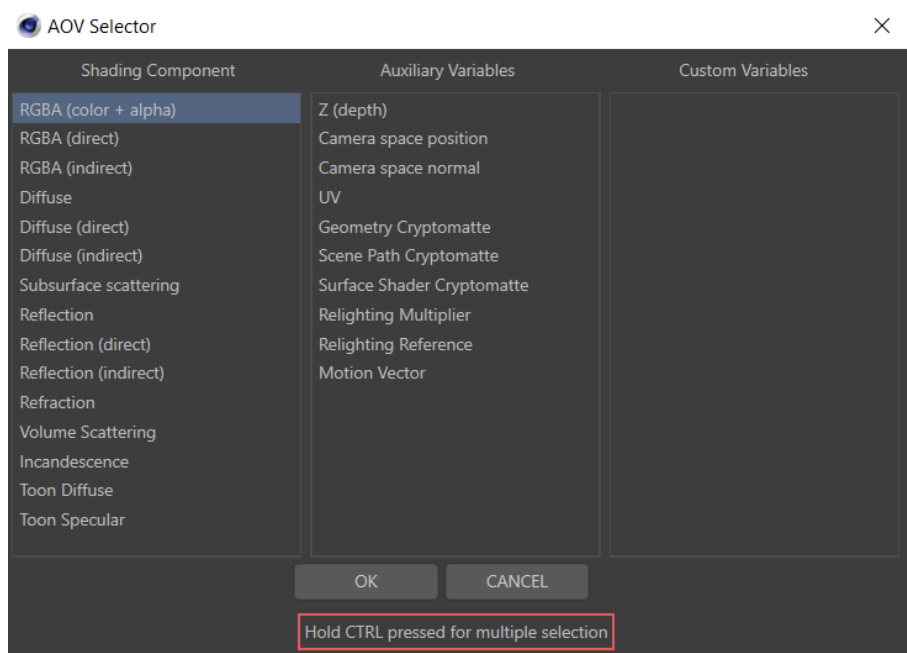


AOV Selector



The AOV Selector.

To add (or remove) AOVs from your list of layers, select (or unselect) the AOVs you want from the following tables:

- Shading Components
- Auxiliary Variables

After you have made your selections and press OK, only those that are selected will appear in the list of layers.

Shading Components

This group of AOVs can be used to deconstruct the *Beauty* image into several components that can be easily re-assembled using a simple additive combination. They each output to a RGB layer.

Each added layer that uses a shading component AOV will be rendered for every lighting element selected in the Multi-Light list.

Here is a description the AOVs that are part of the Shading Components:

AOV	Description
<i>RGBA (color + alpha) / primary</i>	This variable outputs the color and transparency information of the complete image. This is also called the <i>Beauty</i> .
<i>Diffuse</i>	Diffuse contribution.
<i>Subsurface Scattering</i>	Subsurface scattering contribution.
<i>Reflections</i>	Reflections contribution.
<i>Refractions</i>	Refractions contribution.
<i>Incandescence</i>	Incandescence contribution.

Compose the layers produced by these AOVs to re-create the *primary* image in the following manner (a simple additive process):

RGBA (color + alpha) / primary

=
Diffuse
+
Subsurface Scattering
+
Reflections
+
Refractions
+
Incandescence

Auxiliary Variables

Most auxiliary AOVs have values that are relevant regardless of the shaders assigned to the scene elements. For this reason, they are unaffected by the Mutli-Light selection.

The following are the auxiliary variables that are sometimes found to be useful:

Auxiliary AOV	Description
<i>Z (depth)</i>	For a given pixel, this variable outputs the distance between the camera and the object visible at this pixel. It produces a single channel layer. Bit depth should be set to 32 bits. This AOV is defined for all objects regardless of the shader.
<i>Camera Space Position</i>	This variable fills the image with the object position in camera space. Each image pixels is a position (x, y, z) in that space, in floating point values. The origin of the camera space is place at the camera (the center of the image). The camera looks towards the negative Z axis, so all pixels will show a negative value for the Z coordinate.
<i>Camera Space Normals</i>	This variable outputs a color-coded representation of camera space normals.
<i>UV</i>	The built-in UV values of the primitives are output to the first two channels of the rendered image, as float values. Each pixel will contain (u, v, 0.0).
<i>Geometry ID</i>	Every rendered object has an "object ID", which is an arbitrary value defined by the renderer. This variable will produce the object ID as a float value. The image produced contains a single value per pixel.
<i>Scene Path ID</i>	Each rendered scene path has an ID, which is an arbitrary value defined by the renderer. An important distinction between <i>Scene Path ID</i> and <i>Geometry ID</i> is that since the object scene path is used to define the value, each object instance will have a distinct scene path ID, while all instances will share the same geometry ID. This variable produces the scene path ID as a single float value per pixel.
<i>Surface Shader ID</i>	Each surface material has an arbitrary ID, defined by the renderer. All objects using a given material will have the same surface shader ID. This variable produces the surface shader ID as a single float value per pixel.