TheoryExercise.pdf

Houmard, Loïc
297181

Blanc, Jonas
287508

Abi Fadel, Zad
295734

March 2021

1 Introduction

We will show that the formula to compute the color of the pixel c_b with infinite reflections is given by:

$$c_b = \sum_{i=0}^{\infty}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i \tag{1}$$

We will first show by developing the formula for a few terms and then prove it more formally by induction. If we develop it, starting from c_b and using the two formulas provided in the handout, we get:

$$\begin{align} c_b &= (1-\alpha_0)c_0 + \alpha_0 c^1 \tag{2}\\ &= (1-\alpha_0)c_0 + \alpha_0((1-\alpha_1)c_1 + \alpha_1 c^2) \tag{3}\\ &= (1-\alpha_0)c_0 + \alpha_0(1-\alpha_1)c_1 + \alpha_0\alpha_1 c^2 \tag{4}\\ &= (1-\alpha_0)c_0 + \alpha_0(1-\alpha_1)c_1 + \alpha_0\alpha_1((1-\alpha_2)c_2 + \alpha_2 c^3) \tag{5}\\ &= (1-\alpha_0)c_0 + \alpha_0(1-\alpha_1)c_1 + \alpha_0\alpha_1(1-\alpha_2)c_2 + \alpha_0\alpha_1\alpha_2 c^3 \tag{6}\\ &= \sum_{i=0}^{2}((1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i) + \alpha_0\alpha_1\alpha_2 c^3 \tag{7} \end{align}$$

So we see can see that we would get the right formula if we would sum it to infinity instead of stopping the expansion after 3 terms.

2 Induction

We will now do a proof by induction. We would like to show that:

$$c_b = \sum_{i=0}^{\infty}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i \tag{8}$$

Given the following relations:

$$c_b = (1-\alpha_0)c_0 + \alpha_0 c^1 \tag{9}$$

$$c^i = (1-\alpha_i)c_i + \alpha_i c^{i+1} \quad \forall i \in \mathbb{N} \tag{10}$$

The **claim** is the following:

$$c_b = \sum_{i=0}^{n}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{n}\alpha_l)c^{n+1} \quad \forall n \in \mathbb{N} \tag{11}$$

Base case, n=0 (no intersection occurs):

$$c_b = (1-\alpha_0)c_0 + \alpha_0 c^1 = \sum_{i=0}^{0}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{0}\alpha_l)c^1 \tag{12}$$

Let $k \in \mathbb{N}$ be given and suppose (11) is true for n=k. Then:

$$c_b = \sum_{i=0}^{n}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{n}\alpha_l)c^{n+1} \tag{13}$$

$$= \sum_{i=0}^{n}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{n}\alpha_l)((1-\alpha_{n+1})c_{n+1} + \alpha_{n+1}c^{n+2}) \tag{14}$$

$$= \sum_{i=0}^{n}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (1-\alpha_{n+1})(\prod_{l=0}^{n}\alpha_l)c_{n+1} + (\prod_{l=0}^{n}\alpha_l)\alpha_{n+1}c^{n+2} \tag{15}$$

$$= \sum_{i=0}^{n+1}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{n+1}\alpha_l)c^{n+2} \tag{16}$$

Thus, (11) holds for $n = k$+1, and the proof of the induction step is complete.

Conclusion: By the principle of induction, (11) is true $\forall n \in \mathbb{N}$.

Finally, since we only consider a specific number of reflection N, the formula can be simplified:

$$c_b = \sum_{i=0}^{N}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i + (\prod_{l=0}^{N}\alpha_l)c^{N+1} \tag{17}$$

$$= \sum_{i=0}^{N}(1-\alpha_i)(\prod_{k=0}^{i-1}\alpha_k)c_i \tag{18}$$

Where we discarded c^{N+1}, since we only consider the first N reflections and all the coefficients c_i for $i > N$ are considered to be equal to 0.

If we now take an arbitrarily large N that goes to infinity, we get the formula that we first wanted to prove:

$$c_b = \lim_{N\to\infty} \sum_{i=0}^{N} (1-\alpha_i)(\prod_{k=0}^{i-1} \alpha_k) c_i \tag{19}$$

$$= \sum_{i=0}^{\infty} (1-\alpha_i)(\prod_{k=0}^{i-1} \alpha_k) c_i \tag{20}$$

3 Implementation

We've already implemented the lighting function that returns the color of the light at an intersection point given a light and a material, excluding the contribution of potential reflected rays and ambient light, which we will use to compute the c_i.
Additionally, we've derived an iterative solution with this formula instead of the recursive one, which can be more easily computed with OpenGL.

We can implement it with the following algorithm:

1) Initialize the compounded reflection weight Π to 1 and the pixel color to 0.
2) Iterate over the number of reflections.
    3) Set c_i = ambient light.
    4) Iterate over all lights.
        5) Add their diffuse/specular contributions to c_i, using our lightning function.
    6) Update the pixel color by adding $(1-\alpha_i) * \Pi * c_i$ to it
    7) Update the compounded reflection weight Π by multiplying it by α_i
8) return the pixel color