OpenEXR/Id
Isolate any object with a perfect antialiasing

Cyril Corvazier*
Mercenaries Engineering

Benjamin Legros
Mercenaries Engineering

Rachid Chikh
ON Animation Studios

Figure 1: *Any object can be isolated with a perfect antialiasing from a single OpenEXR/Id file: everything, the plane alone, the mask of the plane, the buildings. All images ©2015 LPPTV - LITTLE PRINCESS - ON ENTERTAINMENT - ORANGE STUDIO - M6 FILMS - LUCKY RED*

Abstract

We present a new storage scheme for computer graphic images based on OpenEXR 2.

Using such EXR/Id files, the compositing artist can isolate an object selection (by picking them or using a regular expression to match their names) and color corrects them with no edge artefact, which was not possible to achieve without rendering the object selection on its own layer.

Using this file format avoids going back and forth between the rendering and the compositing departments because no mask image or layering are needed anymore.

The technique is demonstrated in an open source software suite, including a library to read and write the EXR/Id files and an OpenFX plug-in which generates the images in any compositing software.

Keywords: compositing, alpha, matte, layering, openexr

Previous Work

[Friedman and Jones 2015] presents a method to render any kind of mask image using multiple AOVs called cryptomatte. This method is able to store any kind of alpha mask, but no RGB values. Cryptomatte masks are lossy compressed and it is not possible to render them with a lossless compression or to control this compression. Cryptomatte requires to use the Nuke multi channel feature. It may not be easy to integrate it in other compositing tools.

*e-mail:corvazier@mercs-eng.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4371-8/16/07
DOI: http://dx.doi.org/10.1145/2945078.2945136

Complete Coverage Rendering

During the rendering of the image, the renderer accumulates the coverage and the color information of every visible objects in the pixel. Each object is identified by a 32 bit integer. This information is then written in an EXR file, using half deep slices to store the coverage and the color. An extra 32 bit integer deep slice stores the object id.

The objects names table is compressed and stored in an EXR attribute.

Mask By Name

In the compositing tool, a dedicated node reads the EXR/Id file. The user provides a list of regular expressions matching the objects name to isolate. The re2 library is used to quickly match the objects name and a final image is then generated by accumulating the color and the coverage of the selected objects.

The compositing setup is based on the objects names, so it may be used in different shots as long as the objects names are consistent.

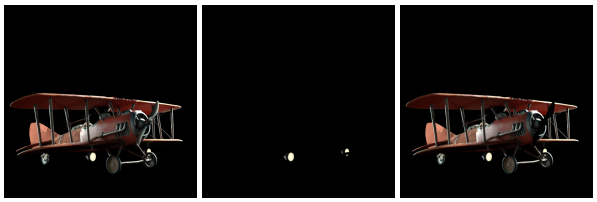

Figure 2: *The plane objects isolated using the* `plane`, `plane.*light` *and* `plane;-plane.*propeller` *expressions.*

Features

The images may include depth of field, motion blur, and transparency effects. For an extra storage cost, it is also possible to include the color and coverage informations of the reflected rays

in order to isolate the selected objects through the reflective and refractive surfaces, or to generalize, any user defined light path.

Performances

During the ray tracing of the image, the coverage processing time remains unoticeable.

The EXR/Id file has an extra storage cost. This cost is compensated by the fact that the EXR/Id files replace the traditionnal mask images and layers which are not needed any more. The images featuring a complex coverage information (lot of different objects per pixel like Figure 5) take a much higher space than a regular EXR.

Image	Regular EXR	Alpha EXR/Id	RGBA EXR/Id
Plane (2K)	7.2 MB	3.5 MB	13.9 MB
Forest (1K)	1.8 MB	3.5 MB	10.5 MB

Table 1: *The different EXR file size.*

To control the file size, we propose two lossy compression methods. We first limit the number of different object per pixels. Passed this limit, the RGBA contributions of the other objects are accumulated in the pixel in a "global" id.

We also propose to prune the RGBA contributions with an alpha lower than a user defined threshold. Those contributions are also accumulated in the "global" id.

Using those lossy compression methods, the whole image stays perfect, but the user selected images are lossy compressed.

Compositing Workflow

A single EXR/Id file is rendered with the coverage and the AOVs color informations. Using the EXR/Id node, the compositing artist isolates an object selection and color corrects it alone. Then, he replaces the object selection in the main image flow using a minus and a plus operation. The object selection is perfectly color corrected into the original image with no edge artifact.

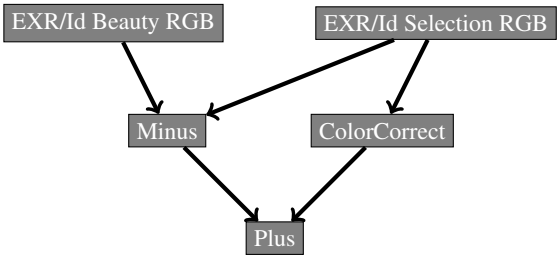

Figure 3: *Compositing workflow. The left EXR/Id node generates the whole image, the right EXR/Id node generates the object selection.*

Results

Using EXR/Id files, the rendering department doesn't have to worry anymore with layering, matte and mask rendering. The compositing department is now able to color correct any object selection of the image with no edge artefact, all that in any compositing software.

EXR/Id images are rendered by Guerilla Render. EXR/Id can be freely integrated in any other renderers.

Deep Compositing Discussion

One natural extension would be to generate deep EXR/Id, where each deep pixel fragment would represent a single object.

Download

OpenExr/Id can be downloaded at http://github.com/MercenariesEngineering/openexrid.

Figure 4: *The plane scene, from The Little Prince movie*

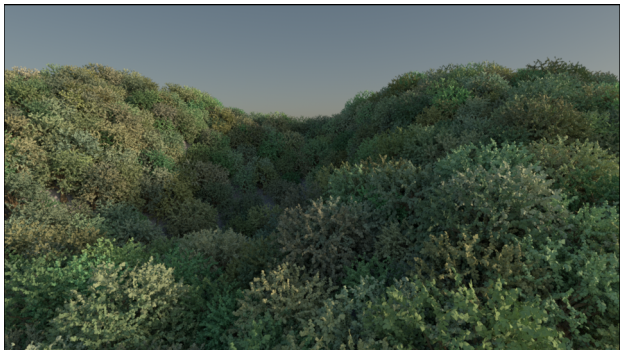

Figure 5: *The forest image. This image shows lot of different objects per pixel on average.*

References

CATMULL, E. 1978. A hidden-surface algorithm with anti-aliasing. *SIGGRAPH Comput. Graph. 12*, 3 (Aug.), 6–11.

FRIEDMAN, J., AND JONES, A. C. 2015. Fully automatic id mattes with support for motion blur and transparency. In *ACM SIGGRAPH 2015 Posters*, ACM, New York, NY, USA, SIGGRAPH '15, 47:1–47:1.

HILLMAN, P. 2013. The theory of openexr deep samples. Tech. rep., Weta Digital Ltd.

KAINZ, F., BOGART, R., AND HESS, D. 2004. The openexr image file format. In *GPU Gems*, R. Fernando, Ed. Addison-Wesley, 425–444.

PORTER, T., AND DUFF, T. 1984. Compositing digital images. *SIGGRAPH Comput. Graph. 18*, 3 (Jan.), 253–259.

SMITH, A. R. 1995. Image compositing fundamentals. Tech. rep.