Denoise JSON Config

The denoiser uses a JSON config file to read in the options and files that are passed in on the command line. A typical JSON config file might look something like this:

```
{
    "primary": [
        "/prod/shot_0001/images/Scene.View_Layer.0001.exr"
    ],
    "aux": {
    },
    "config": {
        "asymmetry": 0.0,
        "flow": false,
        "debug": false,
        "output-dir": "/prod/shot_0001/images/denoised",
        "passes": [
            "color",
            "alpha"
        ],
        "parameters": "/opt/pixar/RenderManProServer-25.2/lib/denoise/20973-renderman.param",
        "topology": "/opt/pixar/RenderManProServer-25.2/lib/denoise/full_w1_5s_sym_gen2.topo"
    }
}
```

Let's take a look at what each section is.

Primary

This should be the list file paths to the primary beauty image that holds the necessary AOVs for denoising (see Denoiser AOVs).

AUX

This should be a list of auxiliary passes/images that you would like to also denoise. These are broken up into different passes like diffuse or specular. An example of what this section might look like:

```
    "aux": {
        "diffuse": [
            {
                "paths": [
                    "cornellbox2__perspShape__directDiffuse.0001.exr"
                ],
                "layers": [
                    "directDiffuse_front"
                ]
            },
            {
                "paths": [
                    "cornellbox2__perspShape__directDiffuse1.0001.exr"
                ],
                "layers": [
                    "directDiffuse_top"
                ]
            }
        ],
        "specular": [
            {
                "paths": [
                    "cornellbox2__perspShape__directSpecular.0001.exr"
                ],
                "layers": [
                    "directSpecular_front"
                ]
            },
            {
                "paths": [
                    "cornellbox2__perspShape__directSpecular1.0001.exr"
                ],
                "layers": [
                    "directSpecular_top"
                ]
            }
        ],
        "albedo": [],
        "irradiance": [],
        "alpha": []
    },
```

In the diffuse section, we have two AOV outputs we would like to denoise. "paths" is the list of paths to the images. "layers" is the channel name(s) that we want to denoise as the diffuse pass. If there are multiple channels of the same type, in the same pass, you can provide a comma-separated list in "layers".

Config

Asymmetry

Controls the asymmetry value, 0 is the best quality, and higher values encourage the denoiser to avoid overblurring, leading to weaker denoising. Note, not all networks support this.

Flow

When enabled, we use OpenCV to do image-based calculation of the apparent motion of objects between consecutive frames, rather than the provided motion vector AOVs in the input files.

Debug

If enabled, will include all AOVs in the output image (including those that were in the primary image) for debugging.

Output-dir

The path to the output directory where the denoised images will be written to.

Passes

The list of passes to denoise. This can include color, diffuse, specular, alpha, albeo, and irradiance. For example, if we did not need alpha to be denoised, we can leave off "alpha" from the "passes" list.

Topology

The path to the .topo file to use. The file describes the neural network topology. For single frame denoising you'll want to use **full_w1_5s_sym_gen2.topo**. For motion blur, for best results, it is recommended to use **full_w7_4sv2_sym_gen2.topo**. However, this requires at least 7 frames in your sequence (three before and three after the current frame). Note, both these topology files do not take asymmetry into account. If you want to use the asymmetry control, you can use **full_w1_5s_asym_gen2.topo**.

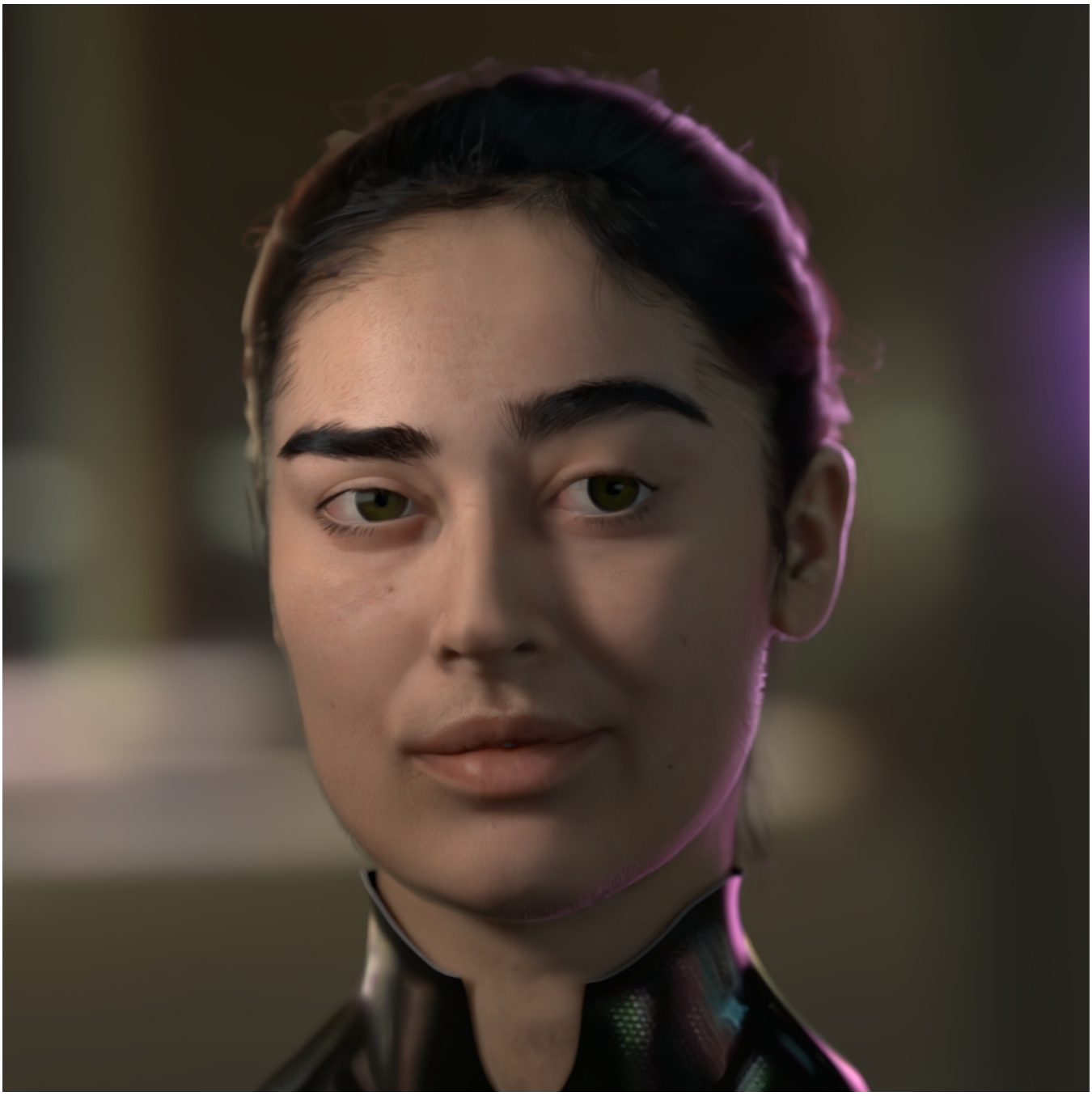
Parameters

The path to the .param file to use. The file contains the neural network weights. If you're using the **full_w1_5s_sym_gen2.topo** topology, this should be paired with **20973-renderman.param**. For **full_w7_4sv2_sym_gen2.topo**, use **20970-renderman.param**.

Example Scene

Let's take a look at an example. Here's our undenoised/primary image.

This image was rendered with 0 min samples, 16 max samples, and pixel variance at 0.015. If we use the JSON file from the top of the page, the denoised output we get looks something like this:

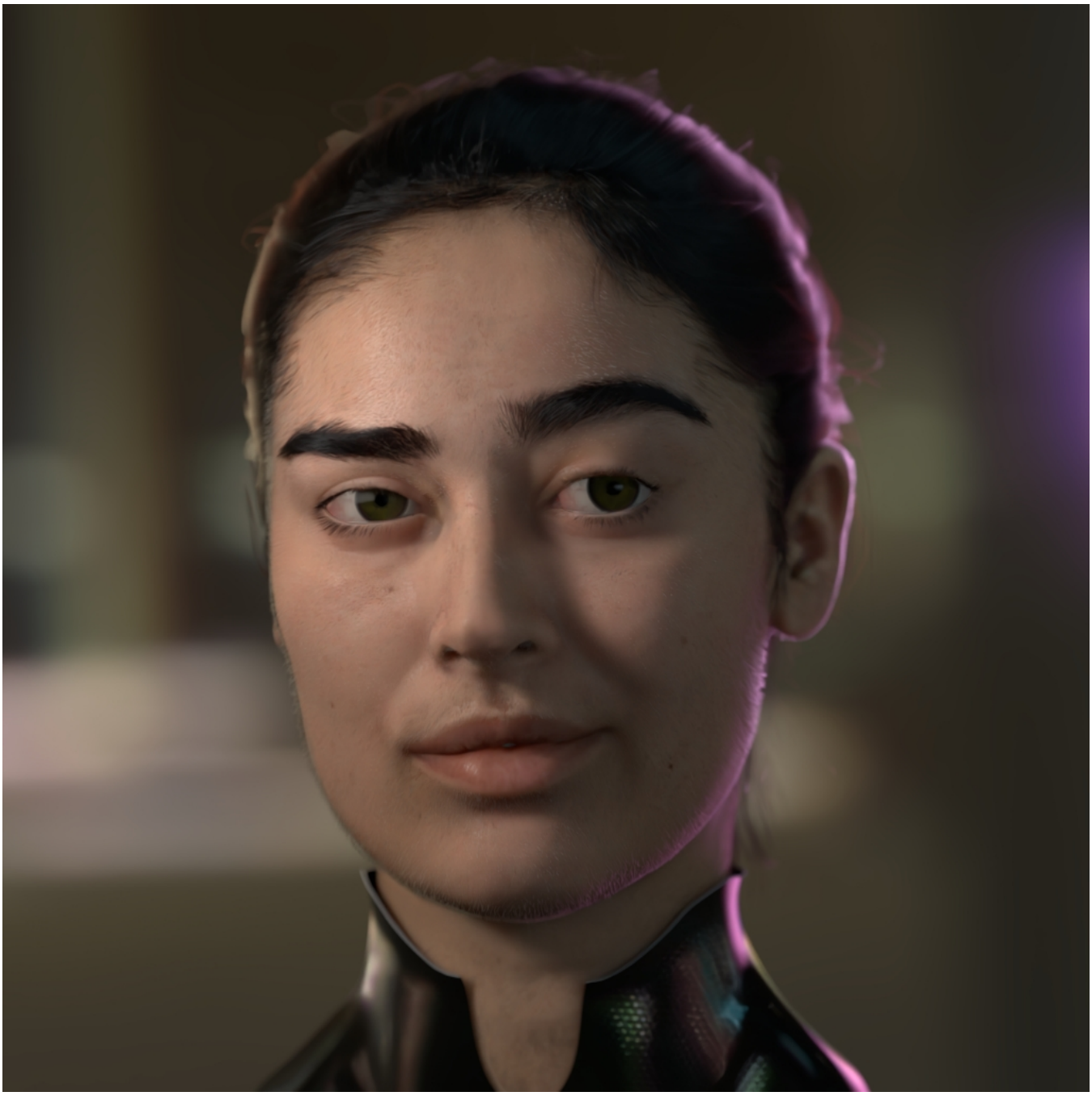


We can improve the denoising a little bit by asking the denoiser to denoise the specular and diffuse, separately. We do this by changing "color" in the passes list to "diffuse" and "specular". Ex:

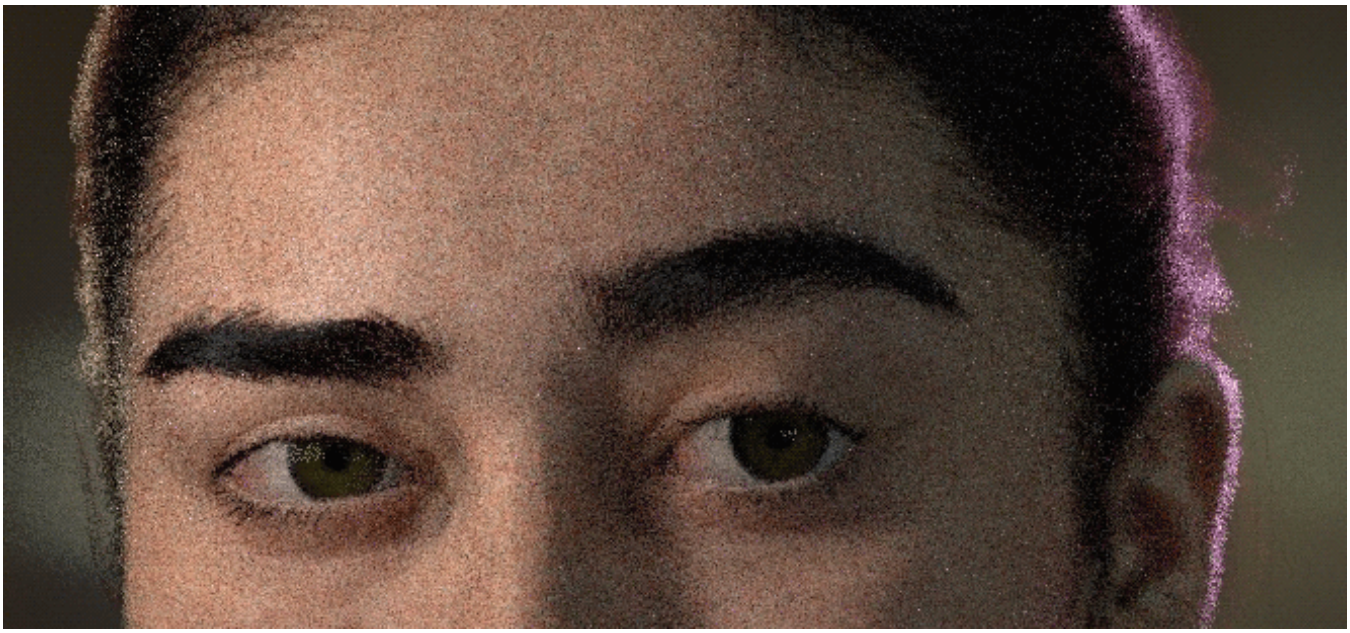
```
        "passes": [
           "diffuse",
                      "specular",
           "alpha"
       ],
```

The denoiser will denoise specular and diffuse separately, and then combine the two to output the final color. **This is the default behavior for all bridge products.**

Notice that you can see more of the specular highlights on the facial hair, just above the left eyebrow.



Illustrating the above example as a sequence of images helps highlight the subtle detail.



Light Groups

To denoise AOVs with light groups, we can use the same primary image, but pass our light group AOV image as part of auxiliary "color" passes. Here's an example of the undenoised light group AOV:

The LPE that generated this image looks like this:

```
"color lpe:C[DS]*[<L.'key_CloseShape'>O]"
```

"key_CloseShape" is the name of our light group. To denoise this, we can use a JSON file that looks like this:

```
{
    "primary": [
        "/prod/shot_0001/images/Scene.View_Layer.0001.exr"
    ],
    "aux": {
        "diffuse": [],
        "specular": [],
        "albedo": [],
        "irradiance": [],
        "alpha": [],
        "color": [
            {
                "paths": [
                    "/prod/shot_0001/images/Scene.View_Layer.rim.0001.exr"
                ],
                "layers": [
                    "beauty_rim_RightShape_key_CloseShape"
                ]
            }
        ]
    },
    "config": {
        "asymmetry": 0.0,
        "flow": false,
        "debug": false,
        "output-dir": "/prod/shot_0001/images/denoised",
        "passes": [
            "color",
                        "alpha"
        ],
        "parameters": "/opt/pixar/RenderManProServer-25.2/lib/denoise/20973-renderman.param",
        "topology": "/opt/pixar/RenderManProServer-25.2/lib/denoise/full_w1_5s_sym_gen2.topo"
    }
}
```

Notice that we added the path to the AOV to the "aux" list, under "color". The "layers" list shows the name of the channel in the image we want to denoise.

If the channel name for your AOV consists of R,G,B, you can use the layer name "default" instead. For example:

```
        "color": [
            {
                "paths": [
                    "/prod/shot_0001/images/Scene.View_Layer.rim.0001.exr"
                ],
                "layers": [
                    "default"
                ]
            }
        ]
```

The denoise result would be written to the path /prod/shot_0001/images/denoised/aux-color. If you had separated out the diffuse and specular into separate files, then they should go into the "diffuse" and "specular" lists, respectively, instead of "color". The denoised result of our example looks something like this: