CCI-36 - Radiosidade

Gabriel Martinz, Marina Moreira

1 Objetivo

O objetivo do presente trabalho foi tentar implementar o algoritmo de radiância para iluminação global. Por dificuldades de implementação, o projeto foi reduzido para iluminação básica.

2 Cenário

Foi criado o cenário no Blender como mostrado no roteiro do laboratório e na figura 1. O cenário foi exportado para um arquivo COLLADA para poder ser manipulado no Python.

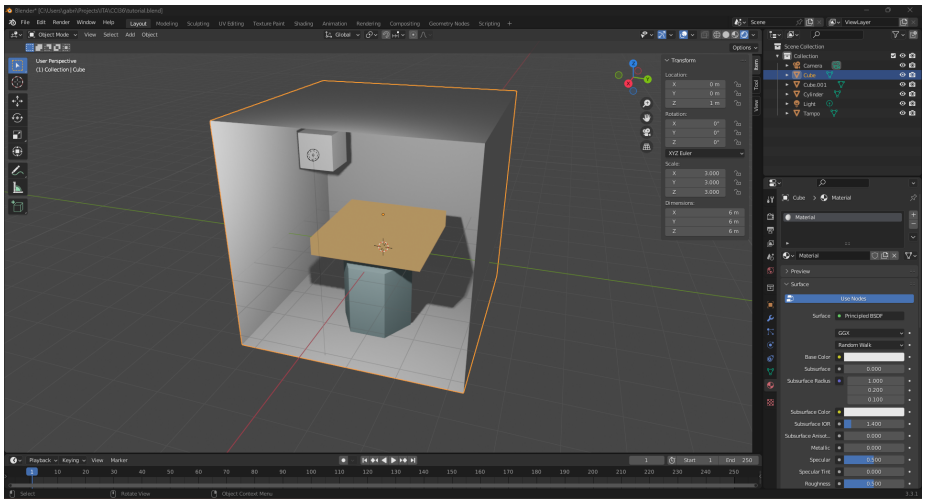

Figura 1: O cenário criado para o cálculo da iluminação.

3 Implementação da iluminação

O código foi implementado em um Jupyter Notebook. O arquivo COLLADA foi importado como XML e os dados necessários foram extraídos. Após isso,

foram criadas estruturas para facilitar a computação de dados necessários para o algoritmo, como:

- normais de face
- área da face
- centroides

Após isso, foi computada a visibilidade de cada triângulo para a fonte de luz. Para computar a visibilidade, foi feito um modelo simples checando a colisão do raio conectando a fonte de luz com o triângulo sendo analisado com todos os outros triângulos da cena.

Para encontrar a colisão, utilizamos a equação vetorial da reta e coordenadas baricêntricas do ponto de intersecção com o plano do triângulo. As equações utilizadas estão mostradas em 1 e 2.

$$R(t) = O + tD \tag{1}$$

$$T(u, v) = (1 - u - v)P_0 + uP_1 + vP_2 \tag{2}$$

Onde O é a coordenada da fonte de luz, D é a direção normalizada do raio e P_0, P_1, P_2 são os vértices do triângulo. Para a colisão, igualamos as duas equações e resolvemos para $\begin{bmatrix} t & u & v \end{bmatrix}^T$:

$$O + tD = (1 - u - v)P_0 + uP_1 + vP_2 \Rightarrow \begin{bmatrix} -D & P_1 - P_0 & P_2 - P_0 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = O - P_0 \tag{3}$$

Resolvendo o sistema, temos também que checar as condições das coordenadas baricêntricas (u, v), ou seja: $u \geq 0$, $v \geq 0$, $u + v \leq 1$. Além disso, é necessário checar que t é positivo e menor que a distância (norma) entre a fonte de luz e o centroide do triângulo analisado. A implementação foi feita com o NumPy para ser rápida.

Somente conseguimos computar a visibilidade até a finalização do presente trabalho.

4 Código

O código está neste repositório do GitHub.

5 Conclusão

Houve muito trabalho na parte de como melhor buscar e fazer as computações necessárias para obter os dados pedidos. Além disso, houve dúvida sobre o que exatamente precisaríamos entregar. Se possível, pedimos auxílio para poder melhorar o presente trabalho.