

# Refactorized Path Tracer for Thin Film Interference - DAE File Guide

Nico Deshler, Daniel Geng

Spring 2020

## Overview

This document details how to use our program for rendering thin film interference on a sphere. It also describes the specifics of our implementation which admits a spatially-variant thin film Bidirectional Scattering Distribution Function (BSDF). This BSDF acts on user-defined thickness data tabulated over the surface of the sphere in an angular coordinate space  $(\theta, \phi)$ .

## 1 Porting Thin Film Data to Renderer

The objects, lighting, and surface attributes for a scene are stored in Collada files. To admit user-defined data we modified the collada parser to identify a thinfilm label within the .DAE files. These files have a structure reminiscent to .SVG files in that scene information is organized hierarchically in libraries. It is within these libraries that we include the data required for calculating the effects of thin film interference, such as spatially-variant thickness values and indices of refraction for the material layers. The following code blocks exemplify how to associate a certain BSDF (in this case our thin film BSDF) with a particular sphere in the scene. It is important to place the code blocks in the appropriate libraries of the .DAE file hierarchy. Otherwise the path tracer will segfault upon reading the input file.

First, we provide an outline of the libraries in the .DAE files. Then we provide the code blocks that need to be included in each library for the path tracer to instantiate an object with a ThinFilmBSDF. All of the user-data is included in Block-I. Thus, we add some notes on the required formatting and assumptions made for the ThinFilmBSDF class.

### .DAE Scene Library Structure

```
<?xml version="1.0" encoding="utf-8"?>
<COLLADA xmlns="http://www.collada.org/2005/11/COLLADASchema" version="1.4.1">
<asset>...
```

```

<library_lights>...
<library_cameras>...
<library_images/>
<library_effects>...
<library_materials>...
<library_geometries>...
<library_controllers/>
<library_visual_scenes>
<scene>...
</COLLADA>

```

### Block-I: <library\_effects>

*Stores the data used to instantiate a thin film BSDF object in the render path tracer.*

```

<effect id="thinfilm-effect">
  <profile_COMMON>
    <technique sid="common">
      <phong>
        <emission>
          <color sid="emission">0 0 0 1</color>
        </emission>
        <ambient>
          <color sid="ambient">0 0 0 1</color>
        </ambient>
        <diffuse>
          <color sid="diffuse">0.64 0.64 0.64 1</color>
        </diffuse>
        <specular>
          <color sid="specular">0.5 0.5 0.5 1</color>
        </specular>
        <shininess>
          <float sid="shininess">50</float>
        </shininess>
        <index_of_refraction>
          <float sid="index_of_refraction">1</float>
        </index_of_refraction>
      </phong>
    </technique>
  </profile_COMMON>
  <extra>
    <technique profile="CGL">
      <thinfilm>
        <thickness_map count = "16" >1000 1000 1000 1000
          1000 1000 1000 1000

```

```

1000 1000 1000 1000
1000 1000 1000 1000

</thickness_map>
<width> 4 </width>
<height> 4 </height>
<n0> 1.00 1.00 1.00 </n0>
<n1> 1.35 1.36 1.37 </n1>
<n2> 1.5 1.6 1.7 </n2>
</thinfilm>
</technique>
</extra>
</effect>

```

The user-data for thin film interference is included within the `<extra>` tag in Block-I. These parameters link directly to the definition of `ThinFilmBSDF` class generated for our project. A description of each field is provided below.

1. **thickness\_map** - A 1D list of floats separated by spaces representing the grid of the thickness values (in nanometers) over  $\theta\phi$ -space in **row-major form**.
2. **width** - number of longitudinal samples (i.e. number of  $\phi$  samples)
3. **height** - number of latitudinal samples (i.e. number of  $\theta$  samples)
4. **n0** - A 3D vector containing refractive indices of the material surrounding the sphere for each RGB wavelength.
5. **n1** - A 3D vector containing the refractive indices of the thin film layer on the sphere for each RGB wavelength.
6. **n2** - A 3D vector containing refractive indices of the material inside the sphere for each RGB wavelength.

## Block-II: `<library_materials>`

*Pairs the thin film material with the thin film effect.*

```

<material id="thinfilm" name="thinfilm">
  <instance_effect url="#thinfilm-effect"/>
</material>

```

## Block-III: `<library_geometries>`

```

<geometry id="Sphere1-data" name="Sphere1">
  <extra>
    <technique profile="CGL">
      <sphere>
        <radius>.4</radius>

```

```

        </sphere>
    </technique>
</extra>
</geometry>

```

#### Block-IV: <library\_visual\_scenes>

*Creates a sphere object in the scene. The sphere is defined with a thin film BSDF in this block. The 'transform' parameter specifies a matrix that positions the center of the sphere in the scene.*

```

<node id="Sphere1" name="Sphere1" type="NODE">
<matrix sid="transform">1 0 0 -0.4 0 1 0 0.3 0 0 1 0.5 0 0 0 1</matrix>
<instance_geometry url="#Sphere1-data">
    <bind_material>
        <technique_common>
            <instance_material symbol="thinfilm" target="#thinfilm"/>
        </technique_common>
    </bind_material>
</instance_geometry>
</node>

```

## 2 Thin Film Interference

Our Thin Film BSDF implementation closely follows the approach detailed in this guide —> (<https://www.gamedev.net/tutorials/programming/graphics/thin-film-interference-for-computer-graphics-r2962/>). Rather than associate explicit wavelengths with each ray we consider the thin film interference effect for each color channel separately, using wavelengths 614 nm, 549 nm, and 466 nm for the Red, Green, and Blue color channels respectively. The reflection and transmission coefficients for each of these wavelengths computed at a certain point on the sphere supplies weights for each color channel. These coefficients are computed using the equations derived at the end of the document. The conclusions of the derivations depart slightly from those in the guide as there were errors in the guide's equations - namely that the  $\cos\theta_1$  factor in the expression for the phase difference should in fact be on the denominator.

Having outlined how the interference effects are computed for a single ray, we emphasize the assumptions that our implementation makes about the input data for the Thin Film BSDF class. Recall that we have limited our implementation to only support thin film interference on the sphere so as to simulate the appearance of bubbles.

#### Assumptions:

Our implementation assumes the angular difference between samples along the

longitudinal and latitudinal dimensions of the sphere are constant. Specifically,

$$\Delta_\theta = \frac{\pi}{N_\theta}$$

$$\Delta_\phi = \frac{2\pi}{N_\phi}$$

$$N_\phi = 2N_\theta$$

Additionally,

$$\theta_i = i \cdot \Delta_\theta, i \in [0, N_\theta]$$

$$\phi_j = j \cdot \Delta_\phi, j \in [0, N_\phi]$$

### 3 Preliminary Results



