Programação 3D - Assignment I

Grupo 02
Francisco Campaniço 83463
João Rafael 83482
Rodrigo Oliveira 83558

Março 2019

Parser NFF

O programa começa por ler o ficheiro NFF definido pelo utilizador no início do código. Seguindo a especificação do formato NFF, adiciona uma câmara, luzes e objetos à cena. A cena (classe `Scene`) contêm uma câmara, um número de luzes definido pelo ficheiro NFF, e objetos também definidos pelo mesmo.

Um objeto (classe `SceneObject`) contém uma classe `Material` e métodos genéricos de interseção e cálculo de normais, que são *overriden* pelas classes que herdam desta.

Dado que o formato NFF não têm uma opção para *Bounding Boxes*, foi adicionada uma opção para tal: `aabb x0 x1 y1 y2 z1 z2`, que cria uma AABB com os limites em cada eixo declarados pelas variáveis seguintes.

Interseções

O raio verifica interseções com objetos ao chamar o método `intersect` de cada objeto, que devolve um *boolean* e a distância t da origem do raio ao objeto. Depois disto, o objeto com menor t é o mais próximo, logo os cálculos seguintes de iluminação aplicam-se a este.

A tabela abaixo contém os resultados de tempo para cada um dos testes disponibilizados. Estes resultados foram obtidos num portátil com um CPU Intel Core i7-8750H (6 *cores*, 2.2GHz) e uma GPU NVIDIA GTX 1060, correndo Kubuntu 18.04 e usando o programa Unix `time` para determinar o tempo de execução.

Teste	Low	Medium	High	Very High
Balls	0m11s	0m18s	12m14s	-
Mount	0m15s	-	8m58s	2h21m3s

Raio - Esfera

O cálculo da interseção raio - esfera contém várias otimizações. Primeiro, se a distância entre a origem do raio e o centro da esfera for maior que o raio da mesma, calcula-se

$$B = x_d(x_c + x_o) + y_d(y_c + y_o) + z_d(z_c + z_o).$$

Se B for negativo, o raio aponta para a direção oposta da esfera pode-se concluir que não há interseção.

Depois, calculamos

$$R = B^2 - C = B^2 - d_{OC}^2 + r^2$$

e se B for negativo, conclui-se também que não há interseção.

Finalmente, concluimos que há interseção, e calculamos

$$t_i = \begin{cases} B - \sqrt{R}, & \text{se } d_{OC}^2 > r^2 \\ B + \sqrt{R}, & \text{se } d_{OC}^2 \leq r^2 \end{cases}$$

para determinar o ponto de interseção e a sua normal, para efeitos de cálculo de cor.

Raio - Plano

A interseção raio - plano é simples; calcula-se o produto interno da normal do plano e a direção do raio. Se esta for zero, o raio e o plano são paralelos e como tal não se podem intersetar.

Depois, calculamos

$$t_i = -\frac{(o-a)\cdot n}{n \cdot d}.$$

Se t for negativo, rejeita-se os cálculos. Caso contrário, usa-se o t para determinar o ponto de interseção e a sua normal.

Raio - Triângulo

Para calcular a interseção entre raios e triânglos, utiliza-se o algoritmo de Möller-Trumbore. Primeiro, calculamos o determinante da matriz com os 3 pontos do polígono, que também pode ser calculada através de

$$det = a_{01} \cdot (d \times a_{02}),$$

a_{01} e a_{02} sendo arestas do triângulo. Se este determinante for zero, raio e triângulo são paralelos e abandonam-se os cálculos de interseção.

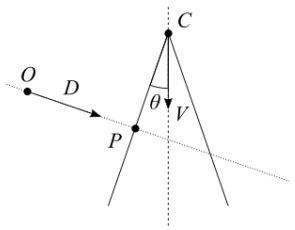
Depois, calcula-se coordenadas baricêntricas u e v. Se u não estiver no intervalo (0,1), v for negativo ou $u + v > 1$, abandonam-se os cálculos.

Finalmente, calculamos

$$t_i = \frac{1}{det}(a_{02} \cdot ((o - v_0) \times a_{01})).$$

Se t for negativo, a direção do raio é oposta ao vetor entre raio e triângulo, por isso abandonam-se os cálculos. Caso contrário, há uma interseção e usamos t para calcular o ponto de interseção e a normal correspondente.

Raio - Cone (Extra)



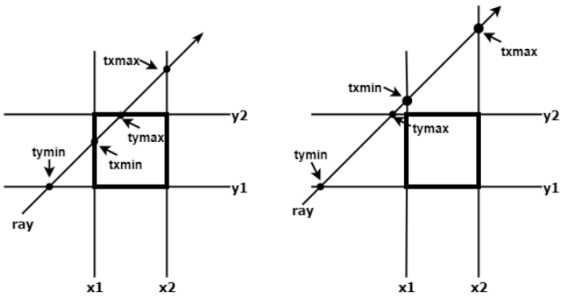
Começamos por descobrir o valor do seno e coseno de θ através de V, C e P. Depois calculamos os valores de a,b e c de acordo com a fórmula quadrática $at^2 + bt + c = 0$:

$$\begin{cases} a & = (\overrightarrow{D} \cdot \overrightarrow{V})^2 - cos^2\theta \\ b & = 2(\overrightarrow{D} \cdot \overrightarrow{V})(\overrightarrow{C}O \cdot \overrightarrow{V}) - \overrightarrow{D} \cdot \overrightarrow{C}Ocos^2\theta \\ c & = (\overrightarrow{C}O \cdot \overrightarrow{V})^2 - \overrightarrow{C}O \cdot \overrightarrow{C}Ocos^2\theta \end{cases}$$

De seguida calculamos o determinante $\Delta = b^2 - 4ac$ e discartamos os casos em que este é menor ou igual a zero. No caso de ser maior apenas rejeitamos o t que está no lado da sombra e vemos se este é menor que uma constante muito pequena. Caso seja maior devolvemos que o raio interseta o cone.

Raio - AABB (Extra)

O método utilizado para detectar colisões entre AABBs e raios foi o algoritmo explicado na aula *Slab Method*, proposto pela dupla *Kay e Kajiya*. Cada *slab* consiste num espaço entre dois planos paralelos, sendo assim, a AABB consiste em interseções entre diferentes *slabs*. O algoritmo calcula três grupos de interseção ao longo do raio. Se houver uma interseção para cada par de *slabs* pelo raio e se todos os intervalos sobreporem-se, então, o raio atingiu a AABB.



Na imagem acima podemos ver um exemplo do algoritmo proposto. A imagem à esquerda exemplifica um caso em que a interseção acontece, como é fácil verificar na imagem os intervalos sobrepõem-se: $txmin < tymax < txmax$. A imagem à direita mostra um exemplo em que não há sobreposição entre os intervalos, logo, não há interseção.

Iluminação Blinn-Phong

Para calcular a cor dos pixeis de acordo com as luzes da cena usamos o modelo de iluminação Blinn-Phong, isto é, começamos com a normal, o vetor que vai do ponto até à fonte de luz e calculamos a sua distância e, por consequência, a atenuação, visto que as luzes são todas *point lights*. Depois faz-se o produto interno entre a normal e o vetor que foi calculado previamente para se encontrar a intensidade e, por fim, as descrições do material para encontrar a componente difusa.

Falta só calcular a componente especular e para isso usamos o *half vector*, que é a soma do vetor de direção da luz com o *view vector* e com isso efetuamos o mesmo calculo usado para a componente difusa e ficamos com a componente especular.

Finalmente usamos as duas componentes calculadas e as descrições dos materiais para encontrar a cor do raio respetivo. Com múltiplas luzes apenas basta somar o resultado desta função com todas as luzes da cena e usando o fator de sombra que vai ser explicado a seguir.

Sombras

Para calcular as sombras começamos por descobrir o vetor de direção da luz em relação ao ponto em questão e depois criamos um raio com origem no ponto mais a direção da luz e com essa direção. Finalmente descobre-se a interseção deste raio com os objetos da cena (como fosse um raio normal) e, caso este intersete um objeto, a função devolve um valor lógico positivo.

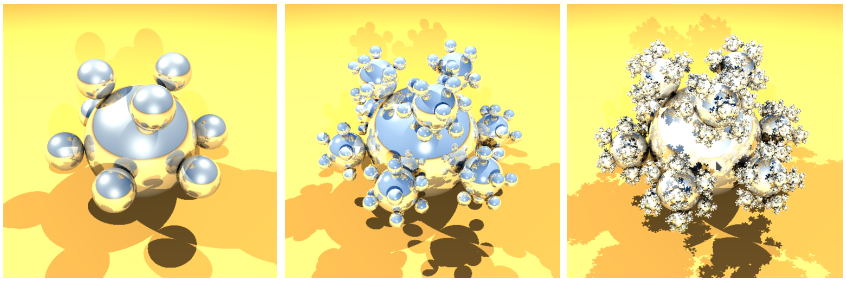
Este valor vai ser usado depois para determinar se o valor devolvido é 0 ou 1 (1 caso tenha intersetado algum objeto), para que depois se possa multiplicar o simetrico desse valor na componente de cor devolvida pelo calculo de Blinn-Phong.

Reflexão

Para calcular a reflexão começamos por ver se o material do objeto é refletor (componente especular maior que 0). Depois calculamos o raio refletor da seguinte forma:

$$p(t) = p_i + t\hat{r}_r.$$
onde,
$$\hat{r}_r = 2(V \cdot \hat{n})\hat{n} - V$$
$$V_n = (V \cdot \hat{n})\hat{n}$$
$$h = V_n - n$$
$$\hat{r}_r = V_n + h$$

Finalmente basta calcular a cor deste novo raio através do algoritmo de *ray tracing* e depois somá-la à cor já previamente obtida.



Refração

Todos os objetos que possuem transmitância passam ao cálculo de refração. A refração foi feita tendo em base a *Lei de Snell*. Com o uso de trigonometria e de forma a garantir que não há reflexão total verificamos se

$$C = 1 - \eta^2 \times (1 - (\overrightarrow{Dir} \cdot \overrightarrow{N})^2) \geq 0, \text{ onde } \eta = \frac{n1}{n2}(\textit{Lei de Snell})$$

Não havendo reflexão total, podemos calcular o raio refletor usando C previamente obtido através da seguinte fórmula:

$$\hat{r}_r = (\overrightarrow{Dir} \times \eta) + (\overrightarrow{N} \times (\eta \times (\overrightarrow{Dir} \cdot \overrightarrow{N}) - \sqrt{C}))$$

Agora basta calcular a cor deste novo raio através do algoritmo de ray tracing e depois somá-la à cor já previamente obtida (já tendo em conta a reflexão).