PROGRAMAÇÃO 3D

Relatório Técnico - Segunda Entrega

"Ray Tracing" [C++]

Grupo 6 – Madalena Assembleia | Rafael Martins | Stéphane Duarte

ÍNDICE

INTRODUÇÃO

Este relatório visa reportar todas as decisões e mecanismos inerentes à implementação da segunda fase do projeto da unidade curricular Programação 3D do Instituto Superior Técnico no ano letivo 2017/2018.

O projeto proposto foi a continuação da implementação do algoritmo de *ray tracing*. *Ray tracing* é um algoritmo de computação gráfica que permite renderizar imagens tridimensionais com um grau de fotorrealismo bastante elevado. A ideia por detrás deste algoritmo é simular os raios de luz existentes no mundo real. A origem dos raios é o observador. Cada raio é depois tratado individualmente, gerando novos raios refratados ou refletidos, consoante os materiais em que incide.

Nesta fase do projeto, foram implementadas estratégias que permitem melhorar o grau de fotorrealismo das imagens. O mecanismo de *soft shadows* permite suavizar a transição de espaços em sombra para espaços com luz. O mecanismo de *anti-aliasing* permite melhorar o cálculo de cor em cada pixel e também suavizar transições. O efeito de *depth of field* permite simular o efeito de foco/desfoco num plano, similar ao olho humano. Por fim, foi implementado *uniform grid* como método de aceleração para melhorar o tempo de renderização deste algoritmo.

Todo este projeto detalhado nos seguintes capítulos foi implementado em C++, com auxílio do *OpenGL,* do *Glew* e do *Freeglut*.

ALTERAÇÕES SIGNIFICATIVAS

Para a implementação das novas funcionalidades no algoritmo, foram necessárias algumas restruturações no código entregue na primeira fase.

Foi corrigida a classe *BoundingBox* que passou a ter apenas como atributos as coordenadas dos pontos mínimo e máximo. Quando se lê o ficheiro nff, todos os objetos ficam com uma bounding box associada. Foi eliminada a classe *Plane* uma vez que é impossível criar uma bounding box nestes objetos da cena.

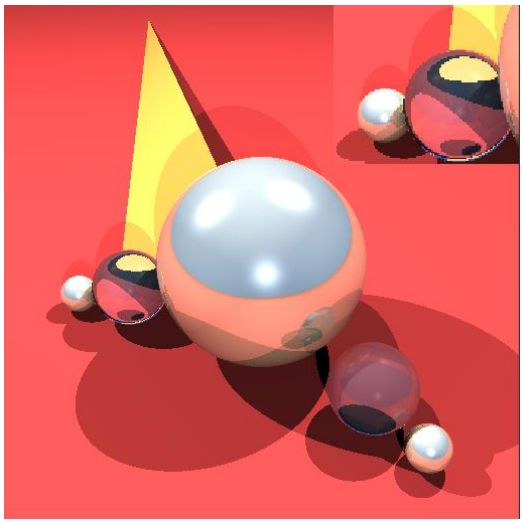
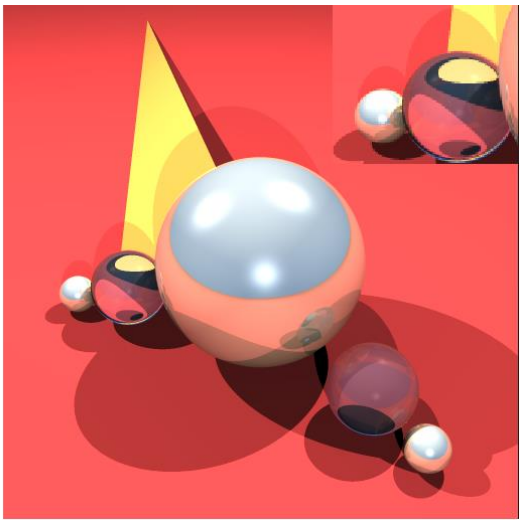
Foi criada a classe *Object* da qual as classes *Sphere* e *Triangle* herdam atributos e métodos. Esta alteração permitiu simplificar as funções de cálculo de interseções e reduzir a quantidade de código repetido.

Foi revisto o código com vista a melhorar a gestão de memória que, na primeira entrega, afetava muito negativamente os tempos de execução. Conseguimos melhorar os tempos para cerca de 10 a 20 vezes mais rápido, dependendo dos cenários.

O programa é agora interativo, permitindo ao utilizador mudar propriedades da renderização da cena através do toque de teclas conforme apresentado no ecrã.

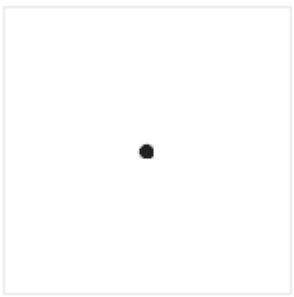
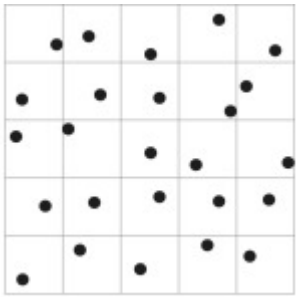
ANTI-ALIASING

Os métodos de *antialiasing* permitem reduzir o ruído na imagem e suavizar as transições através do aumento de cálculos da cor resultante num determinado pixel.

A imagem ilustra o resultado da cena sem *antialiasing* (à esquerda) versus o resultado da cena com *antialiasing* (à direita).

Este projeto recorre ao *método de Jittering*, que divide cada pixel num conjunto regular de áreas para as quais lança depois um raio primário numa posição aleatória.

Anteriormente, lançava-se um raio primário para o centro do pixel. Agora, o pixel é divido em áreas para as quais se escolhe uma posição aleatória para o cálculo de cor. No fim, a cor resultante do pixel resulta da divisão da soma de todas cores pelo número de áreas.

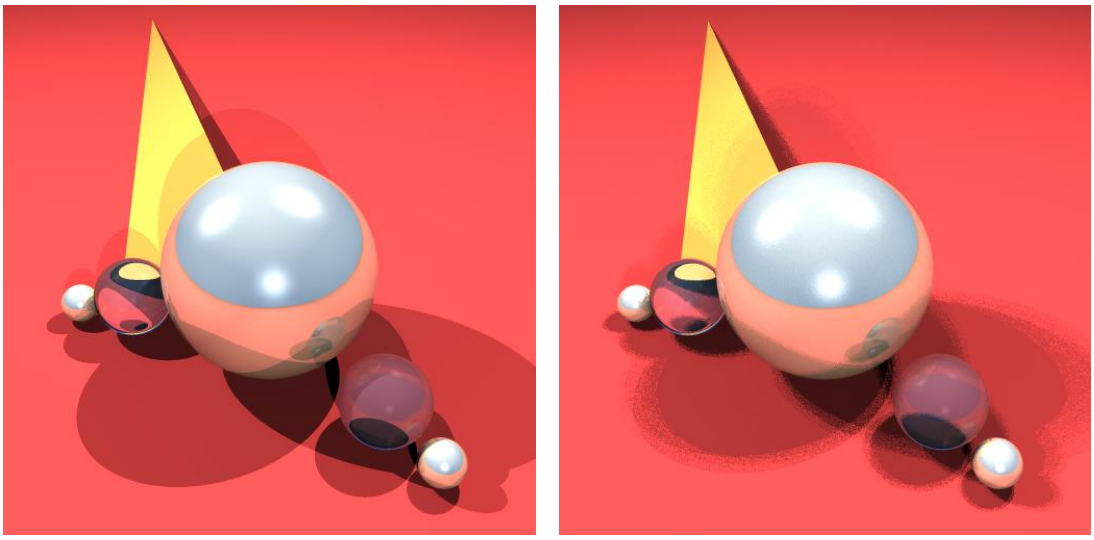
Em pseudo-código, tem-se:

```
for each pixel (i, j) do
        c = 0;
        for p = 0 to n-1 do
                for q = 0 to n-1 do
                        c += ray-color (i + (p + ξ)/n, j + (q + ξ)/n)
        c_ij = c / n^2
```

sendo c_{ij} a cor resultante no pixel de coordenadas (i, j), ξ um número aleatório entre 0 e 1 e n^2 o número de áreas por pixel.

SOFT SHADOWS

As *soft shadows* permitem fazer uma aproximação das sombras da cena às sombras do mundo real, suavizando a transição entre o local em sombra e o local em luz quando assim o deve ser feito.

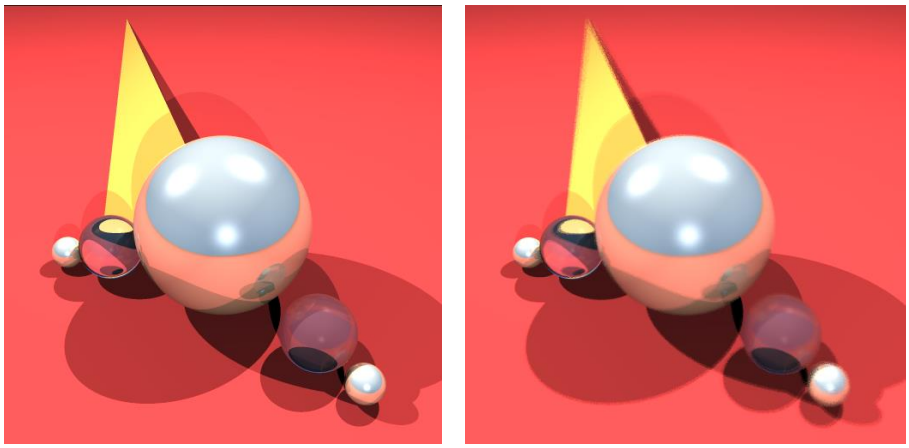


A imagem ilustra a cena com *hard shadows* (à esquerda) versus com *soft shadows* (à direita).

Ao invés de tratarmos a luz como um ponto, a luz passa a ser uma área, ou seja, um conjunto de pontos de luz. Quando é calculada a direção do raio desde o ponto inicial ao ponto de luz, é escolhido aleatoriamente um ponto na área de luz para o cálculo.

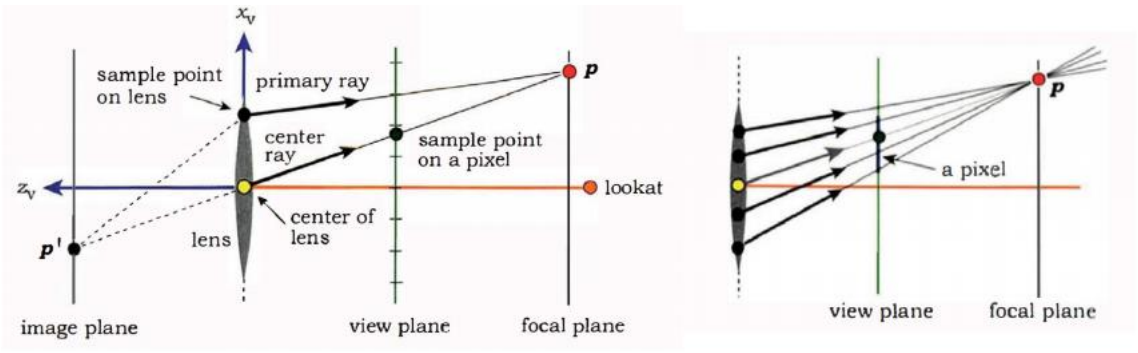
DEPTH OF FIELD

O efeito de *depth of field* permite simular na cena o foco em determinado plano, semelhante ao funcionamento do olho humano ou de uma câmara fotográfica.



A imagem à esquerda ilustra a cena sem plano de foco enquanto a da direita foca a cena nas quarta e quinta esferas.

Para a execução deste efeito começa-se por definir o número de samples a tratar e, para cada um, é gerado um raio para o cálculo de cor. Determina-se o ponto na lente, resultante de um ângulo aleatório e que serve de origem ao raio. Depois, e com o uso das coordenadas do ponto no *view plan*, calcula-se o ponto no plano de foco. Com as coordenadas do ponto no plano de foco e do ponto na lente determina-se a direção do raio que é, posteriormente, utilizado para o cálculo da cor.



A cor apresentada no pixel é média da soma das cores de cada sample.

ESTRUTURA DE ACELERAÇÃO (UNIFORM GRID)

Para acelerar o algoritmo de *ray tracing* recorreu-se a uma *uniform grid* que divide o espaço da cena de forma uniforme. Esta *grid* é gerada aquando da criação da cena. Esta estrutura acelera a renderização da cena pois o cálculo das interseções de raios com objetos passa a ser efetuado apenas com os objetos presentes nas celas em que o raio passa, ao invés de todos os objetos da cena, como até então era efetuado. Além disto, e como as celas são testadas ordeiramente, os cálculos param assim que o raio atinge o objeto mais próximo da sua origem.

A *grid* tem associada uma *bouding box* definida pelos pontos mínimo e máximo de todas as *bounding boxes* de todos os objetos da cena. Após a criação da *bounding box*, geram-se $N^{1/3}$ celas em cada direção, representando N o número de objetos na cena. Este tamanho pode ser, no entanto, alterado por um *mFactor* presente no código. Quanto maior o fator, maior será a *grid*. As celas estão definidas na classe *Cell*. Dado isto, percorrem-se todas as *bounding boxes* de todos os objetos da cena para associar cada um a uma ou mais celas nas quais se enquadram.

A travessia na *grid* é feita tal qual descrito em *"A Fast Voxel Traversal Algorithm for Ray Tracing"*. O raio percorre ordeiramente as celas, calculando as interseções com os seus objetos, quando presentes. Está previamente definido o "tamanho" de cada passo e em que direção o deve dar, tal como a condição de paragem.

CONCLUSÃO

O projeto tornou-se muito mais interessante e visualmente apelativo com estes efeitos que simulam fotorrealismo.

De todos os efeitos, o *antialiasing* é o consome mais tempo de execução. O efeito de *soft shadows* em nada altera o tempo de execução.

Podemos concluir, através da tabela em anexo, que a estrutura de aceleração aumenta em muito a performance do algoritmo. No ficheiro "*balls_medium.nff*" a renderização fica seis vezes mais rápida mas, quando testado o ficheiro "*balls_high.nff*", fica cerca de 300 vezes

mais rápida, passando de mais de uma hora a 12 segundos, quando os efeitos não são aplicados.

REFERÊNCIAS

AMANATIDES, John; WOO, Andrew, "A FAST VOXEL TRAVERSAL ALGORITHM FOR RAY TRACING", Toronto: University of Toronto.

PEREIRA, João Madeiras, "DISTRIBUTION RAY-TRACING", Disponível em: https://fenix.tecnico.ulisboa.pt/downloadFile/1689468335601274/5%20-%20Distribution%20Ray-Tracing.pdf, consultado a 04-04-2018.

PEREIRA, João Madeiras, "RAY-TRACING WITH UNIFORM SPATIAL DIVISION: PRACTICE", Disponível em: https://fenix.tecnico.ulisboa.pt/downloadFile/1689468335601275/Grid_Practice.pdf, consultado a 04-04-2018.

PEREIRA, João Madeiras, "GEOMETRY INTERSECTIONS", Disponível em: https://fenix.tecnico.ulisboa.pt/downloadFile/1407993358870437/Ray-Geometry%20intersections.pdf, consultado a 17-04-2018.

ANEXOS

TECLAS DE CONTROLO

Tecla a – Ativa/Desativa antialiasing

Tecla s – Ativa/Desativa soft shadows

Tecla d – Ativa/Desativa depth of field

Tecla g – Ativa/Desativa uso de estrutura de aceleração

TEMPOS DE EXECUÇÃO (Ficheiro: balls_medium.nff)

TEMPO DE EXECUÇÃO	GRID	SOFT SHADOWS	ANTIALIASING	DEPTH OF FIELD
24.251 s	NÃO	NÃO	NÃO	NÃO
67.004 s	NÃO	NÃO	NÃO	SIM
198.190 s	NÃO	NÃO	SIM	NÃO
596.887 s	NÃO	NÃO	SIM	SIM
26.352 s	NÃO	SIM	NÃO	NÃO
70.007 s	NÃO	SIM	NÃO	SIM
222.732 s	NÃO	SIM	SIM	NÃO
608.145 s	NÃO	SIM	SIM	SIM
4.507 s	SIM	NÃO	NÃO	NÃO
11.897 s	SIM	NÃO	NÃO	SIM
32.634 s	SIM	NÃO	SIM	NÃO
95.047 s	SIM	NÃO	SIM	SIM
4.543 s	SIM	SIM	NÃO	NÃO
12.335 s	SIM	SIM	NÃO	SIM
33.084 s	SIM	SIM	SIM	NÃO
98.400 s	SIM	SIM	SIM	SIM