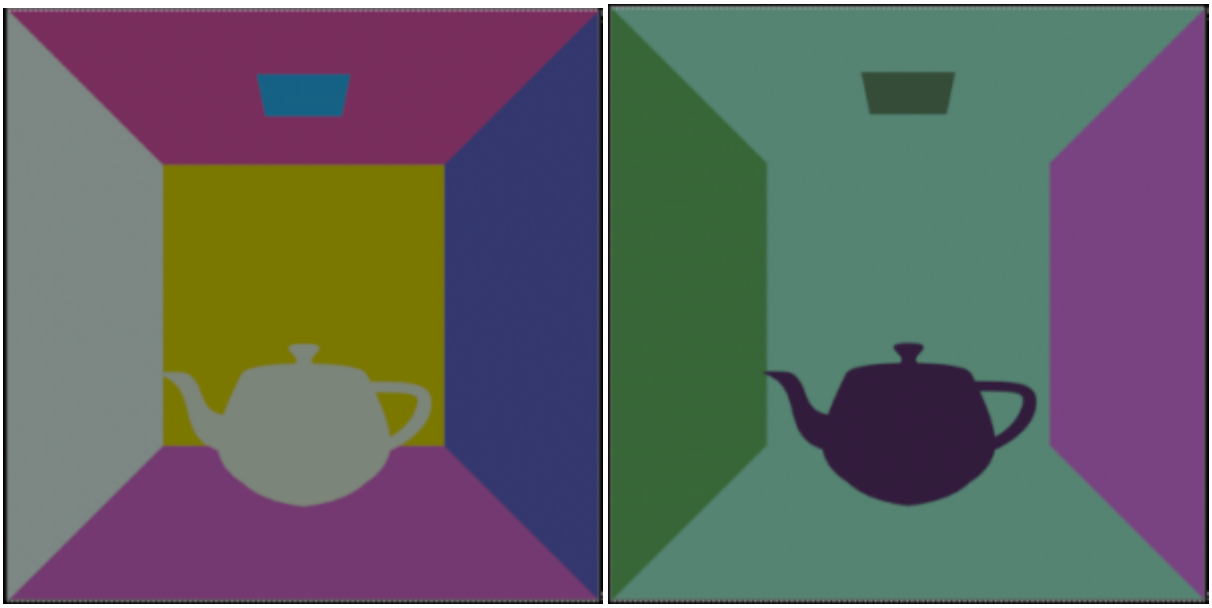
PxrCryptomatte

Cryptomatte was developed by Psyop Studios for use in production pipelines. The current version is Cryptomatte 1.2.0

The output of the sample filter allows artists to (mostly) automatically generate IDs for use in a compositing package such as Nuke for selecting and masking operations to specific objects or material selections. You will find examples in the Katana and Maya pages on Cryptomatte output. Below are object and material results of Cryptomatte output on a simple scene with a teapot.

- Cryptomatte limitations
  - Crop renders write the whole image with black padding. These will still be correctly aligned with the main render.
  - Volumes will render as opaque in the current release.
  - Checkpointing is not currently supported.



Parameters

Filename

The file name of the EXR image to write to, the default is “cryptomatte.exr”. For now this is (required) written to a separate file from your beauty or denoise EXR output.

Manifest

Legal values “None”, “Header” (default), and “Sidecar”. The Cryptomatte format allows for an optional manifest in JSON format mapping names to ids. This can either be included as a string metadata field in the EXR header or stored in a sidecar file. It may also be omitted.

Layer

What property to use to identify and group objects. May be “Name” (default) to use the objects' direct names, "Path" to use the full outliner paths to the objects, "Material" to group objects by shader network, or "Attribute" to use a custom user string attribute. The choices here also determine the names of the metadata strings stored in the EXR header and the names of the channel in the EXR image itself.

Attribute

If "Attribute" was chosen for the Layer, this will allow you name a custom string user attribute to use.

Levels

How many id/coverage pairs to include, default 6

Accuracy

Extra id/coverage pairs computed but not stored in the file, the default is 4. If the number of objects contributing to a pixel does not exceed the sum of levels and accuracy then the computed matting for that pixel will be exact. Otherwise, the matting will be approximate and increasing this may produce a better estimate (up to a point) for less significant objects in a pixel, but it does so at the expense of memory.