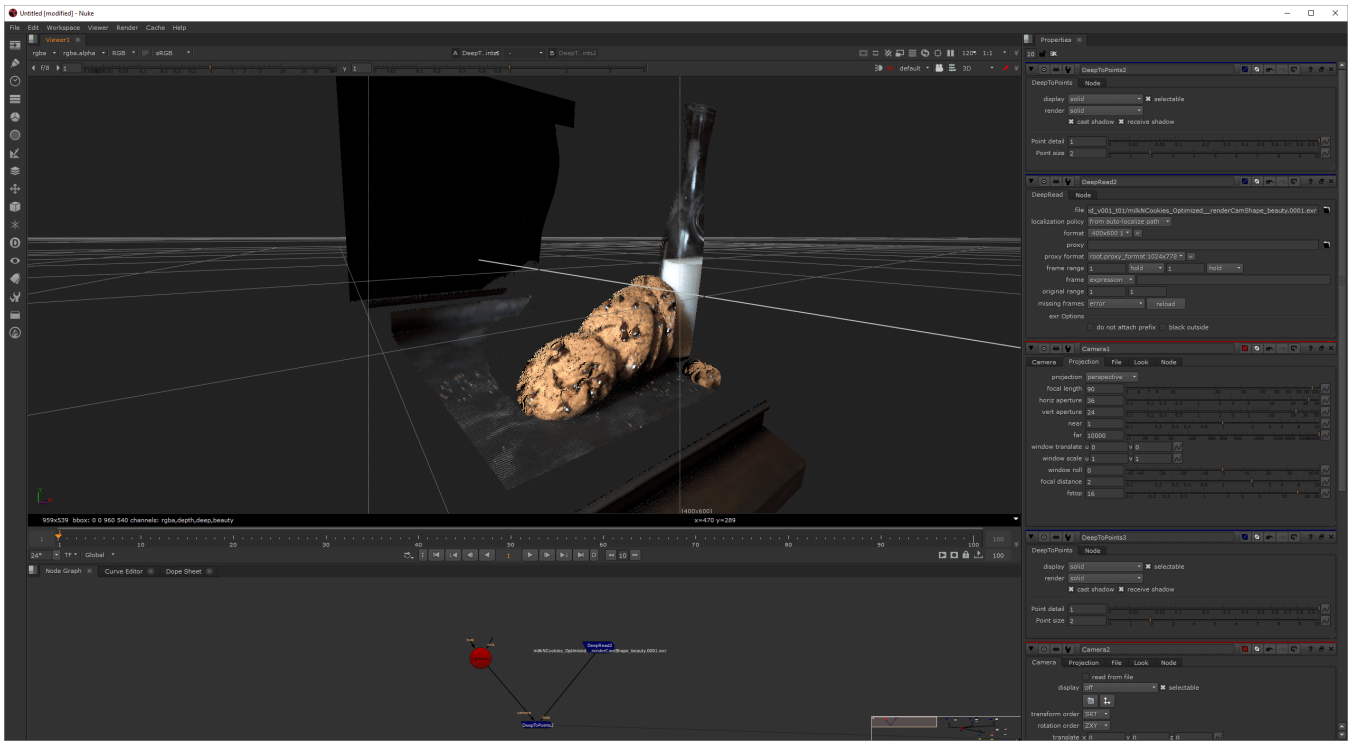
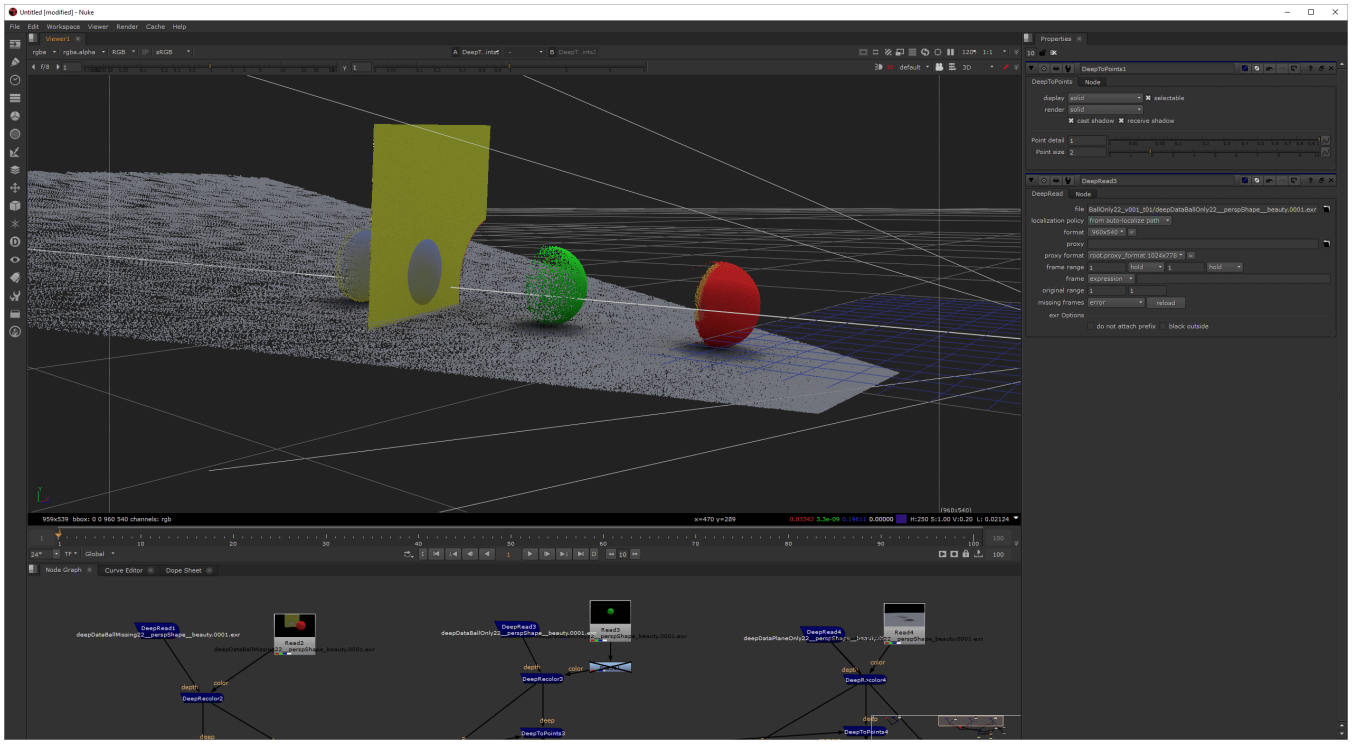
DeepEXR



The Milk and Cookies Tutorial rendered to DeepEXR and converted to points in Nuke.



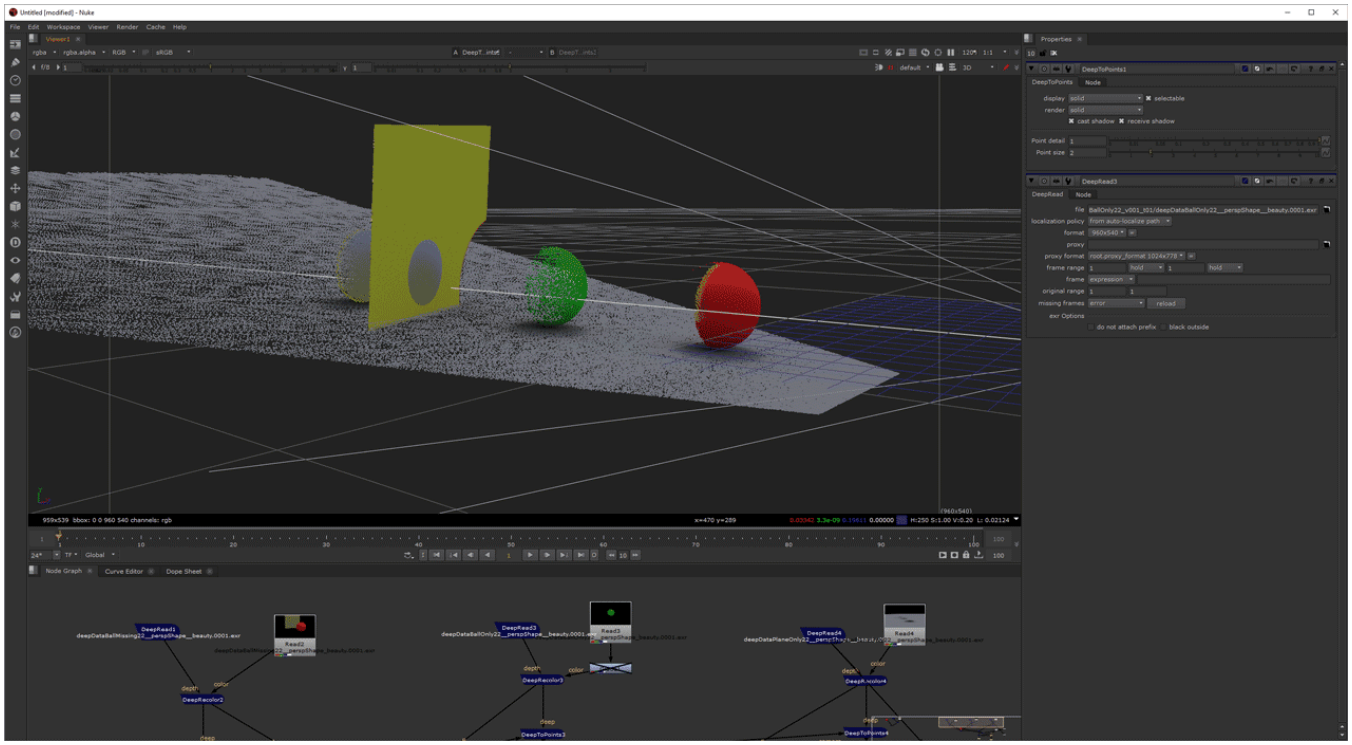
A simple scene rendered in layers and composed in Nuke using DeepEXR

DeepEXR files can store sample data into the format. This means where a ray from the camera intersects objects, the file can store this point and more associated data. This allows external packages reading the data to display this 3D information along with the rest. Be sure and use the same camera settings when viewing the data or it could end up looking skewed.

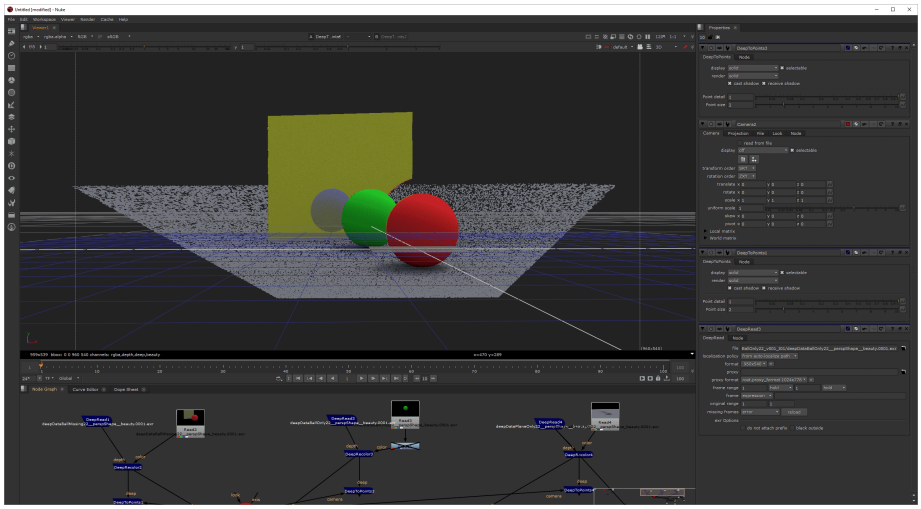
Deep data has many benefits. It allows artists to control complex layering in 3D space. You can re-project color information later should it change and adjust it using compositing nodes. You can apply better quality camera effects like depth of field and even composite objects into something like a fog field or other volumetric effect without re-rendering. Or maybe you need to nestle a baby bird into some tall grass surrounding them, you can do that with deep data.

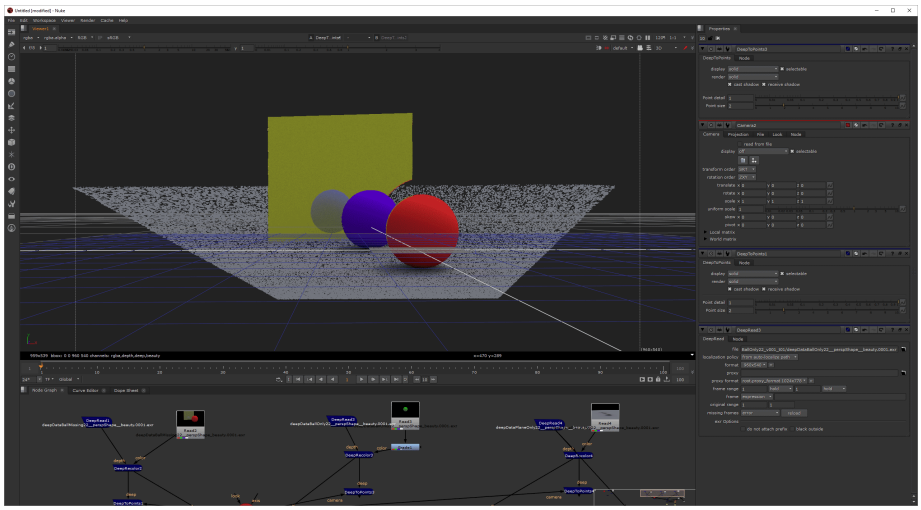
For quality, you can use the render option for deepshadowerror to control the merging of samples where larger values merge more samples but may introduce precision errors in compositing.

Note in the image directly above, opaque objects cut out the object behind them while transparent objects (objects with partial presence) can reveal what's behind them like the blue ball behind the yellow wall. (Since the wall was rendered at the same time, the blue ball color is baked into the wall.) Note that the green ball, despite being opaque, does not cut into the yellow wall, this is because it was rendered separately with primary visibility off and then merged in Nuke along with the floor that has the shadows baked in.

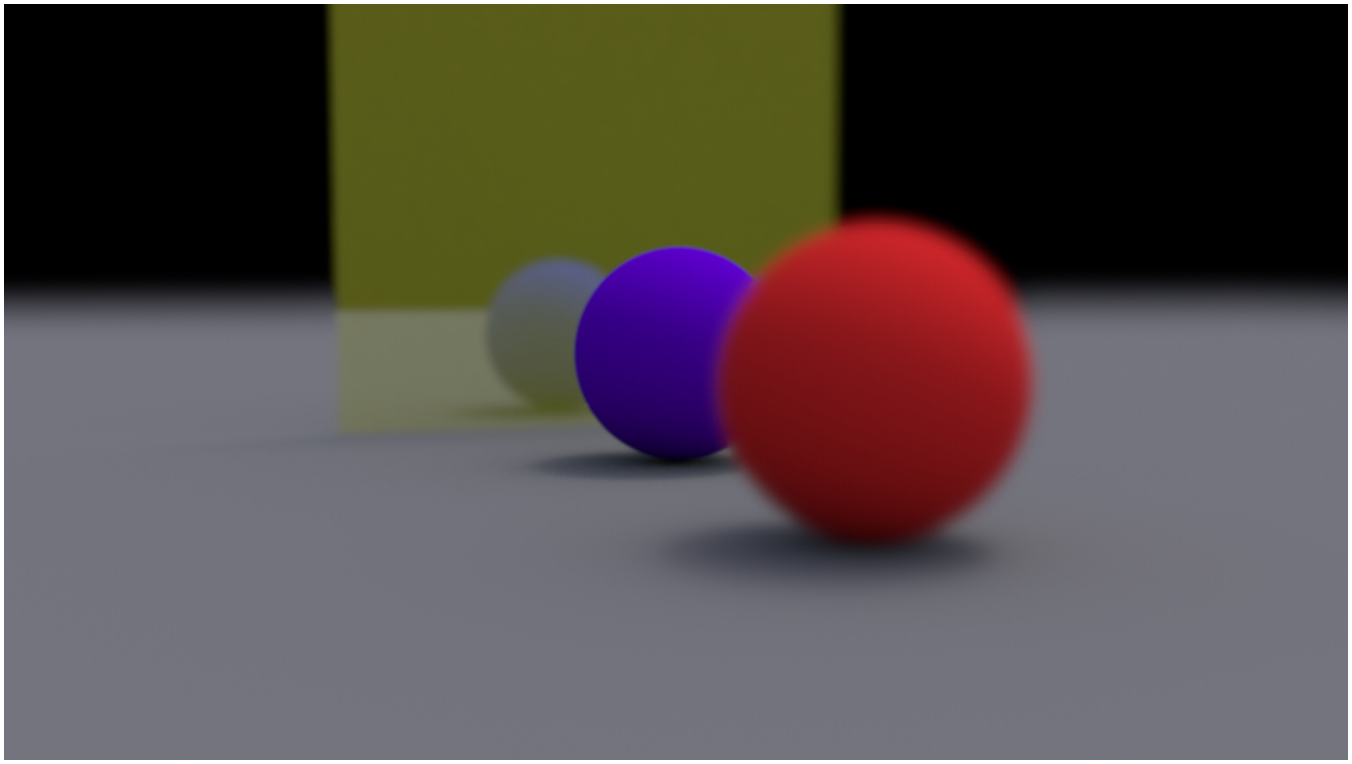


Here's an example of an object changing color and being merged back. We turn the green ball purple.





And below is a render with the color change and added depth of field using zdefocus in Nuke.



DeepEXR

Below are the common settings available to artists using RenderMan in bridge products.

Storage

- Tiled: Image is saved in tile formatted blocks of data
- Planar/Scanline: Is stored as a scanline format, this is preferred for reading in Nuke

Type

- Half Precision (16-half)
- Float Precision (32-bit float)

Typically Half is good enough for color AOVs and passes. Float is recommended for Data AOVs and passes like Z depth or P

Compression

We handle merging samples internally and automatically to provide the best balance between size and correctness, as such the options below are what is available to control final size of the output frames.

- ZIPS is default and recommended for compositing workflow as it is stored as a single scanline
- ZIP is also useful but slightly slower in compositing packages at bundles of 16 scanlines
- You can find out more about the less common choices on the Wiki page for OpenEXR