Fully Automatic ID mattes with support for Motion Blur and Transparency

Jonah Friedman, Andrew C. Jones*
Psyop

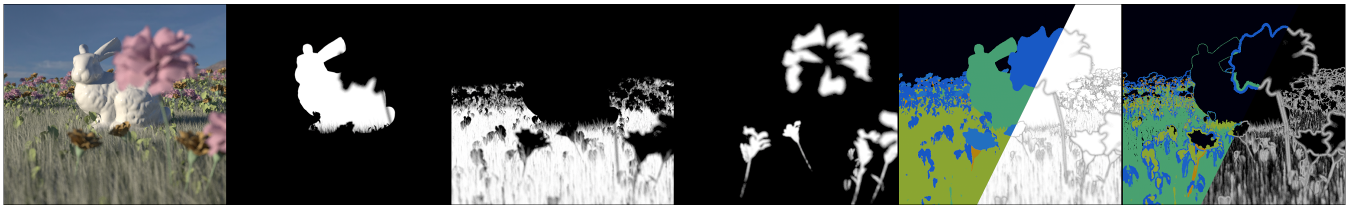

Figure 1: *"Beauty" render with depth of field and partially transparent grass (a). ID mattes taken from: bunny (b), multiple partially transparent grass objects selected by material (c) and arbitrarily selected foreground objects (d). ID-coverage pairs with colorized IDs and coverage channels for first (e) and second (f) depth layers.*

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Bitmap and framebuffer operations;

Keywords: rendering, ID matte

1 Introduction and Motivations

In 3D production for commercials, television, and film, ID mattes are commonly used to modify rendered images without re-rendering. ID mattes are bitmap images used to isolate specific objects, or multiple objects, such as all of the buttons on a shirt. Many 3D pipelines are built to provide compositors with ID mattes in addition to beauty renders to allow flexibility.

A complication in providing ID mattes is that multiple objects can occupy the same pixels. Reasons for this include anti-aliasing, filtering, motion blur, depth of field, and transparency. As a result, the most common way of providing ID mattes is in the form of RGBA images. In these images, an ID matte exclusively occupies a single channel of a series of RGBA images. The number of RGBA images needed to give compositors flexibility quickly accumulates. It can also be time-consuming for 3D artists to set up ID mattes, especially if they have to guess which mattes will be required.

Another approach is using an ID-coverage pair. The ID channel encodes the ID of one object per pixel. The coverage channel encodes how much of the pixel's value is contributed by this object. This method is unable to handle cases where multiple objects per pixel are important. It is possible to guess the ID of the second object and use inverted coverage, but this covers two objects at most. ID generation is also important, a per-object value from the renderer may not stay the same from shot to shot or even from frame to frame. Per-object IDs may also be too granular to be useful, and a number of objects grouped into one ID would be more useful.

*e-mail:jfriedman@psyop.tv, ajones@psyop.tv

SIGGRAPH 2015 Posters, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3632-1/15/08.
http://dx.doi.org/10.1145/2787626.2787629

2 Our Approach and Implementation

Our approach is closely related to the ID-coverage pair. We present two innovations: automatic generation of useful IDs from names, and ranked ID-coverage pairs to arbitrary depth.

Our system generates useful groupings of objects using already-available data: the names of objects and materials. Object names are broken into names and namespace. These names and namespaces stay stable from frame to frame and from shot to shot. Object names have the same granularity as using an object ID (cf. figure 1d), but remain the same from shot to shot and frame to frame. Using namespaces for an ID allows objects shared by an asset to be grouped together. Material assignments are typically already created by artists to group objects with similar properties (cf. figure 1c). All of these names have also already been chosen by artists to meaningfully describe their contents.

We hash the names as samples are taken and use the hash as our IDs. We repeat this process three times in our implementation, creating IDs out of names, namespaces, and material names. The names are also added to a manifest which will later allow reverse name lookups. These create a much better user experience for compositors, as their selection lists can be presented to them as a list of names.

At render time our system computes hashes from names and then records these hashes in arbitrary output variables. The samples are weighted using an ordinary filtering kernel, such as Gaussian or Blackman-Harris. If the sample is partially transparent, its positional weight is divided among its depth samples. The IDs are then ranked by their accumulated weight. The highest ranked ID is considered the most important ID in the pixel and its weight will be its coverage value. Together this is the first ID-coverage pair (cf. figure 1e). The second highest ranked ID and its coverage value make up the second ID-coverage pair (cf. figure 1f), and so on to arbitrary depth. The layers are stored in a multi-channel OpenEXR file. We use six levels of depth as our default value, meaning each pixel will contain an accurate matte for its six most contributing IDs. In rare cases where more is needed, an artist can simply increase the depth value.

We have found that in real-world production, our system works very well as a practical replacement for the virtually all ID mattes. The user experience for both creating and using mattes has been vastly improved. In further work, we plan to explore transferring shading adjustments done in compositing back to 3D assets by taking advantage of the shared names.