

## **Exporting UV maps out of 3DS Max and Maya**

This documents outlines how to generate UV maps out of 3DS Max and Maya using Mental Ray (MR) and VRAY with Reference Renders and Reference Scenes and Reference Compositing Projects. The application in mind is to then use RE:Map UV or RE:Map Inverse UV in your favorite compositing application.

This is work in progress, the intent is to update this document over time. Please report any error or change in behavior you notice.

We would like to thank Chad Capeland for helping us document this process and testing the best approaches.

Abbreviation used:

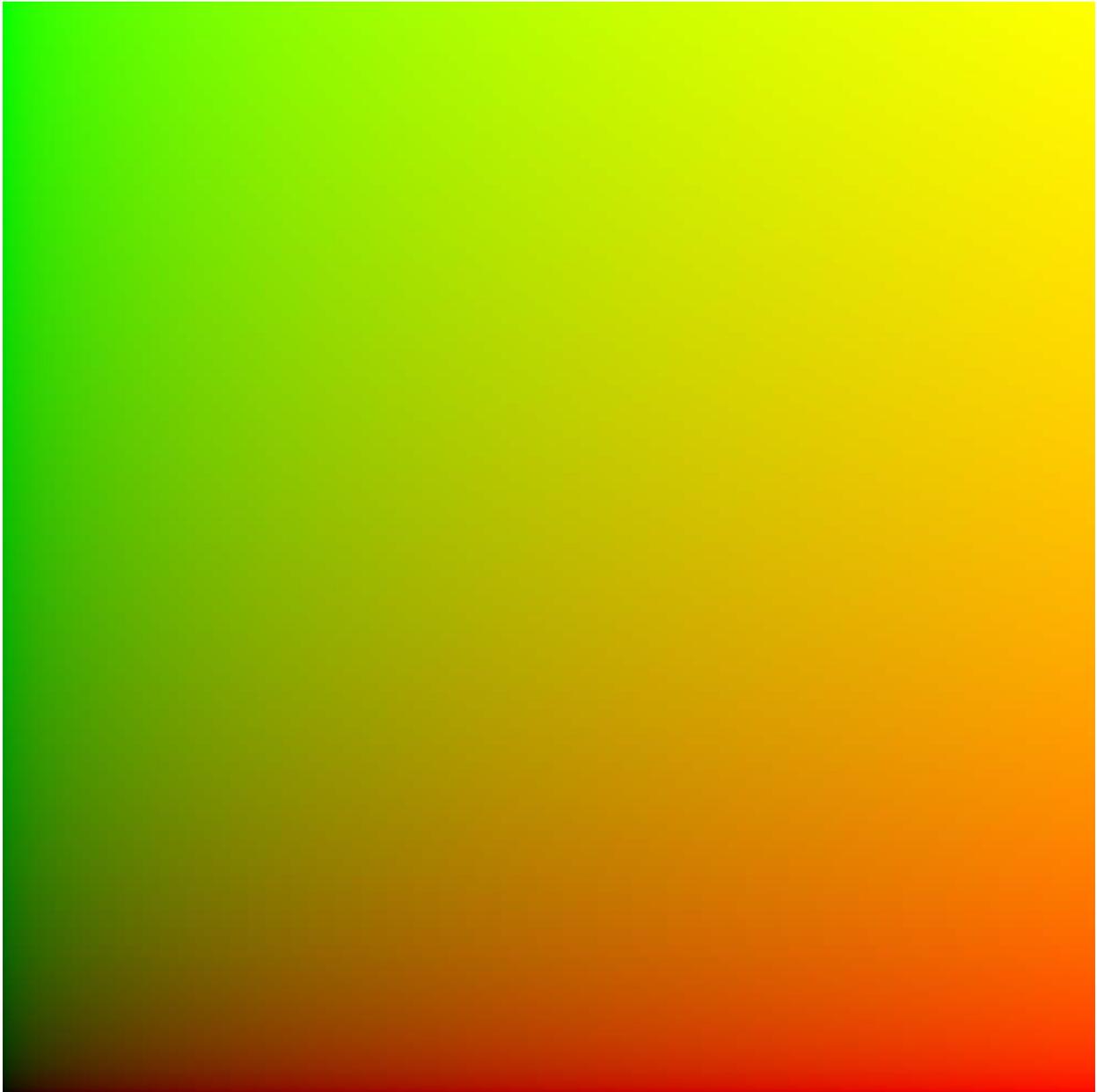
MV: Motion Vectors

UV: Texture Coordinates

MR: Mental Ray (an nVidia product that is included in Autodesk 3D animation systems).

VRAY: A render product from ChaosGroup

## UV Mapping Introduction



This is an example of what a reference UV map would look like: The lower left is 0 in red and 0 in green, the top is 1.0 in red and 1.0 in green. The Red channel represents normalized indexing into horizontal texture space and the Green channel represents the vertical indexing.

Note RE:Map UV and RE:Map Inverse UV often needs the alpha that corresponds to the area to UV map definition. The alpha channel in RE:Map case allows the opacity to be hard coded with the UV maps, allowing to set reflections for example. Without such alpha all undefined values would have 0,0 as coordinate, so would all index the lower left pixel of the image.

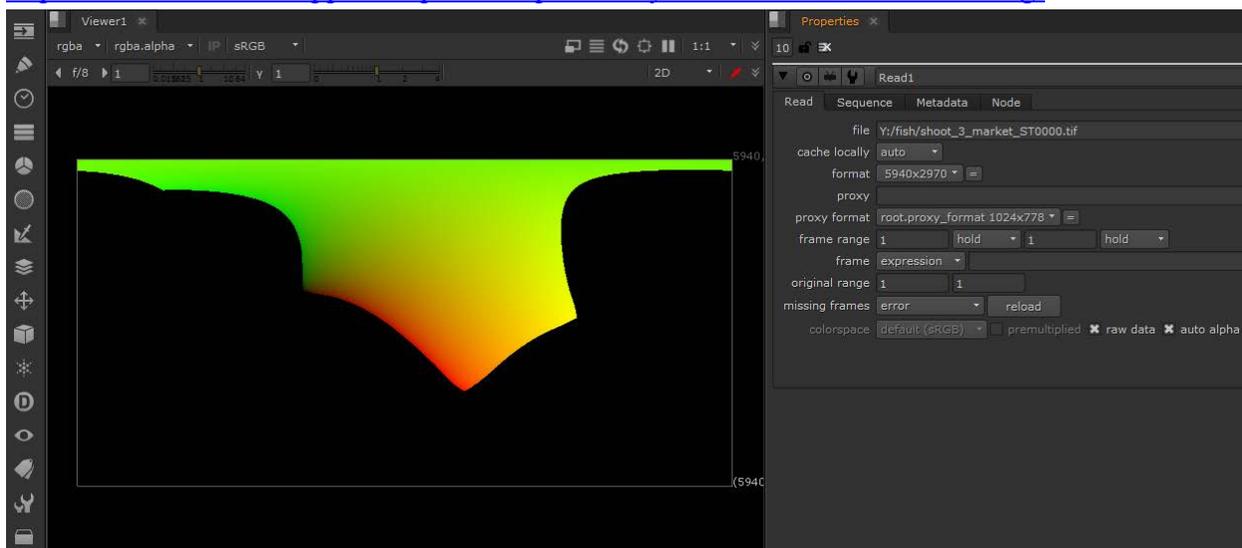
Since earlier used in Renderman of UV and ST coordinates terminology, many applications have sort of made them synonyms or applied their own meaning to it. Originally, in Renderman 25

years ago ST referred to the user-assigned coordinates while UV was the actual surface parameterization. For example one UV unwraps a surface, and warps the texture to fit – the result map would be ST. Since most systems in time have adopted the terminology UV assignment this is what we use here. For our purposes they are functionally synonyms here. The other distinction is we support two UV sampling methods, the classical UV mapping projection with RE:Map UV, and the reverse projection (as used in 3D paint and similar tools) with RE:Map Inverse UV.

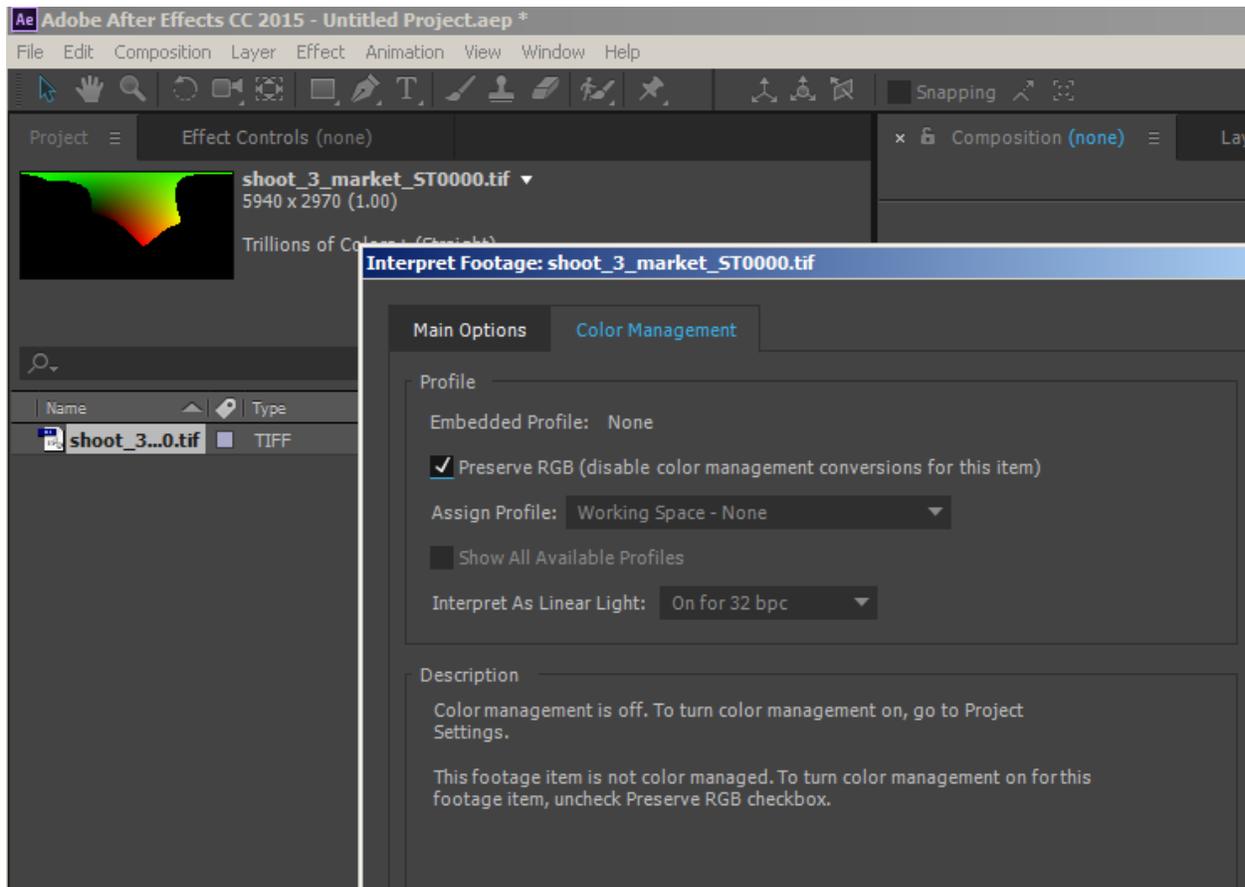
Another important thing to consider is that this mapping (the actual values encoded in the image) are totally sensitive to color space conversion. You don't want any gamma or other color transform applied to the UV maps proper as these values have a particular meaning.

For example in Nuke, notice that when you import a 16 bit UV map, Nuke defaults to sRGB in the Read Node interpreter which will internally trigger a conversion, therefore changes the meaning of the values, so produce improper result. 16 bit UV (true 16 bit not EXR float which is like effective 13 bit in 0-1 space) still has a lot of production value (smaller file size and enough for continuous smooth surfaces at least until 4K, your tolerance might vary). 8 bit UV map is a real no no. There is just not enough spatial coordinates precision with 8 bit.

<http://revisionfx.com/support/faqs/hostfaqs/foundry-nuke-known-issues-tracking/>



Similarly in After Effects, you might need for that sequence to turn off using the Color Management tab of Interpret Footage. Also note that even if we let you work with 16 bits per channel UV map, it is still recommended in AE at least to work in a floating point project as AE in 16 bits is actually 0-32768 (you have a bit less of precision than is actually encoded in the file, which matters as the size of the UV map grows). Of course it's a no no to use AE in an 8 bit project for this work as it will collapse your high bitdepth UV maps to 8 bit. This will generate blocky results.



In AE, if you use Extract EXR, there is no color management process. Also the latest version

The purpose of these documentation notes is to calibrate your process down to final compositing. We highly recommend you do a simple workflow / pipeline calibration test before you come to final conclusions. Doing this ourselves we even discovered that Mental Ray Classic and the new version don't even sample UV the same way. So starting with RE:Map V3 we added a Match Renderer menu. It's not that a method is better than the other, it's just that it can become critical if you want to match a 3D renderer.

Maya users you can skip the following section (for 3DS Max users) and do directly [here](#):

### 3ds MAX

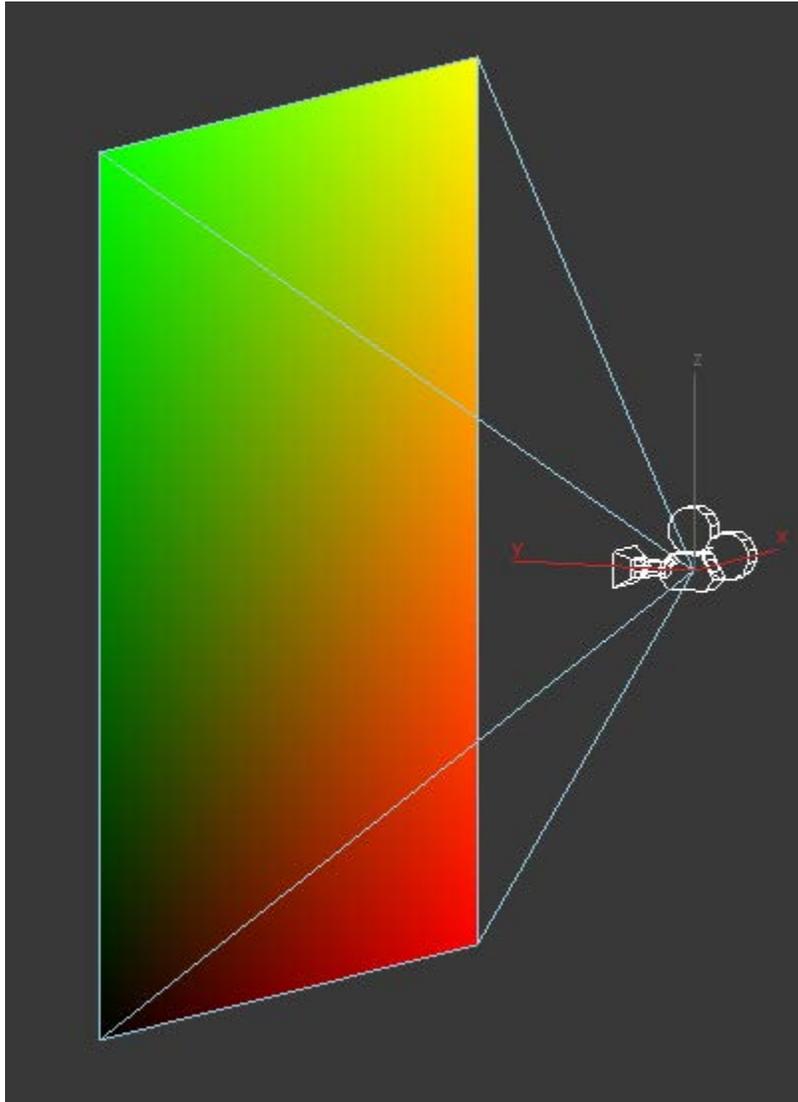
Scenes included: (made in 3ds Max 2015)

ReferencePlane\_MR\_B02.max

ReferencePlane\_MR\_B03.max

ReferencePlane\_VRay\_B03.max

ReferencePlane\_VRay\_B03\_.max



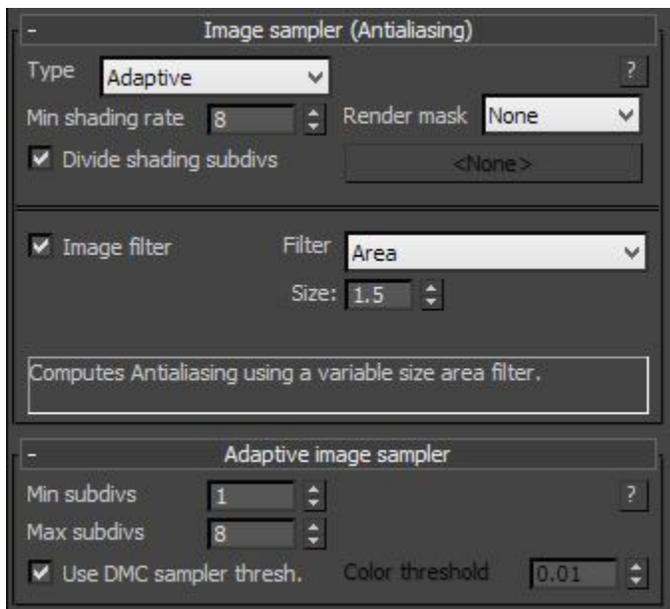
Note this is work in progress, and at techsupport here we did noticed that it seems to change a bit each version, for example are the alpha actually saved if you are 16 bit int versus 32b float, TIFF versus another format... Something we found right away was that you can't save EXR's with UV G-buffer channels from Mental Ray in 3DSMax 2015. There's a memory allocation bug and MR crashes. Rendering to RGBA, however, works, so that's what we did. While we also found a bug using EXR with V-Ray that we reported to Chaos Group.

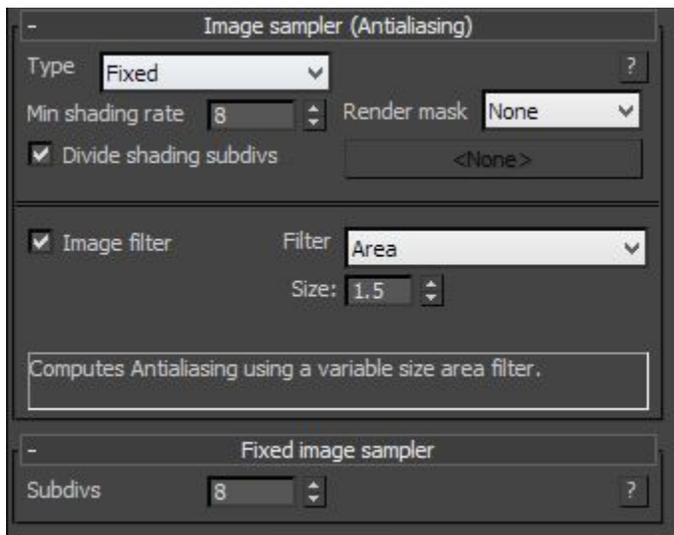
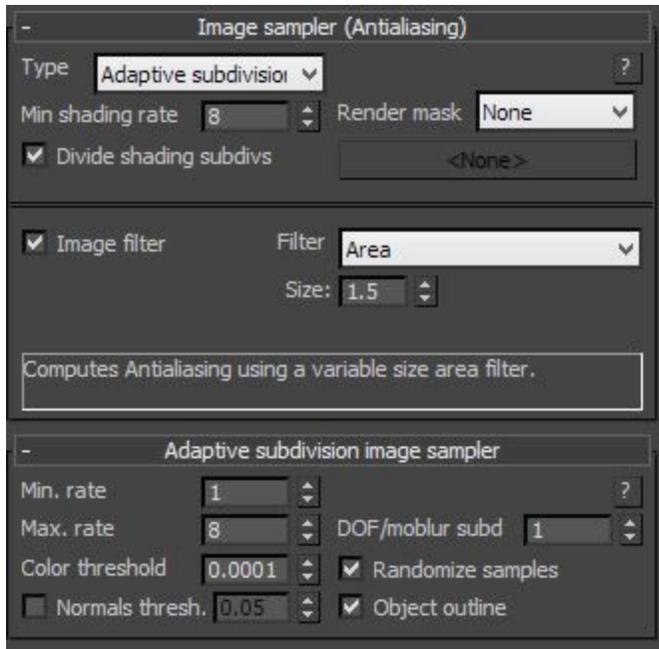
## V-RAY and 3DS Max

For image sampling settings (shading rate), at least for V-ray, there is a tradeoff between accuracy and speed. If you increase the shade rate or subdivs, you'll get a more accurate render, but it's much slower, and it's diminishing returns, so the user needs to decide what's best for them, if the renders are fast enough or good enough with a faster setting, then so be it. A symptom of not enough samples would be a noisy UV mapping.

So not enough sampling – noisy, not enough bit depth precision – blocky

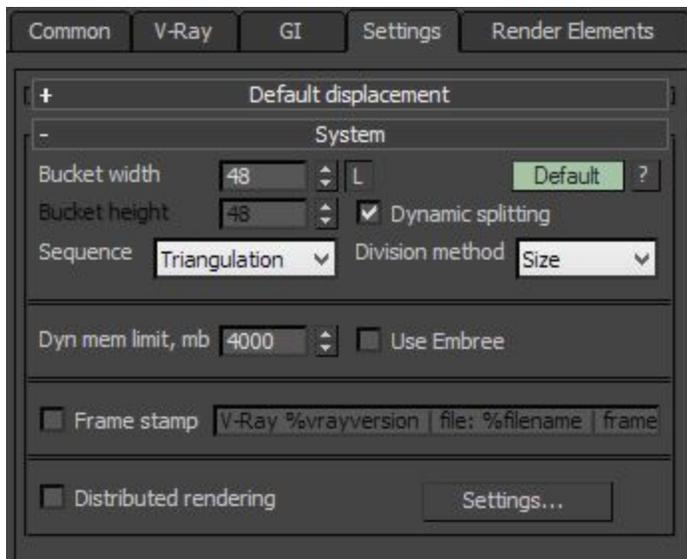
Note that V-Ray for Maya does apply a 2.2 gamma to TIF files. This can't be disabled. The workaround is to fix it in post (using the Interpret Footage panel in AE for example), or just use EXR.





A screengrab for the Render Elements, though we don't recommend using it, it does produce the correct result, it's just a bit of extra effort and data.

Also included below is a screengrab showing where you disable Embree. It doesn't slow down rendering much of anything (because the shading is so simple) but it does make less artifacts in some cases when there is a dense mesh.



We didn't include "Progressive" mode because it appeared inefficient for this sort of thing.

Render Elements

Elements Active     Display Elements

Add ...    Merge ...    Delete

Name	Enabled	Filter	Type
VRayExtraTex_Map #1	On	Off	VRayExtraTex

Selected Element Parameters

Enable     Enable Filtering

Name:

...

Output to Combustion™

Enable

...

Create Combustion Workspace Now ...

VRayExtraTex parameters

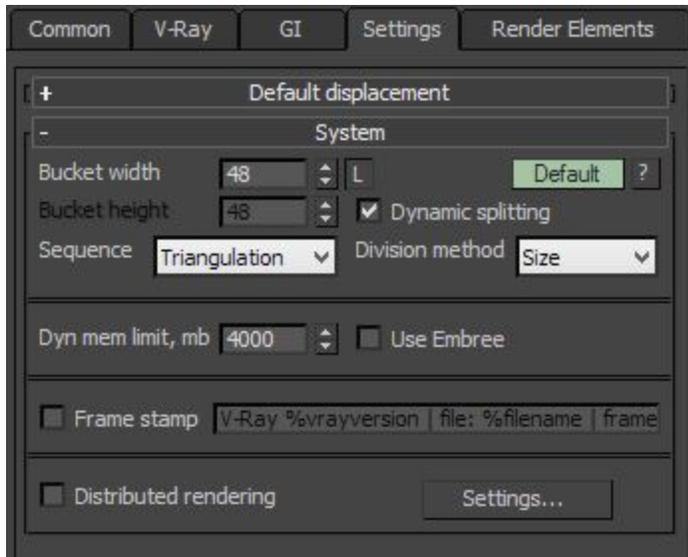
vrayVFB.....

auto update name.....

texture.....

consider for antialiasing.....

affect matte objects.....



Note V-Ray doesn't have a non-raytraced option, but it does offer several sampling options, as well as a choice of using Intel's Embree engine.

The "Fixed" sampling with Embree is terrible. Lots of artifacts. Without Embree, it gives good results, but there are still some artifacts, but it's unclear how much of an issue it will be for you if for example all you is cell phone screen replacement.

The "Adaptive Subdivision" without Embree gives good results if the sample rate is high enough. Low sample rates gets noisy. Embree adds single pixel artifacts.

The "Adaptive" method is similar to "Adaptive Subdivision". We can't see a real difference, in either case you basically set the error floor and sample ceiling.

The "Progressive" method makes a lot of aliasing errors. We currently do not consider really useful for this sort of application.

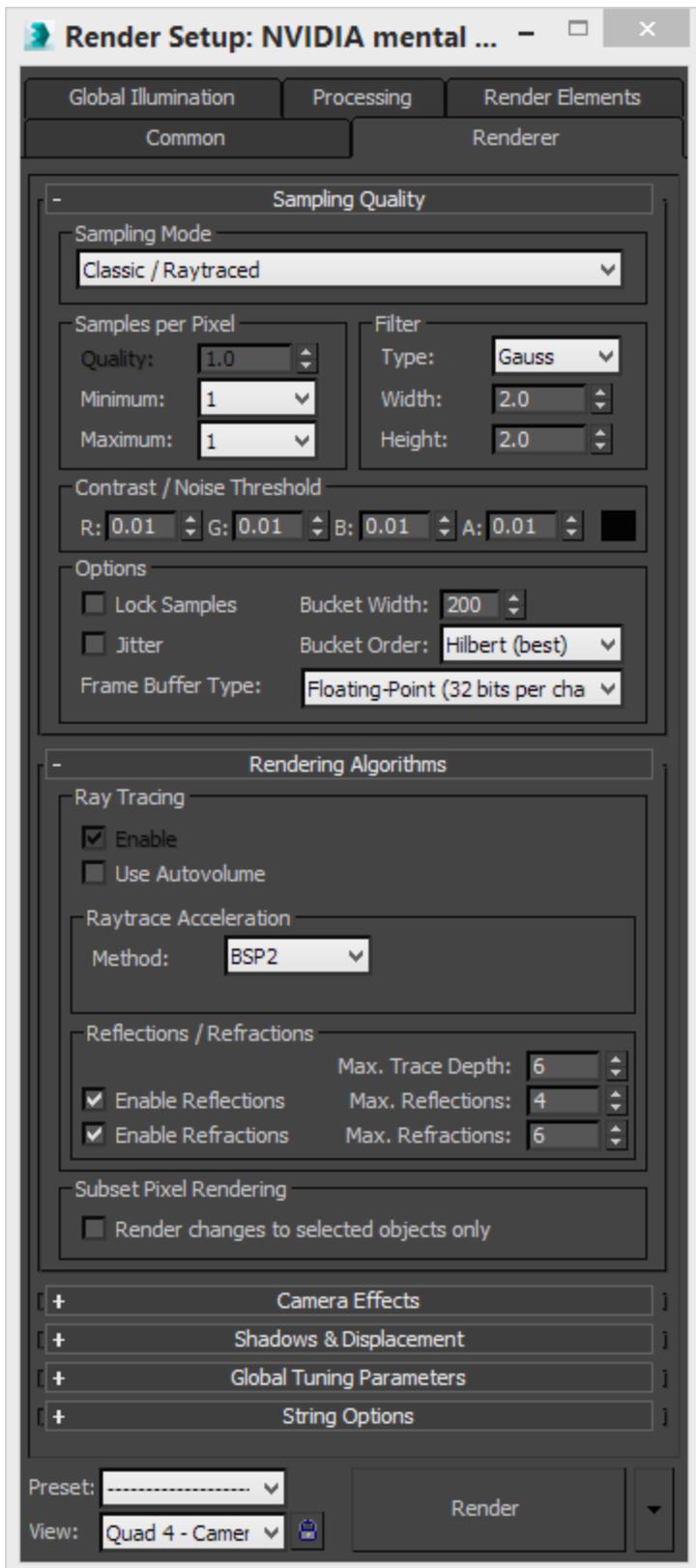


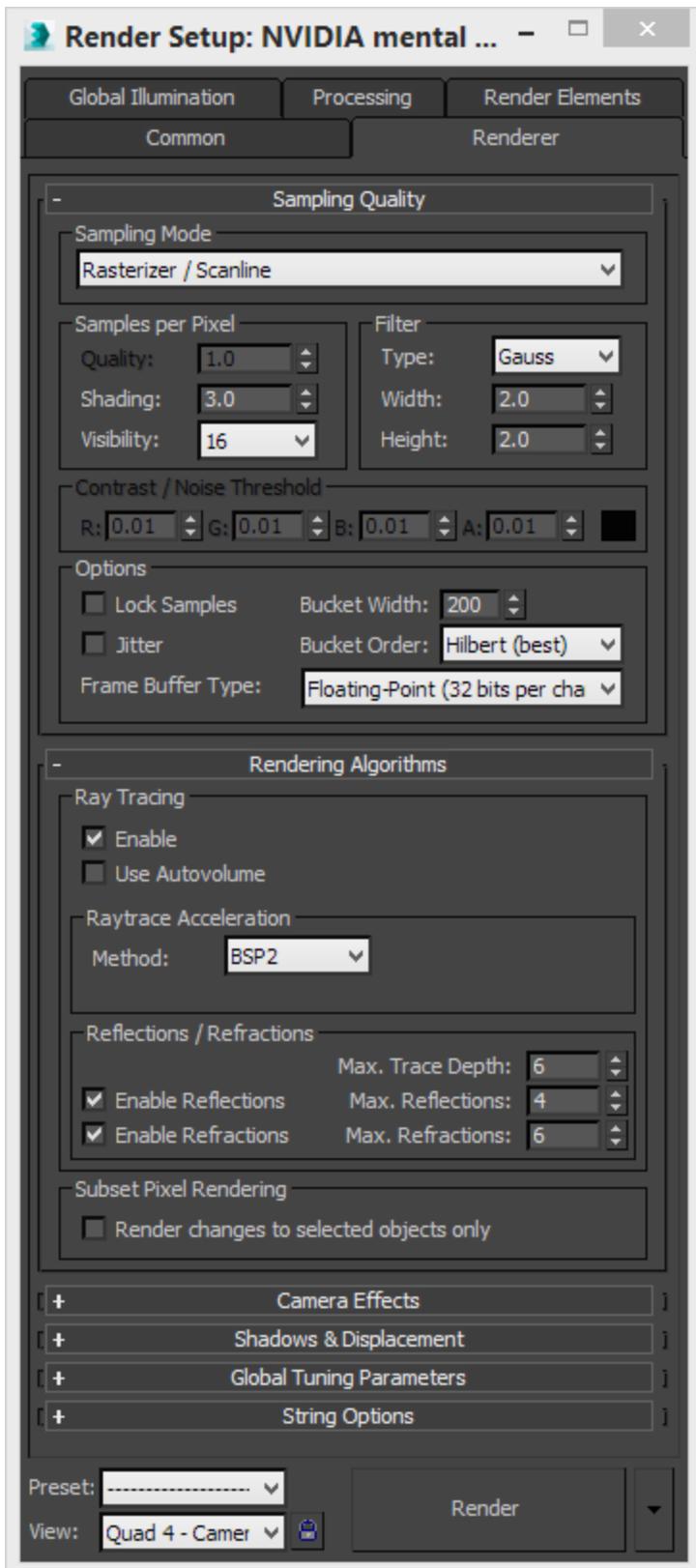
Fixed sampling might be the way to go as it's fast, but Adaptive Subdivision seems to produce the least mean error. The difference is small so it probably does not matter on simple UV like animating a telephone with a UV map screen to apply in post.

### **3DS Max and Mental Ray**

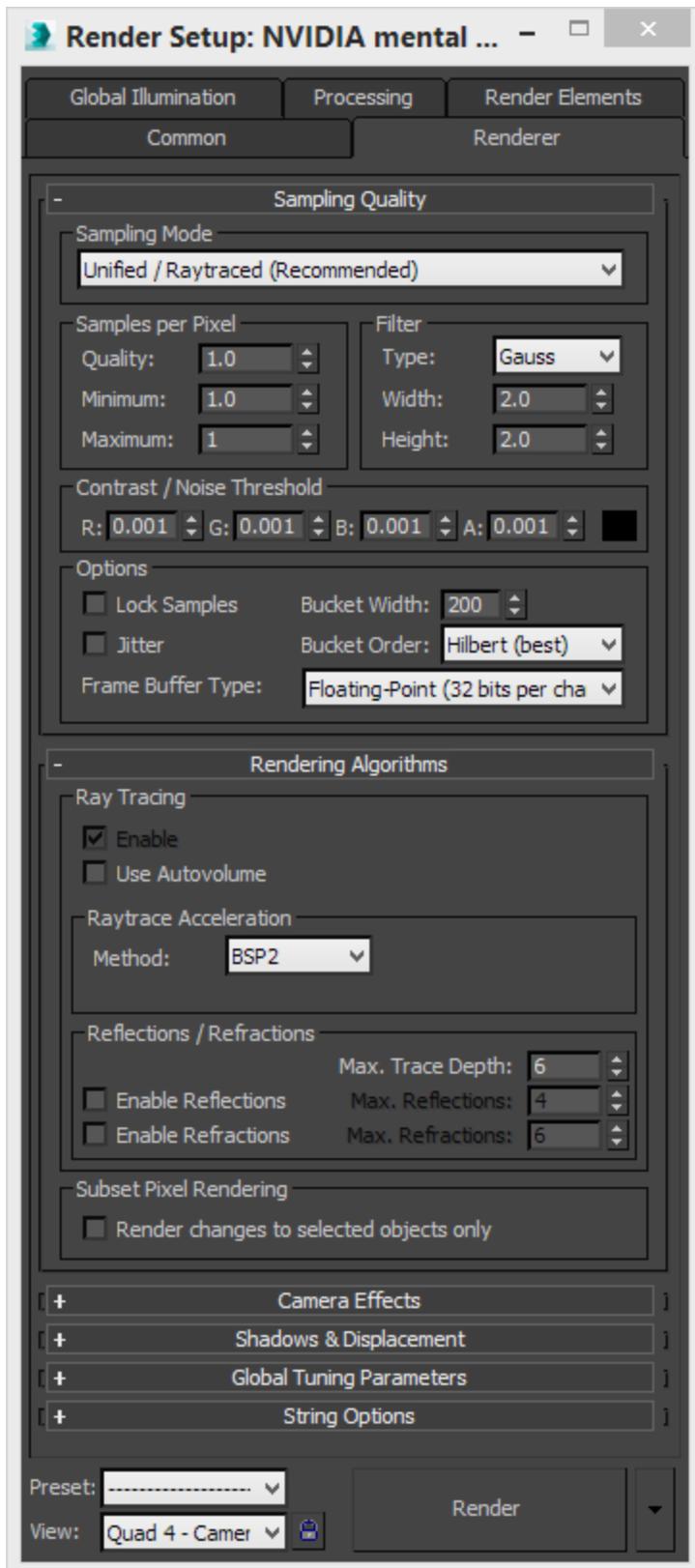
We didn't use the Mental Ray hidden shader "UV Generator". We mention it exists in case you are interested. There are simpler way to do this so no need to discuss this.

Autodesk recommends with MR Classic that you don't set Min and Max to the same value as in the pix below. And we recommend that the Minimum be at least 1.0. Obviously **turn off jitter sampling**. This is one of the most common issue here at techsupport.

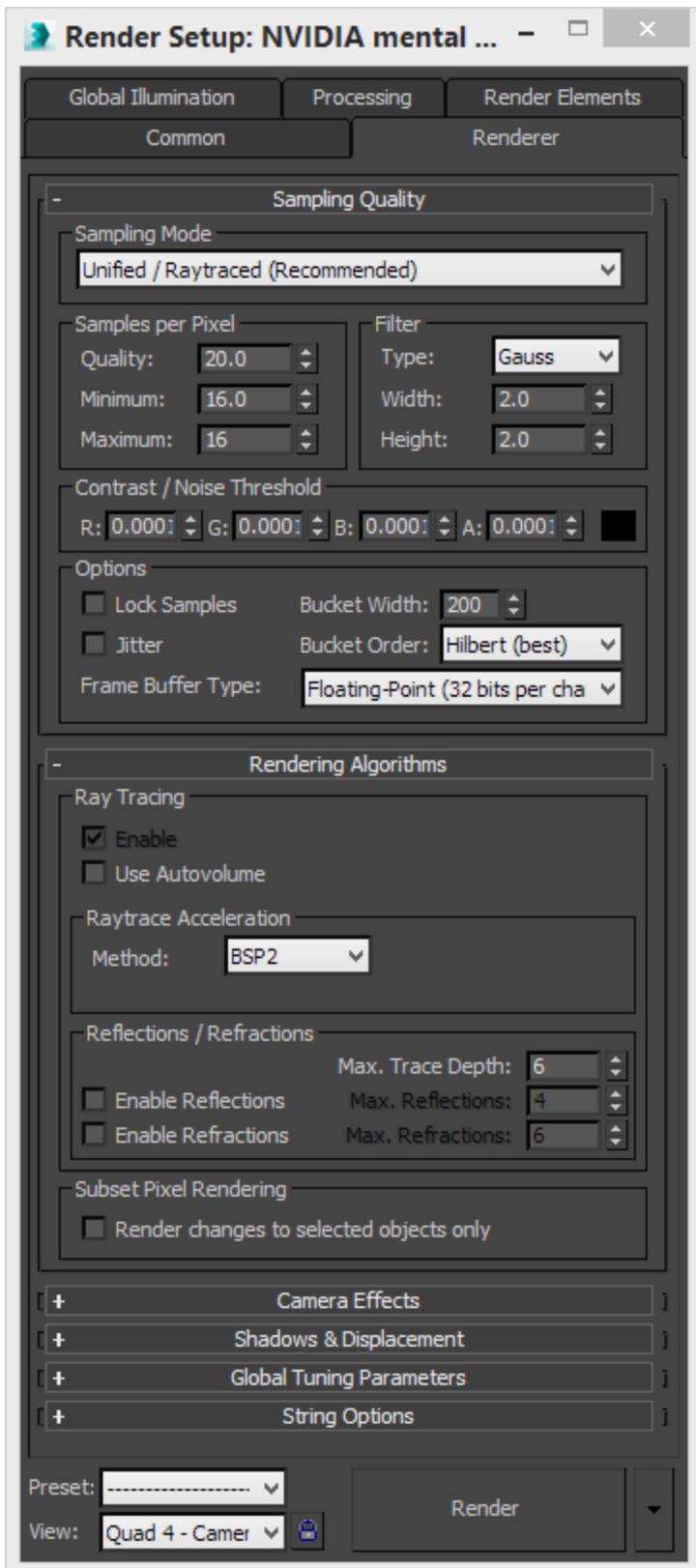




Note the Sampling Rate (1.0) below. Quality in Sampling means the number of samples per pixel. The lower the more noisy or aliased. Unlike what is usually recommended for RGBA (beauty pass renders) you might be better to put Min and Max samples the same value so the UV map is not affected by pixel value contrast. A UV map render does not follow the usual logic for sampling contribution.

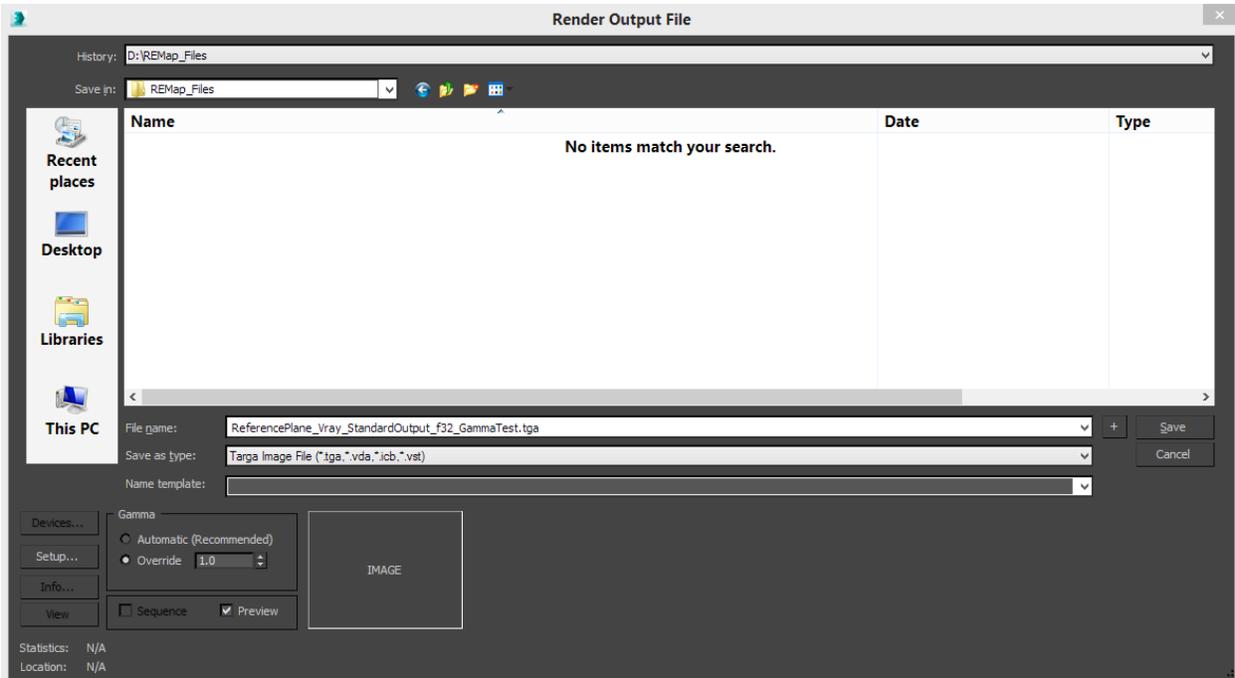


And here 16.



In the Render Output settings, when you select a file output, you get a dialog which contains a section for "Gamma". If you save to a format like TGA or TIF, even 16-bit TIFF, then the default of "Automatic (Recommended)" will apply gamma correction before saving the file. You can override that with the "Override" option. If you use a format like EXR, then either option will result in a linear image.

Be aware that even if you configure 3ds max to use a gamma of say, 2.0 or 2.8, it doesn't matter, the "Automatic (Recommended)" will still save them with a 2.2 gamma. So instead of "Automatic (Recommended)" it should be labeled "Gamma 2.2 (Recommended)".



## MAYA

Scenes included: (made in Maya 2015)

ReferencePlane\_MR\_Direct\_place2dTexure\_A02.ma

ReferencePlane\_MR\_RenderPass\_UVPass\_A04.ma

ReferencePlane\_Vray\_Direct\_place2dTexure\_A02.ma

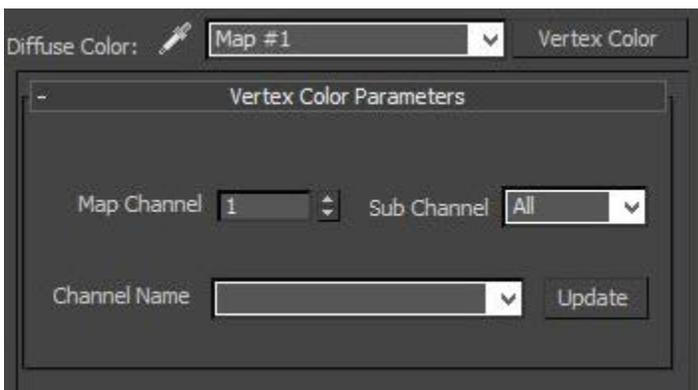
ReferencePlane\_Vray\_RenderPass\_ExtraTex\_A02.ma

There are two fairly straightforward ways of getting the image, just like 3ds max. You either do a normal render with self-illuminated objects textured with the UV values, or you use a render element to output the UV's to another channel.

Below is the material hypershade for the normal render. The place2dTexture's UV's are pumped into the RG of the surfaceShader and assigned to the object. The nice part is that this works for any renderer in Maya and it shows up in the viewport. It's obviously pretty simple to set up too. This would be our recommended workflow, hands down.



If you are using the UV's in the model, you can use the Vertex Color map.

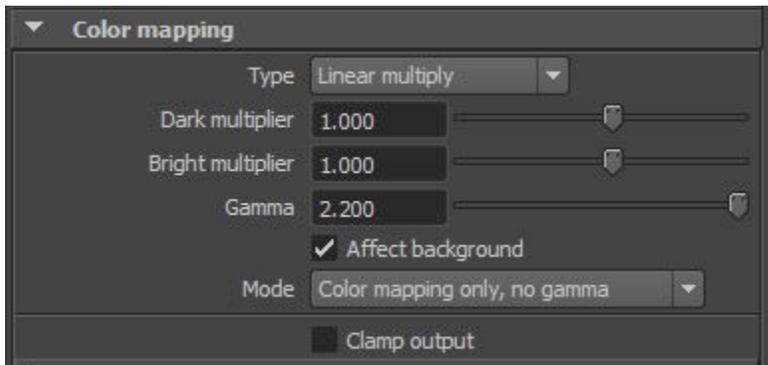


Placed in the Diffuse Color slot of a self-illuminated material, it will quickly and easily give you whatever map channel you need. Just set the Map Channel to be the UVW channel you want to output.

This method has the advantage of showing up correctly in the viewports, while the UV Generator does not. It also has the advantage of being a much faster shader. It's roughly 25% faster to render than UV Generator.

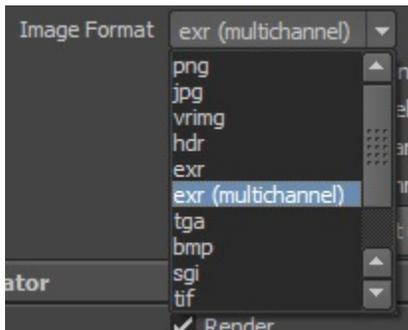
Now careful about this little color management panel if you export UV or MV via RGBA images!

## Maya and V-Ray



Note: The Gamma defaults to 2.2, but the Mode defaults to ignoring it.

If you render to EXR multi-channel, you will need to regenerate the UV map with alpha within a tool before using RE:Map.



As an alternative to hypershade, if you want to have render passes through the renderers themselves, though, that's an option. Below is the setup for Vray Render Elements. However you need to use the Extra Tex element because there is no UV element. So you have to convert the UV's to a texture for this to work.

Render Layer UVPass

Render Using V-Ray

VRay Common VRay GI Settings **Render Elements** RT Engine

Available render elements

- Atmospheric Effects
- Background
- BumpNormals
- Caustics
- Coverage
- Custom Color
- DR Bucket
- Diffuse
- Extra Tex**
- GI
- Light Select
- Lighting
- Material ID
- Material Select
- Matte shadow
- Multi Matte
- Normals
- Object ID
- Raw GI
- Raw Light
- Raw Reflection
- Raw Refraction
- Raw Shadow
- Raw Total Light
- Reflection
- Reflection Filter
- Reflection glossiness
- Reflection hilight glossiness
- Refraction
- Refraction Filter
- Refraction glossiness
- Render ID
- SSS
- Sample Rate

Added render elements

- Y vrayRE\_Extra\_Tex

Add Remove

Save Import

Enable/Disable All

Separate folders

Separate RGBA

Use referenced

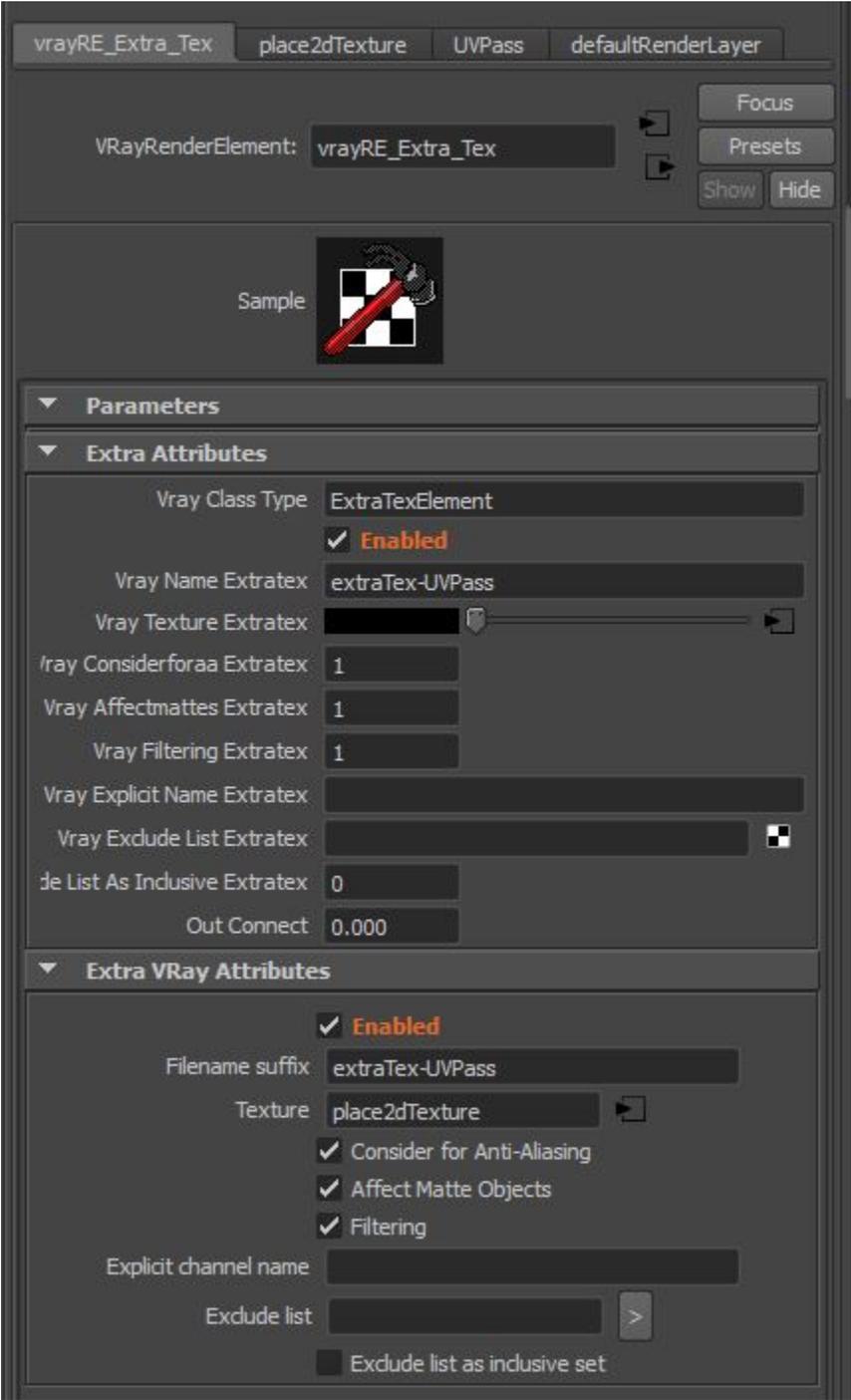
Unfiltered fragment method Best coverage

Render element separator .

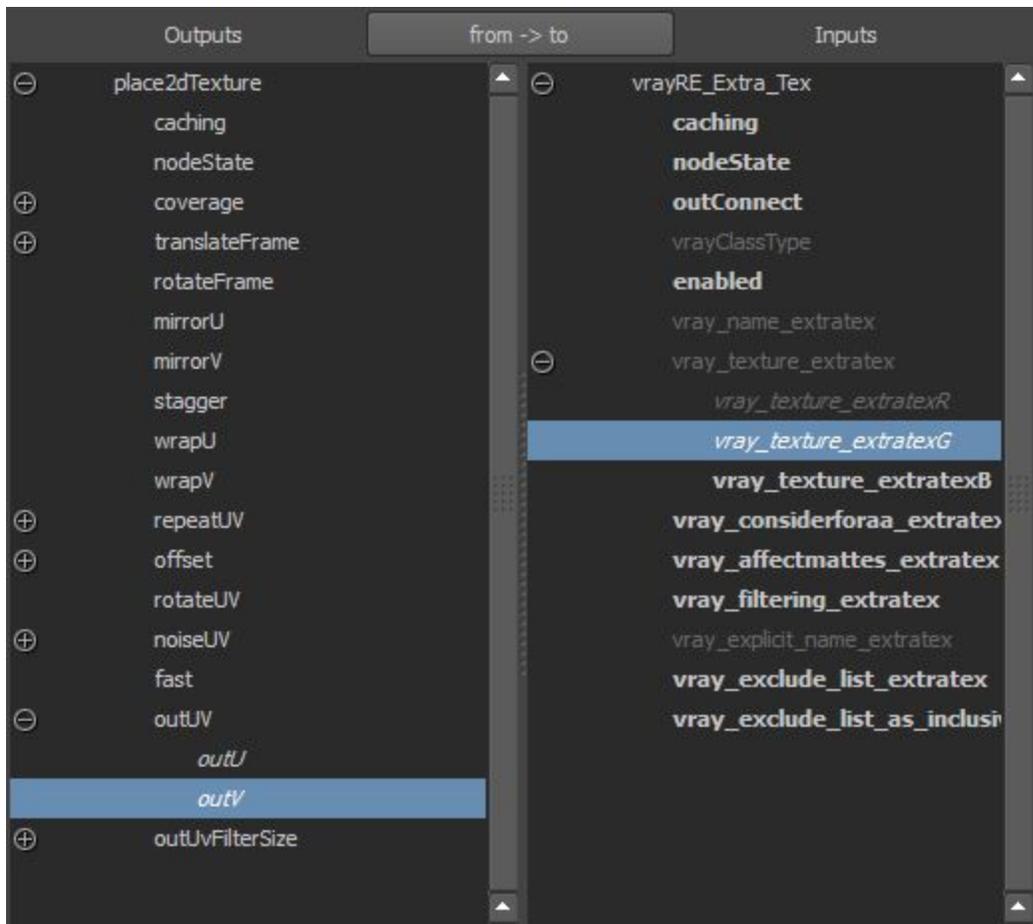
Deep merge mode By render ID

Deep merge zdepth thresh. 1.000

So what we see below is using the place2dTexture as the Texture for the Extra Tex element. We have set the name to be "extraTex-UVPass" but you can name it whatever you want.



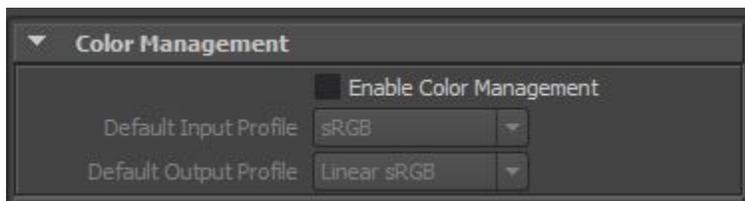
So what we see below is using the place2dTexture as the Texture for the Extra Tex element. We've set the name to be "extraTex-UVPass" but you can name it whatever you want.

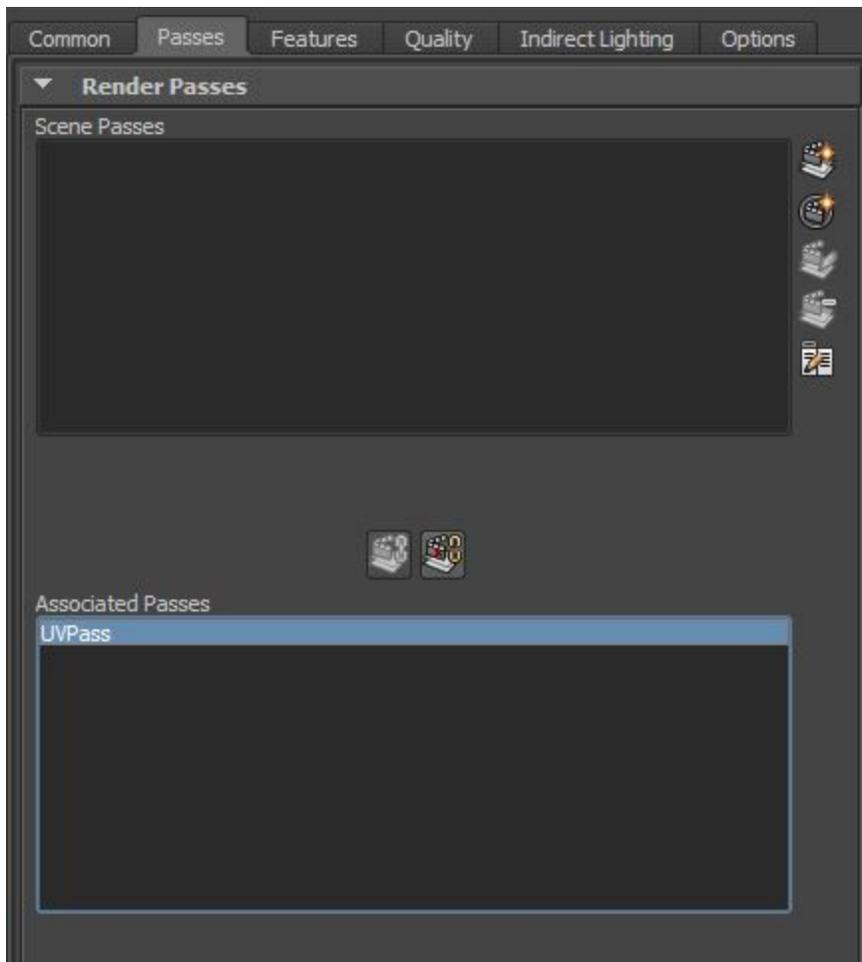


## Maya and Mental Ray

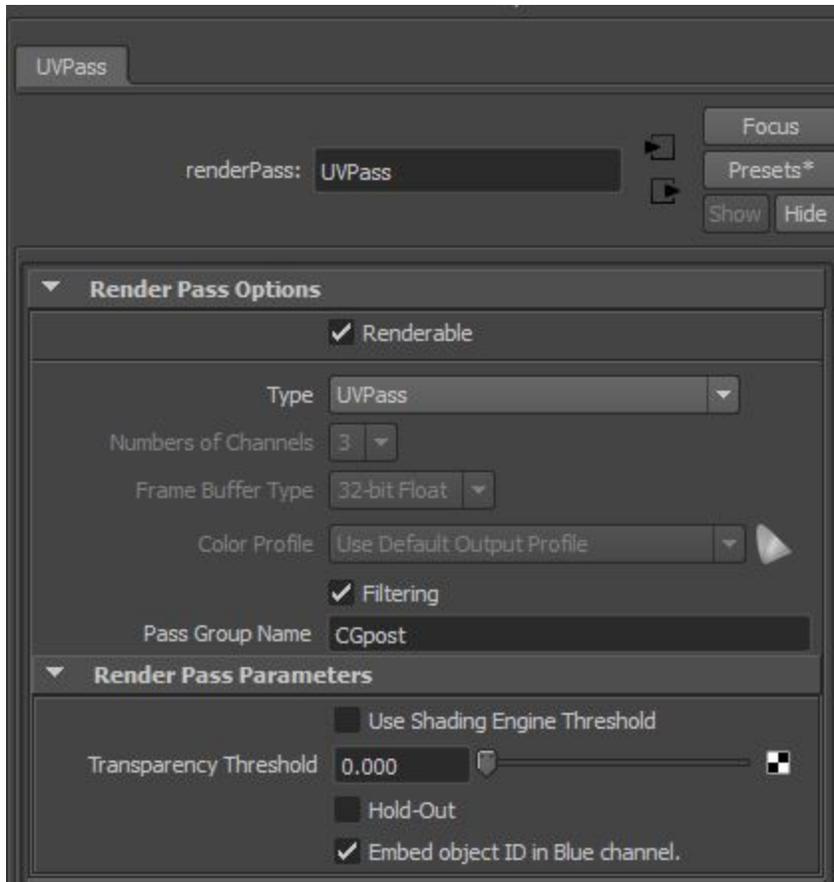
For Mental Ray, it's simpler to create a render pass. Just create a render pass and associate it with the current render layer.

Be aware of the Color Management tab (must be OFF).

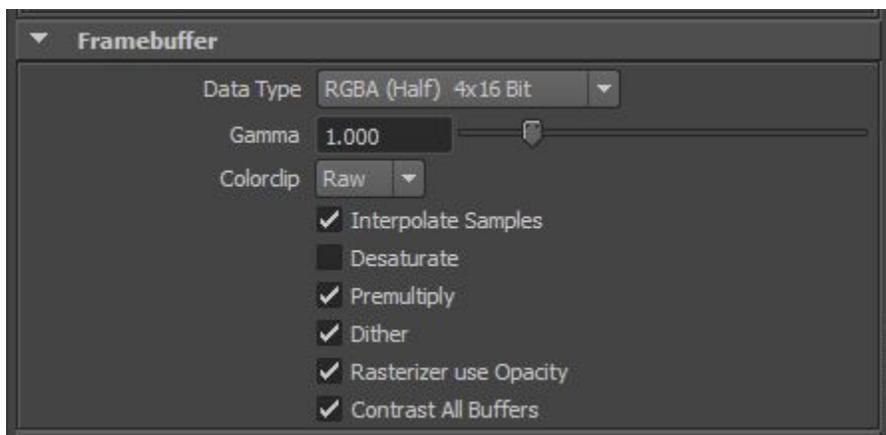




Then in the attributes for that pass, set the Type to UVPass. Turn on filtering and then you are good to go. Just remember to use batch rendering.



For Mental Ray, there's a Framebuffer setting too, but it defaults to float and linear gamma, so that's good.



Note, we don't recommend using half float for UV maps unless it's for mapping small images (thumbnails / icon size). In practice half float are about 13 bits + 1 of precision between 0 and 1.0 – 8097 steps is not much if you want sub-pixel precision when rendering a slanted triangle. Already 16 bit short provides more precision – in theory 65535 steps, although note AE in a 16 bit project actually works with 15 bits + 1 of precision (0-32768). So in AE with 16 bit integers

images (short) there is a lot of precision value to actually maintain your project in floating point. All this depends on application. Flat screen replacement is a lot more tolerant.

In the version we are currently testing Mental Ray in Maya, it appears to only works on EXR's. Other formats are not supported. Now note that half float have very little precision outside the 0 to 1.0 range so like 16 bit integers (shorts) it is a good idea particularly as the image gets bigger to use a respectful Displace value.

