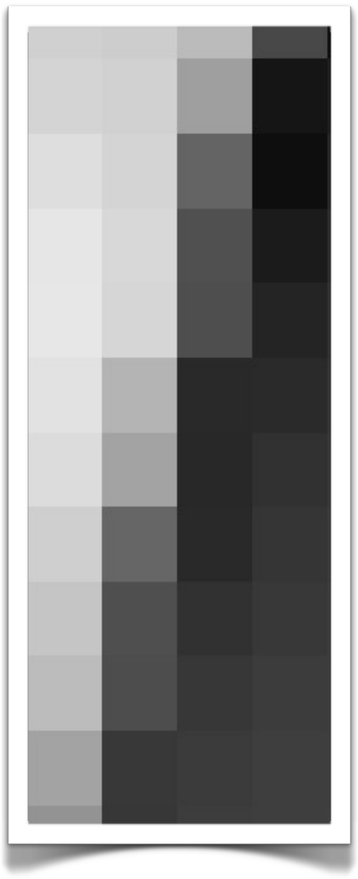
Pixel Samples



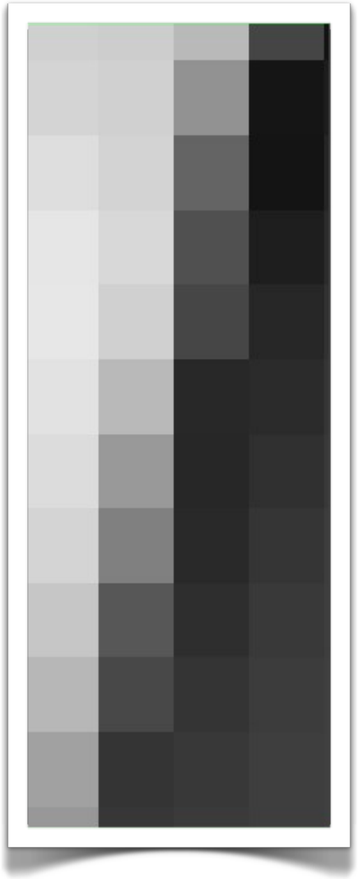
- 3X3 Samples
  - have separate samples in case you are using non square pixels or working with anamorphic film
  - What is Jitter?
    - When shooting rays at a surface to calculate color randomly jitters rays to get more natural looking results.
    - Look at 1980 films before jitter to see the super clean, crisp look.
    - Leave at 1

SIDE EFFECTS SOFTWARE

Zoom into a Pixel



Jitter = 1

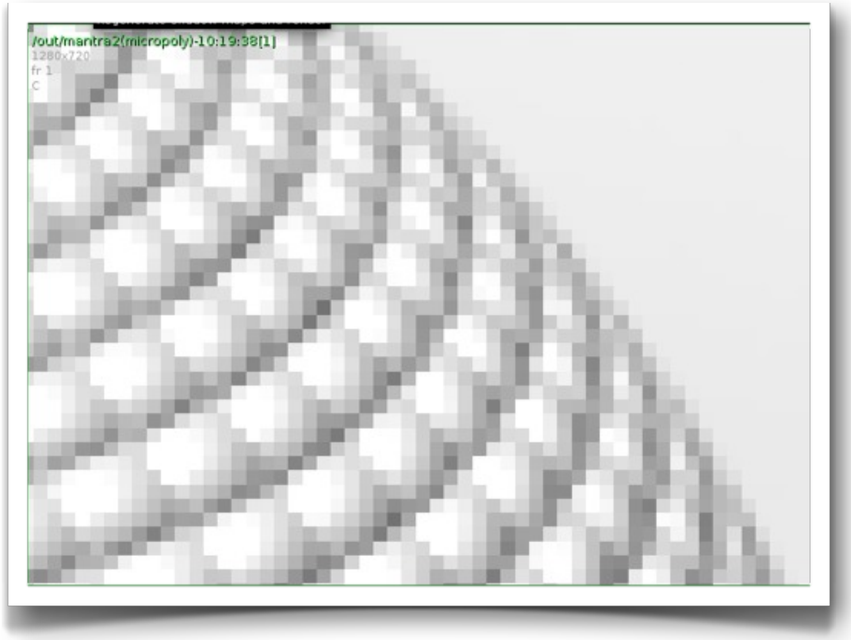


Jitter = 0

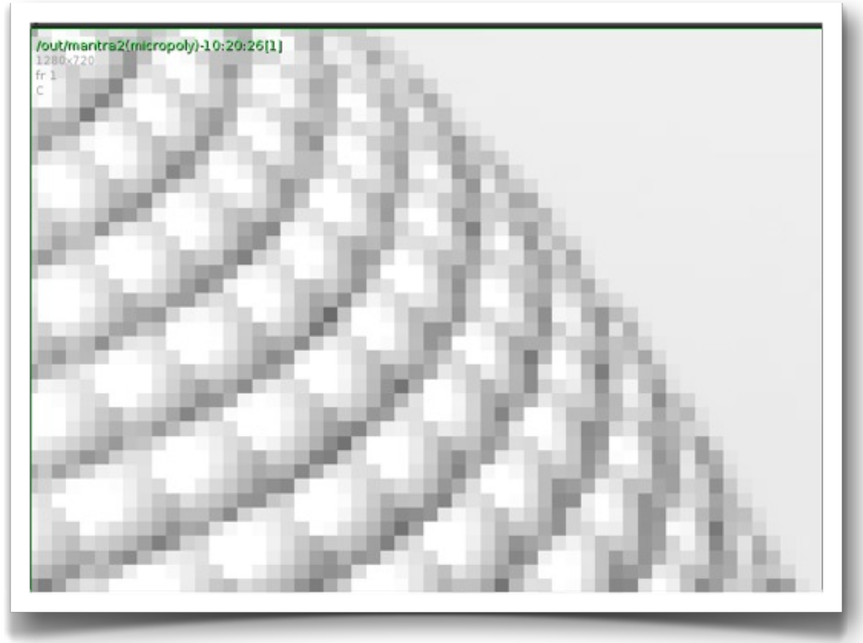
- Notice how with a Jitter of 1 the pixels are not as uniformly shaded as with jitter = 0
- Pixel Samples used by all three rendering engines
- For Micropolygon rendering - one more step of refinement. Will refine to one pixel
- In Micropolygon every Vertex has to be rendered/shaded.
  - Not useful for Volumes

When do I need to Increase Pixel Samples

- If the Geometry has a lot of detail or texture sample has a lot of high frequency noise increase pixel samples
- If not reduce Noise Level for better results

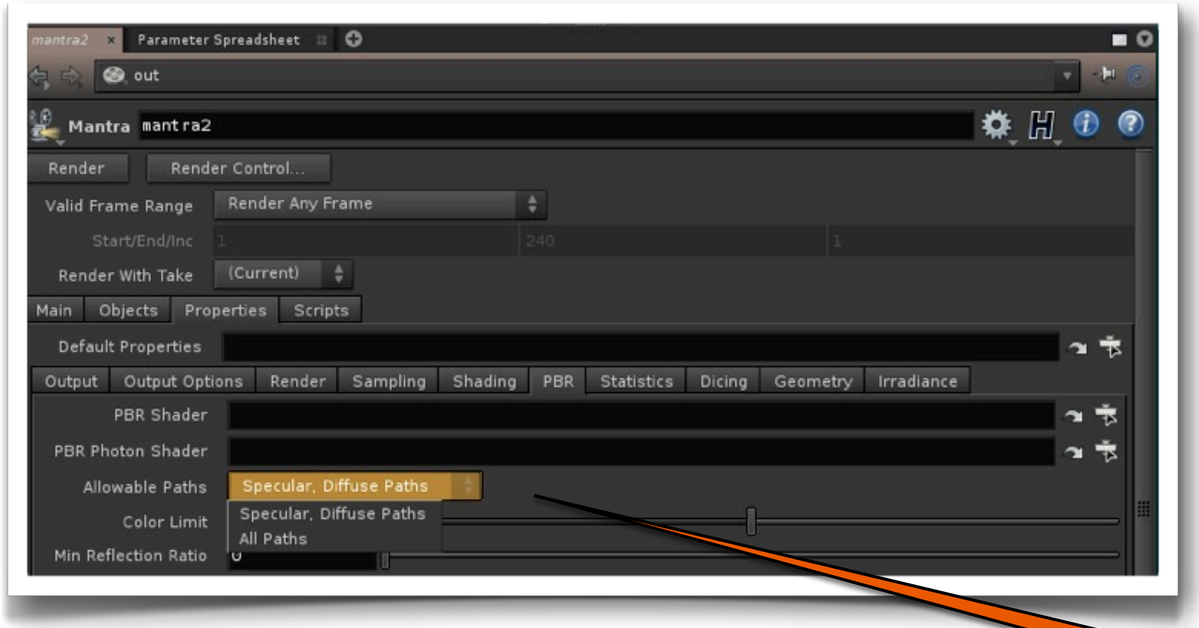


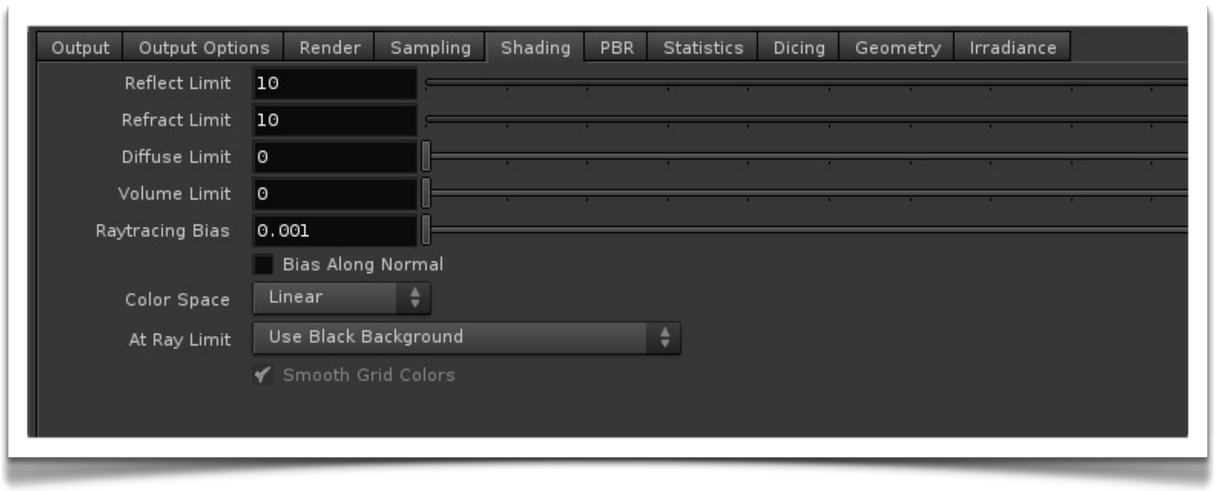
6x6 Pixel Sample



2x2 Pixel Sample

Allowable Paths





- In PBR Pixel Sampling is Tied to Allowable Paths
- Also...
  - under Shading
  - Look at Reflect Limit, Refract Limit, Diffuse Limit

NxN Pixel Sample is fired off for each Pass Selected

SIDE EFFECTS SOFTWARE