Fast Ambient Occlusion Rendering on Fur

August, 2008

Nhat Phong Tran

student of the 3rd term,
Babelsberg Film School

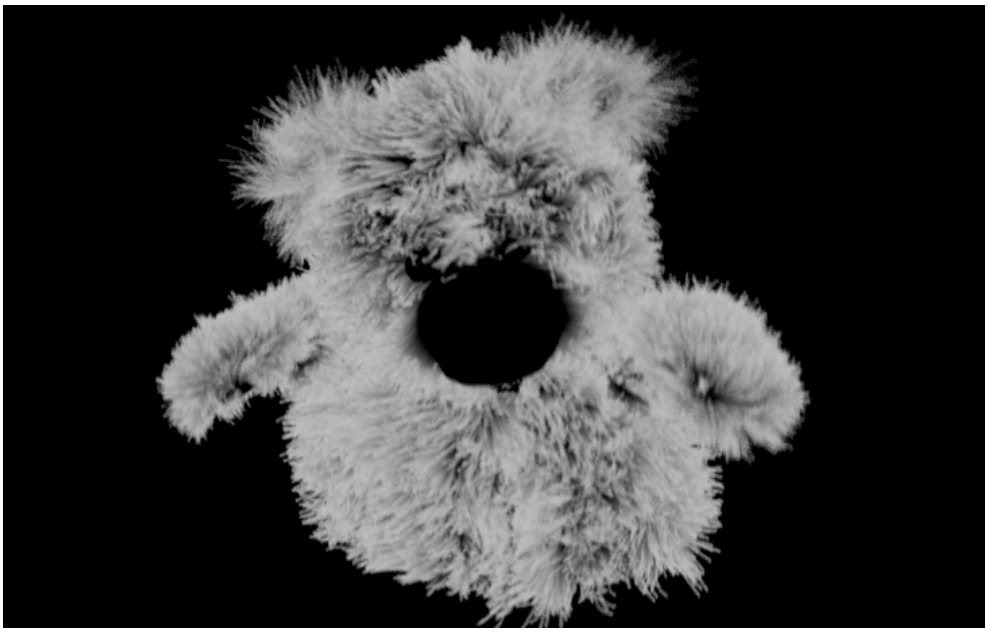

Figure 1:
The ambient occlusion contribution of a
cg teddy rendered in HD with our approach.
(without ambient occlusion 37 mins.,with ambient occlusion 37 mins.)

I. PREFACE

Rendering accurate ambient occlusion in fur is a time and memory expensive task. Though the results are very accurate and take all objects into account, the visual quality of the resulting image is often no compensation for the rendertime required. In the following chapter a way is described to effectively fake the look of ambient occlusion in fur and thus suggest the existence of an ambient lightsource. The goal was to achieve a believable look of ambient occlusion which can be rendered with motion blur, handle a very large amount of fur geometry, does not have any artifacts and render without producing much overhead and thus be memory efficient and saves much time.

II. CRITERIAS OF BELIEVABLE AMBIENT OCCLUSION IN FUR

If we analyze photographs or convincing rendered computer-graphics images with fur, we can easily extract visual components and criterias which imply ambient occlusion. In general the hair of the fur is darker at the root and brighter at the tip. This criteria applies to most kinds of fur which have in common that they are of an uniform length and do not show up any bald spots. If the fur to be rendered is of this kind, we can simply describe a shader which exactly reproduces this look postulating that the above mentioned prerequisites apply. So the pseudo-code for our shader would look like this:

```
surface hair()
{
```

```
        color hairColor = root color mixed with tip color;

        color diffuse = diffuse illumination loop;
        color specular = specular illumination loop;

        float ambientOcclusion = the length of each hair;

        result_c =  (color * diffuse + specular) *
                    ambientOcclusion;
}
```

Listing 1:
Pseudo-code shader of our basic approach

This approach would be well suited for most fur styles. But if the fur considerably differs in length and if it even has bald spots we need to additionally implement a varying float which marks edges between short and long fur and scales down the occlusion effect (changes are shown in bold letters):

```
surface hair()
{
        varying float edges; //assuming 1=edge and 0=noEdge
        float multiplier = 1;

        color hairColor = root color mixed with tip color;

        color diffuse = diffuse illumination loop;
        color specular = specular illumination loop;

        float ambientOcclusion = the length of each hair;

        ambientOcclusion = normalize(ambientOcclusion);
                           //to have values between 0 and 1

        ambientOcclusion =  (1-((1-ambientOcclusion) *
                                (1-edges * multiplier));
                                // a screen merge op to preserve
                                // occlusion data but excluding
                                // it in edge areas

        result_c =    (hairColor * diffuse + specular) *
                      ambientOcclusion;
}
```

Listing 2:
Shader in pseudo-code with an
ambient occlusion multiplier on edges of fur.

With this approach we can easily fake ambient occlusion in fur. Since it does not require any additional pre-renderpasses or does not comprise ray-tracing algorithms, it is very memory efficient and fast to render.

III. LIMITATIONS

Since the basic approach assumes the fur is of uniform length and of even distribution, our enhanced approach might be suitable for special cases with fur having bald spots and non-uniform lengths. But if the enhanced approach is used, the proceduralism of this shader will almost be gone because it requires an artist to manually mark the ‘edge zones’. There will also be a limitation if it is required to have other objects casting ambient occlusion on the fur or if the fur is technically required to occlude itself (i. e. armpits). These cases would exceed the limits of the approaches.