Linear Workflow (LW) with Autodesk Maya Mentalray

In my experiences as a teacher, technical director and CG pipeline developer, most Maya artists don’t understand linear workflow (LW) and would rather ignore it. That would be fine if Maya followed this proper rendering workflow by default, but it does not. Without this rendering and display pipeline you can't make correct lighting decisions, particularly for photo-realistic rendering. Even edge anti-aliasing doesn't look right on a non-gamma boosted image viewer.

Many CG production facilities lock down their color pipeline so new artists are not tempted work without it. At Industrial Light and Magic we called it "gamma wars". Many of the newly hired artists, including myself, didn't like the "look" of a properly boosted display. Renders looked flat and unsaturated. Eventually the veteran color scientists at ILM convinced the protesting newbies with irrefutable facts, until the next batch of new hires started the "gamma wars" all over again.

Linear Workflow (LW) with 3D applications pertains to color managing the viewer, image textures and constant color values to display renders with the correct contrast. Some people call this contrast, gamma. A linear workflow pipeline includes proper display gamma as well as making sure incoming textures and constant colors are in a linear color space. If you want to know more, my favorite color scientist, Charles Poynton’s Color and Gamma website explains the gamma issue in great detail.

This is a step by step tutorial on how to set up Linear Workflow (LW) in Autodesk Maya when rendering with Mentalray. The content is specific to versions prior to Maya2016 or Maya2015Extension. Color Management which was added to Autodesk Maya 2011, allowed artists to quickly setup a LW for Mentalray. Prior to Maya2011, setting up LW was far more convoluted and varied widely. For Mentalray LW is now mostly automated, but there are some relatively minor issues which still need to be addressed manually. You should be aware that the Maya software renderer does not support this color management. Readers should be familiar with the following Maya interface panels: RenderView, RenderSettings and Hypershade.

Included with this tutorial is a zipped Maya2014 project with a typical background image and four Maya scenes. There are two scenes per renderer. One is for 8-bit (png) output and the other is for float (exr) output. These scenes have the appropriate color management settings for interactive display and will produce correct output using interactive and batch rendering.

Figure 1 is a composite using Gimp the free paint package of the foreground render blended with the normal mode over the original background. As you can see the render layer blends seamlessly with the background even though the background is an sRGB image and we are working in a linear workflow (LW).

Figure 3

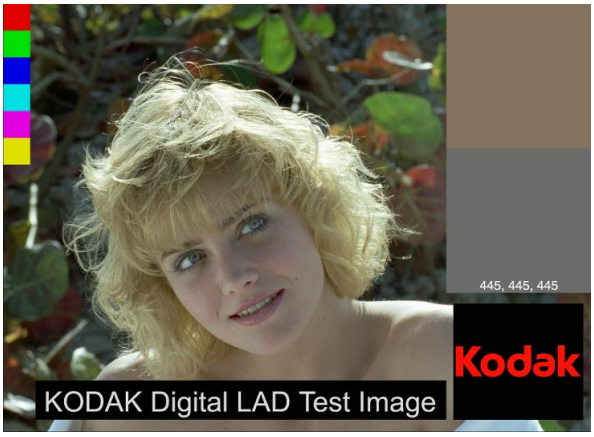


Figure 1

Figure 2

The CG layer (Figure 3) includes six constant color rectangles, two constant color spheres, two constant color rectangles, a gradated ramp rectangle. The matching RGB values can be acquired from the background image (Figure 2) with a paint app like Photoshop or Gimp or the interactive Maya screen color picker. The CG layer in this case is rendered with Mentalray which has a option to un-premult. This un-premult option results in an RGB/Alpha png image that blends with no edge fringing because the RGB channels are actually divided by alpha. This is not required when rendering openexr images since that format assume a premulted output.

The next two close-up crop images show the difference in the falloff of CG lighting with LW and without LW. It's subtle in this case but still evident. I can tell Figure 4 is the LW render because the light edge is more abrupt matching how light actually reacts to a rounded surface in the real world. Figure 5 is the non LW render. You can see why some artists might prefer the non LW "look" but this is not photo realistic.

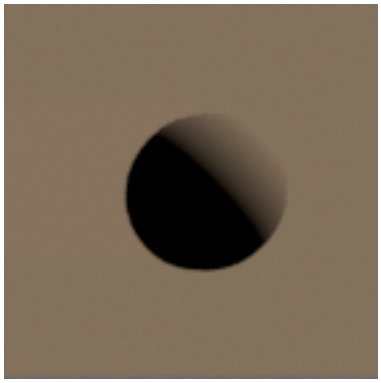


Figure 4

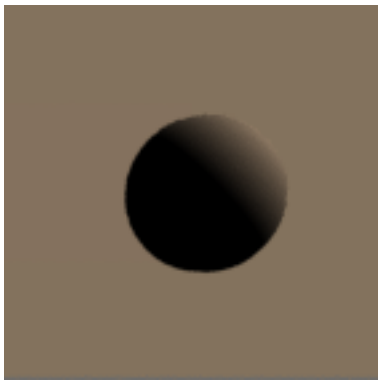


Figure 5

8-Bit Output with Mentalray

This section covers linear workflow (LW) for **8-bit** Mentalray output for integer image formats like png, jpg or tiff. These image types require a gamma 2.2 or more commonly a sRGB color output profile.

First, we set up the appropriate output attributes as in Figure 6. Make sure to Enable Color Management toggle is turned on in the RenderSettings Common tab otherwise no color management is done. The Default Input Profile specifies how incoming texture maps are processed prior to rendering. **sRGB** implies that the textures are gamma inverted in order to comply with linear workflow (LW). More about this later on in the section discussing the Maya **File** Hypershade node. When the Output profile is set to **sRGB,** final output as well as interactive renders, will have a gamma boosted color profile.

To summarize set the Enable Color Management toggle on, Default Input Profile to sRGB and Default Output Profile to sRGB as in Figure 6. Set the Image format to PNG to get 8-bit output. Notice I'm using the handy File name prefix feature where <Scene> is one of the special keywords to tell Maya to embed the name of the Maya scene in the output image name.

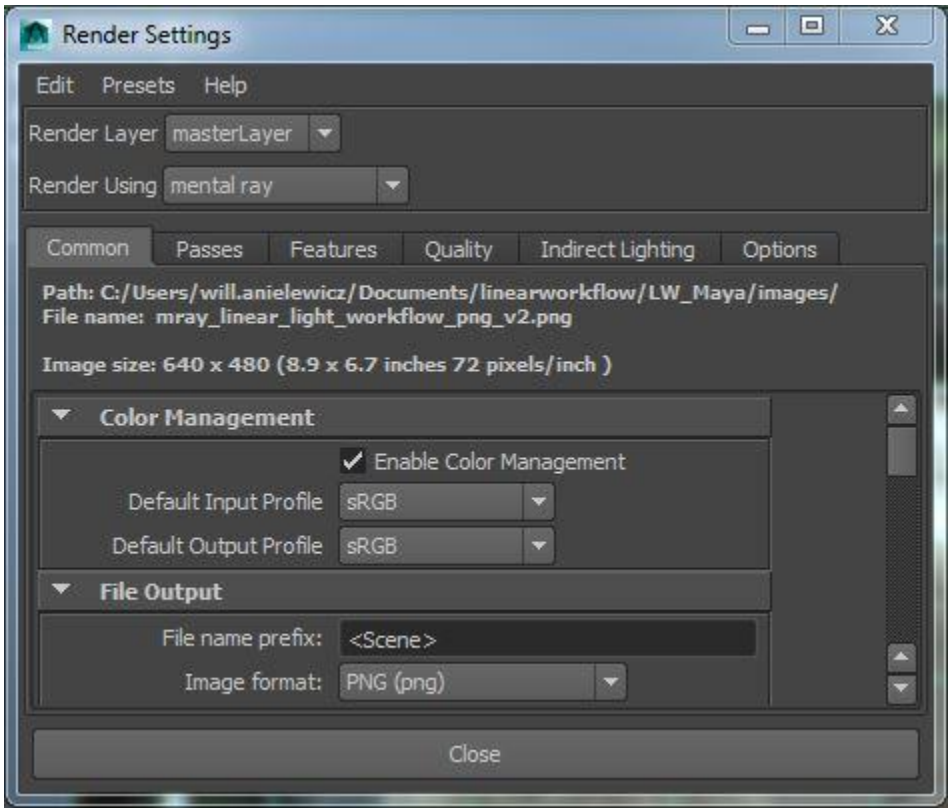


Figure 6

Next we need to set up Maya's interactive render viewer Renderview for LW. Figure 7 shows how to access RenderView's ColorManager from the "Display" pull-down.

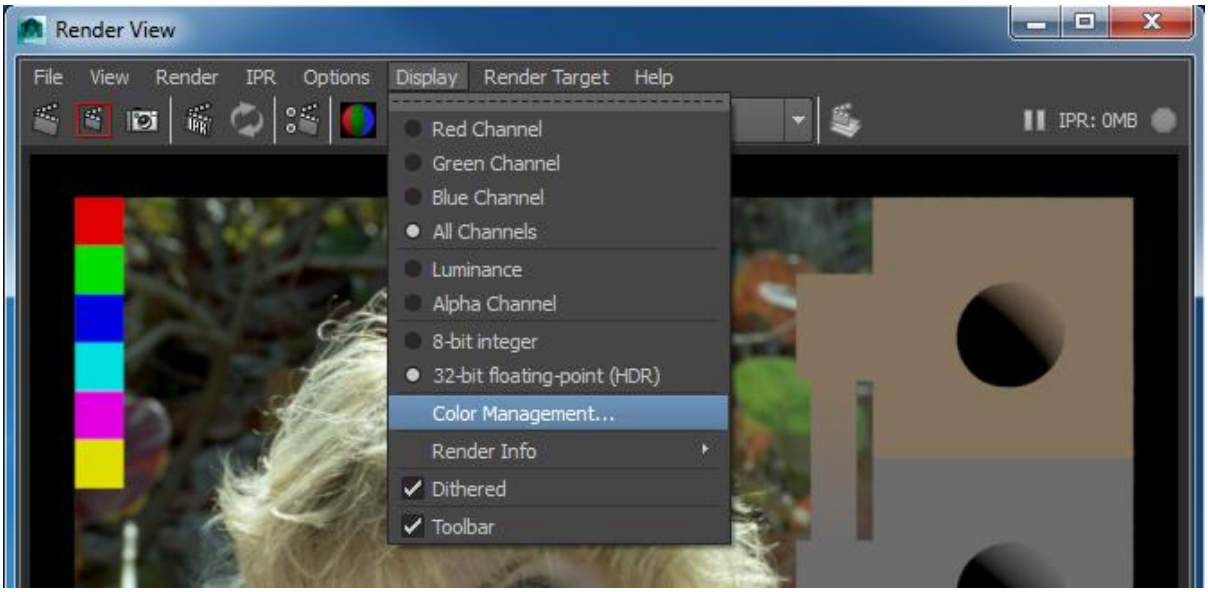


Figure 7

After clicking this option you will see the defaultViewColorManager attribute panel on the right in the Maya attribute editor panel. Set them as I have in Figure 8.

Image Color Profile is set to **sRGB** and Display Color Profile is set to **sRGB**. The labelling of these attributes is admittedly a bit confusing but just go along with my suggestions. Basically we are telling the Renderview ColorManager not to boost the display gamma since the output itself will already have the **sRGB** color profile applied.

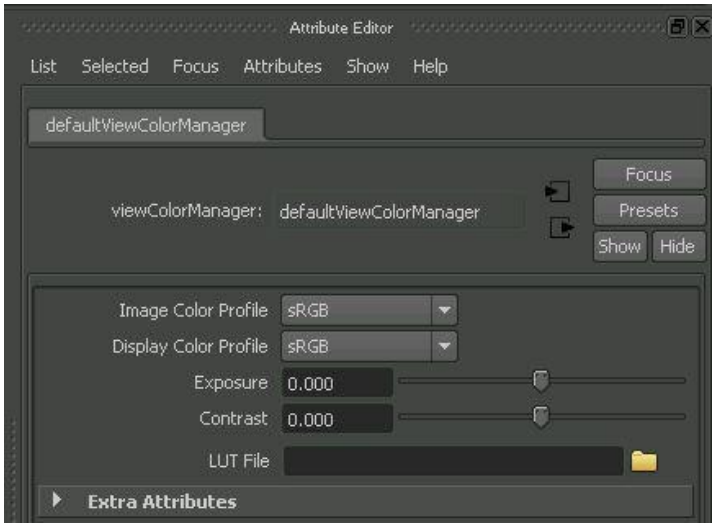


Figure 8

Float Output with Mentalray

Now let's describe the LW settings required for rendering a float image format like Openexr illustrated in Figure 9. As with 8-bit set the Enable Color Management toggle **ON** in the RenderSettings Common tab. The Default Input Profile is **sRGB**. Like 8-bit output this states that the incoming textures require gamma inversion in order to be in linear color space. The Output profile is set to Linear **sRGB,** so the final output will not be gamma boosted. This is expected for float images, but it implies that the viewer will need to be gamma boosted. Set the Image format to OpenEXR to get float output. I also usually set the Image compression to PIZ which results in smaller file sizes. Again the handy File name prefix keyword <Scene> tells Maya to embed the name of the Maya scene in the output image name.

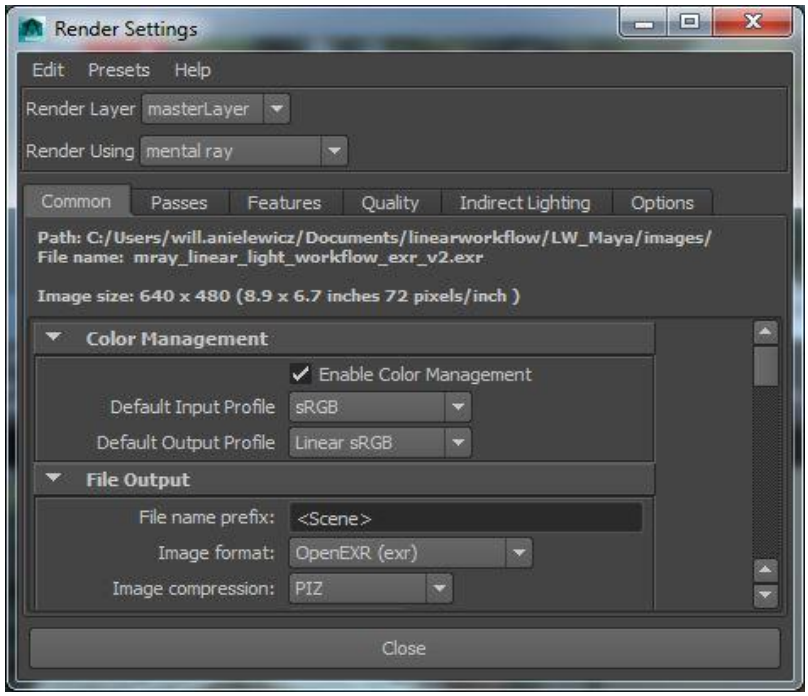


Figure 9

Set the RenderView Colormanager attributes as in Figure 10; Image Color Profile is **Linear sRGB** and Display Color Profile is **sRGB**. Although the labelling of these choices are in my opinion a bit confusing, Linear sRGB states that the will be gamma boost.

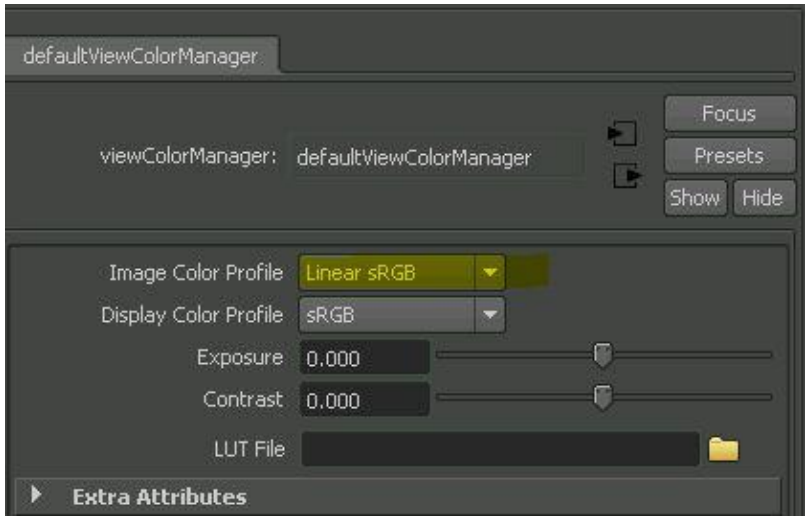
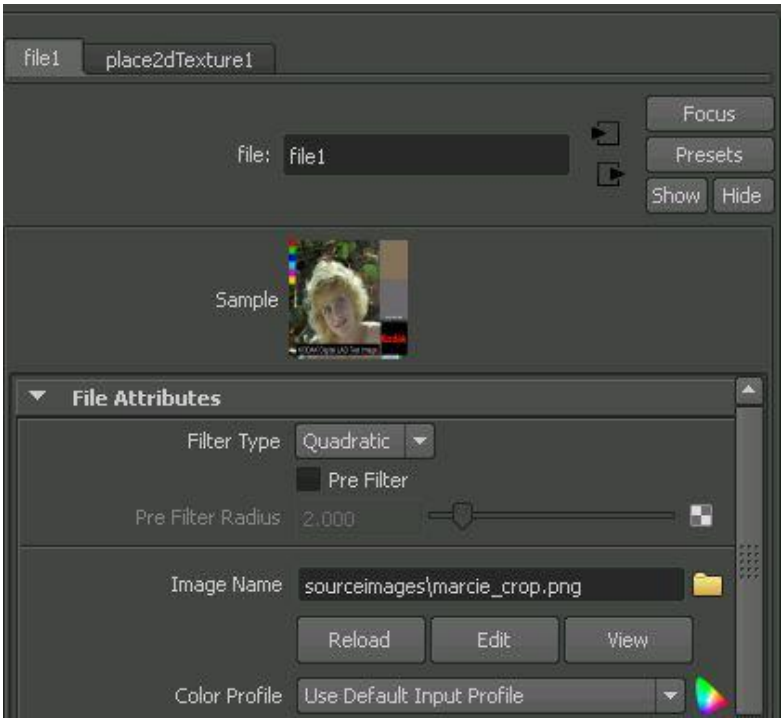


Figure 10

You are now ready to do float output and interactively view them in a linear workflow (LW).

File Node settings for Mentalray

Next, let's cover some of the hypershade nodes in these scenes. The background texture (a sRGB gamma boosted png file of the famous Kodak Marcie image widely used in color grading) is a surfaceShader with a file node attached to the color attribute. Note the choice in the Color Profile pull-down of the file node. It is set to "Use Default Input Profile". This is directly connected to the RenderSettings Common tab ColorManagement Default Input Profile which is set to sRGB. This means that the texture will be automatically gamma inverted by 0.4545 at render time. In fact Maya will used an sRGB inversion color profile which will more accurately match the sRGB profile of the incoming texture file.



Hypershade solid color RGB values for Mentalray

This section is the same for both Mentalray. Taking a look at the hypershade material network for the brown sphere which has a Maya Phong material attached, we notice there is a gamma node directly connected to the color attribute of the phongShader. The RGB values of the color attribute of the gamma node were eye drop picked from the original Marcie png image in a paint program. Those RGB values were typed into the Value attribute of the gamma node. We then set all three Gamma values to 0.4545. This inverts the specific RGB color so that it fits into our LW setup and matches the background color. This would be even more accurate if there were a hypershade color node that included the means to specify sRGB, but its close enough for most eyes. The grey sphere is done in the same way. For single colors this is a more efficient approach than using a ramp node and a gamma node. For ramp nodes you have to attach a gamma node to the output in order to invert the gamma. It's unfortunate that Maya does not have a Color Profile pull-down in these nodes just like the file node. If they had, this manual aspect of setting up LW in Maya would be unnecessary.

Figure 42

Hypershade network ramp RGB values Mentalray

This section is the same for both Mentalray. For ramp nodes you have to attach a gamma node to the output in order to invert the gamma. Again it's unfortunate that Maya does not have a Color Profile pull-down in these nodes just like the file node. If they had, this manual aspect of setting up LW in Maya would be unnecessary.

Figure 13