



This tutorial shows you how create a point cloud and brickmap to speed up occlusion when your working with large sets that require moving camera shots.

Requirements:

Autodesk Maya
RenderMan for Maya plug-in.

A BrickMap is a good way to speed up your render times if you have a complicated environment that also has a moving camera shot. By creating a BrickMap you no longer have to calculate the occlusion on each frame; it simply references the BrickMap file.

You will want to wait to create a BrickMap pass until you have finished modeling your set, as you can't make any adjustments or changes to your set once you create the BrickMap, or else you will have to re-create it.

OK, let's start:

1. Open your environment file.
2. Make sure the file has been cleaned up. Check for:
 - Double sided
 - Subdiv scheme
 - Normals facing out
 - Delete history
3. Go to Window>Rendering Editors>Renderman>Shared Geometric Attributes
 - Click [Create]
 - Click [Edit]
 - Add the RenderMan setting [Dice Rasterorient] and un-check the check box to off.
(This ensures even tessellation of the scene regardless of Camera position)
 - Select all of the polygon geometry in your scene and click [Attach]
4. Create a new RenderManENV light from the features tab in the RenderMan render globals.
5. Set the settings to roughly:
 - Shadowing = colorbleeding (or occlusion).
 - Sampling Mode = sampled
 - Samples = 512 or higher for production quality (1024 is production quality)
 - Max variation = 0 or 1. Zero will be the best looking but will take a long time to render.
6. Still in the features tab click the checkered icon next to the Bake field to create a rmanMakeGlobalDiffuse3dPass.



7. In the Passes Tab select the 'rmanMakeGlobalDiffuse3dPass' and set the

- Caching Behavior = Compute
- Phase = Once Per Job

8. From the passes tab select the 'rmanRenderGlobalDiffuse3dPass' set:

- Cull Backfacing = Check box off
- Cull hidden = Check box off

9. From the Create Menu in Maya create a camera and name it whatever you like. In this tutorial we will call it [BAKECAM].

Notes:

Bake maps are dependent on your render size. If you setup your render globals to be 720x405 chances are your render will be to low quality to use. Setting it to 1440x810 will let you be able to zoom in a little bit. The higher the resolution the longer it will take to render but the closer you will be able to get before it starts to degrade.

If you have a super computer this is not a problem but if you run into render problems due to the image size then you need to render multiple times and stitch the pointcloud files together into a BrickMap. I'll add how to do this at the end of the tutorial.

10. Move the BAKECAM camera into a location that it encompasses the entire scene.

12. In the Advanced Tab of the Render Globals you can change the output directories field for [Render Data] to a folder path if you want to change the location where your BrickMaps should go else they will end up in the default location.

13. Make sure that your BAKECAM is set to be renderable and is the only renderable camera in the scene and save your file.

14. Now you can either:

- In the render view select the BAKECAM camera and render it.
- or
- render close Maya and render from the cmd. (see below)

15. Render from the cmd.

- Click start>Run
- Type cmd and press [enter]
- Type cd\ and press [enter]



16. Check to make sure you have a folder in this location. If not, then create a folder. Don't forget to change the project name to your current project.

z:\MAYA\projects\projectname\renderman this is where your files will be rendered to.

17. Enter the code below into the cmd. (You will need to adjust the drive letter depending on your setup. (Also change the projectname to your current project)

```
"z:\Program Files\Alias\Maya7.0\bin\Render" -s 1 -e 1 -r rman -pad 4 -of mayaiff -proj z:\MAYA\projects\projectname\renderman -im test -rd z:\MAYA\projects\projectname\images z:\MAYA\projects\projectname\filename.ma
```

18. Press enter and let the job run.

19. When the job finishes the BrickMap file will be located at:

z:\MAYA\projects\projectname\renderman\filename\data\

20. To use the BrickMap open your Maya scene file back up in Maya.

- In the RenderMan RenderGlobals select your RenderMan ENV light and press on the checked button to enter your ENV light settings.
- Click on the checked box next to the Bake tab of the RenderMan ENV light.

21. In the Passes Tab select the 'rmanMakeGlobalDiffuse3dPass' and set the

- Caching Behavior = Reuse
- Phase = Once Per Job

22. Do a test render.

If the image that you get back is solid gray then you typed in the wrong command on step 17. Make sure that you have the correct location of your BrickMap file and that the "..bkm" has been removed from the name in the command line.

If you have correctly done the steps above you should have a rendered image of your set with occlusion that renders quickly even if your BrickMap file was a couple hundred megabytes.

You can now set up a camera animation and render out a sequence of images just as you normally would.



PART 2:

How to combine multiple BrickMaps using windows cmd.

Sometimes if your set is large and you need very high resolution BrickMaps you will need to render more than one and combine it. What you will need to do is follow the steps above until step #13 and instead of copying the BrickMap file copy the PointCloud file to the images directory.

1. Let's assume that you will need four Point Clouds renders before you stitch them together.
2. Divide the area of you set up into four locations visually.
3. Place your camera as the image shows below in section #1. *(Make sure to overlap a little.)*



4. Save the file and render it from the cmd following steps #14 through #20, except name the Point Cloud "pointcloud1..ptc" after moving it into your images folder.
5. Re-open the Maya file and move the camera into section #2, save and repeat the above steps naming the new Point Cloud "pointcloud2..ptc".
6. Repeat the steps until all BrickMap files are rendered.
7. Once all files are rendered and in your images folder it is time to stitch them together.
8. From windows.
 - Click Start>Run
 - Type cmd and press [enter]
9. In the cmd prompt:



- type `cd\` [press enter]
- type `cd c:\Program Files\Pixar\RenderManForMaya7.0-1.0\bin` [press enter]
(Directory location might change depending on your install)
- type (see below)

```
brickmake.exe z:\MAYA\project name\images\pointcloud1..ptc z:\MAYA\project name\images\pointcloud2..ptc  
z:\MAYA\project name\images\pointcloud3..ptc z:\MAYA\project name\images\pointcloud4..ptc z:\MAYA\project  
name\images\brickmap.bkm
```

- [press enter to run the above script]

10. Depending on the size of the Point Cloud files, this could take quite a while to work. In the end though you will end up with a file named `brickmap..bkm` in your images folder.

11. Rename the file to:

- `filename_rmanMakeGlobalDiffuse3dPass..bkm`

12. Move the file to:

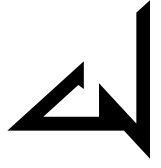
- `z:\MAYA\projectname\renderman\filename\data`

12. To attach it to your scene follow steps #20 to #21.

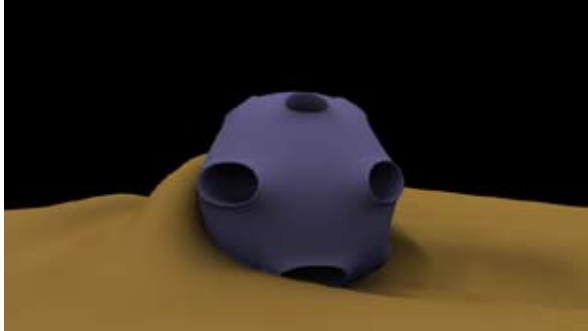
If you decide to rename your ..bkm file then you will have to re-link it in your pass settings as it uses the pass name as well as the image output name in the render global's.

You can always render, and if it does not work as expected, look at the command line history to see where it errored and thinks the BrickMap is located. I would advise moving the BrickMap to the location where Maya is looking for it instead of trying to make Maya look where you currently have the BrickMap located.

*Remember K.I.S.S.
Keep It Simple Stupid*



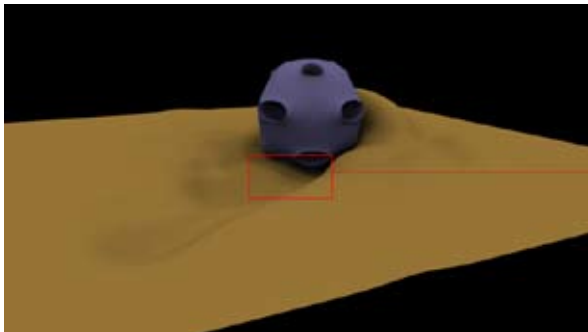
Here are some rendered examples:



This is a correctly rendered BrickMap.



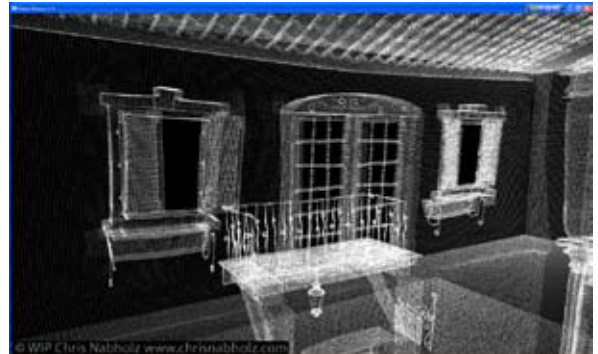
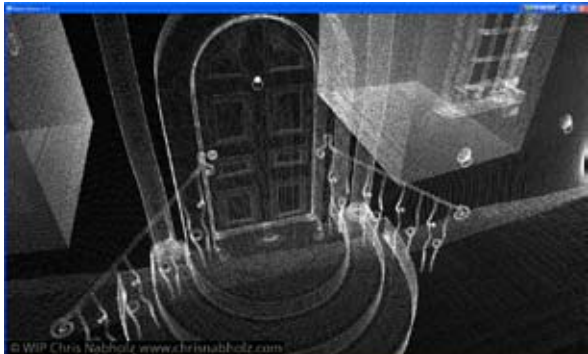
This is when you're incorrectly linked.



Another BrickMap render from a distance.



And a close up to show the degradation



These are both point clouds that are rendered along the way before being converted into BrickMaps. If you have RenderMan Pro Server you can view the point cloud files using a program located in the bin folder named ptviewer.exe.