**Universidade do Minho**
Escola de Engenharia
Departamento de Informática

**Mestrado em Engenharia Informática**

Nuno Cardoso

# Integração do Optix no Maya: Síntese Interativa de Imagens de Alta Qualidade

Dissertação de Mestrado

*Orientador:* Luís Paulo Peixoto dos Santos

**Braga, 22 de Outubro de 2015**

# AGRADECIMENTOS

Queria agradecer ao Professor Luís Paulo Santos pelo apoio, sugestões e orientação que tornaram esta dissertação possível. Agradeço também ao meus Pais e Amigos que me deram um enorme apoio durante todo o processo.

# RESUMO

Atualmente a síntese interativa de imagens de alta qualidade baseada em física é um dos principais desafios da computação gráfica, devido a ser um processo computacionalmente intensivo e complexo. Este processo tem vindo a tirar partido da capacidade que os GPU's possuem de efetuarem cálculos massivamente paralelos, levando a Nvidia a desenvolver uma API de Ray Tracing, denominada Optix, que permite efetuar cálculos de interseções de segmentos de reta com geometria tridimensional no GPU.

As aplicações de criações de conteúdos tridimensionais, como o Autodesk Maya, vêm acompanhadas de processos de síntese de imagem baseados em física e são normalmente usados para a produção de imagens de alta qualidade ou foto-realistas. No entanto, a maioria destes processos são executados no CPU, podendo a produção de uma única imagem exigir várias horas. Estas aplicações de síntese de imagem baseadas na física óptica estão atualmente a convergir para a utilização dos GPU's, o que permite uma maior rapidez na produção de imagens.

Pretende-se com esta dissertação desenvolver uma aplicação de síntese interativa de imagens de alta qualidade, integrá-la com uma aplicação de criação de conteúdos tridimensionais e avaliar as suas possibilidades. Esta aplicação tem como finalidade possibilitar a visualização de alta qualidade de conteúdos tridimensionais a taxas interativas e permitir ao mesmo tempo manipular as fontes de luz, materiais dos objetos, posicionamento dos objectos e posicionamento da camera.

Os resultados mostram que atualmente a síntese de imagem baseada em física, recorrendo ao GPU, permite a produção de imagens de alta qualidade a taxas interativas.

# ABSTRACT

Currently, high quality physically based rendering is one of the main challenges in computer graphics, due to its computational intensity and complexity. This process has been taking advantage of the massive parallel calculations allowed by the GPU's, leading Nvidia to develop a Ray Tracing API, called Optix, which performs intersection calculations of line segments with three-dimensional geometry on the GPU.

The three-dimensional content creation software, such as Autodesk Maya, provides physically based rendering processes typically used in the production of high quality or photo-realistic images. However, most of these processes are executed in CPU, and the production of a single image can take several hours. These rendering processes based on optical physics are currently converging to the use of GPU's, which allows greater speed in production of images.

The aim of this thesis is to develop an interactive renderer of high quality images, integrate it with a three-dimensional content creation software and evaluate the results. This process is intended to allow the display of high quality three-dimensional content at interactive frame rates, allow light manipulation, objects materials and camera transformations.

The results show that currently physically based rendering processes running on the GPU allow the production of high quality images at interactive frame rates.

# CONTEÚDO

Conteúdo iii

i INTRODUÇÃO 3

1 MOTIVAÇÃO 4

2 CARATERIZAÇÃO DO PROBLEMA 5

3 ESTADO DA ARTE 7

ii SÍNTESE INTERATIVA DE IMAGENS DE ALTA QUALIDADE 10

4 PROCESSO DE SÍNTESE DE IMAGEM 11

4.1 Path Tracing 12

4.2 Optix 12

5 INTEGRAÇÃO COM O MAYA 14

5.1 Concepção da Aplicação 15

5.2 Pipeline de Comunicação de Dados 16

6 AVALIAÇÃO DE RESULTADOS 18

7 CONCLUSÃO E TRABALHO FUTURO 32

7.1 Conclusão 32

7.2 Trabalho Futuro 33

# LISTA DE FIGURAS


Figura 1 Estrutura do grafo implementado para suporte a geometria e materiais. 13
Figura 2 Integração com o Maya - Viewport superior direito. 15
Figura 3 Integração com o Maya - Viewport central. 15
Figura 4 Arquitetura da aplicação. 16
Figura 5 Esquema do pipeline de comunicação de dados. 17
Figura 6 Maya Software - Cenário com 1 fonte de luz. 20
Figura 7 Mental Ray - Cenário com 1 fonte de luz. 21
Figura 4 Implementação Dissertação - Cenário com 1 fonte de luz. 25
Figura 5 Maya Software - Cenário com 7 fontes de luz. 26
Figura 6 Mental Ray - Cenário com 7 fontes de luz. 27
Figura 3 Implementação Dissertação - Cenário com 7 fontes de luz. 31

# LISTA DE TABELAS

Tabela 1 Comparação dos tempos de produção das imagens (tempos em milisegundos) 19

# Parte I

# INTRODUÇÃO

# 1

# MOTIVAÇÃO

A luz no mundo real interage com a matéria e superfícies que a rodeiam criando fenómenos visuais complexos como transferência de cor, reflexões ou cáusticas. A simulação destes fenómenos tem sido o objetivo último da computação gráfica desde os seus primórdios. Estando os princípios fisícos da interação luz-matéria bem definidos, a sua eficiente computação é ainda um desafio a resolver Ritschel and Dachsbacher (2012).

Na computação gráfica, a síntese de imagem é o processo de produção de uma imagem bidimensional a partir da interpretação de um conjunto de dados que descrevem um cenário tridimensional. Este processo é usado extensivamente no desenvolvimento de vídeo jogos, simulações, efeitos visuais e na indústria do cinema. A síntese de imagens foto-realistas de cenários complexos é um dos principais desafios da computação gráfica, devido a ser um processo computacionalmente intensivo, complexo e requerer um elevado tempo de produção quando a técnica de síntese utilizada simula questões de iluminação global Pharr and Humphreys (2010).

Existem diferentes algoritmos de síntese de imagem, variando desde simples e rápidos até complexos e precisos que simulam o comportamento da luz com maior exatidão. Uma das técnicas de síntese de imagem é a síntese baseada em física, cujo objetivo é simular a realidade usando princípios físicos para modelar a interação da luz com a matéria. Esta técnica faz uso de algoritmos de ray tracing para simular o trajeto que os raios de luz percorrem a partir de uma fonte de luz, num cenário tridimensional e as suas possíveis interações com as superfícies desse cenário. Tradicionalmente esta técnica é bastante custosa em termos computacionais sacrificando o desempenho em prol do realismo, podendo a produção de uma imagem levar tempos significativamente demorados. Atualmente, com a evolução dos GPU's e com a possibilidade de estes executarem programas personalizados, a síntese de imagem baseada em física tem vindo a fazer uso destas inovações permitindo que este processo de síntese possa ser efetuado a taxas interativas.

Para fazermos uso do processo de síntese de imagem é necessário possuir um conjunto de dados que descrevem um cenário tridimensional. Estes dados podem ser criados usando algoritmos ou software específico para a tarefa. O Autodesk Maya é um software que permite a criação de conteúdos digitais, oferecendo funcionalidades criativas de animação, modelação, simulação, síntese de imagem e composição. Este software é uma plataforma de produção altamente extensível, permitindo o uso da sua API em C++ ou Python para a criação de novas ferramentas ou plugins, sendo ideal para a avaliação de processos de síntese de imagem Gould (2003).

# 2

# CARATERIZAÇÃO DO PROBLEMA

O software de criação de conteúdos digitais como o Autodesk Maya é uma ferramenta essencial na produção de vídeo jogos, filmes de animação e efeitos visuais HP and Nvidia (2014). Este software é capaz de criar cenários tridimensionais extremamente complexos em termos de geometria, materiais, texturas e iluminação. No entanto, sempre que é necessário produzir uma imagem de alta qualidade e elevada resolução dos conteúdos criados, leva-nos a um processo de síntese de imagem bastante demoroso, quando utilizada a técnica de ray tracing. O tempo deste processo de síntese de imagem aumenta proporcionalmente com a complexidade dos conteúdos criados, com a resolução pretendida para a imagem final e com a complexidade dos fenómenos de transporte de luz simulados, podendo a produção de uma única imagem demorar tempos bastante elevados quando comparado com a síntese de imagem em tempo real baseada em rasterização.

O atual estado de desenvolvimento dos GPU's permite-nos efetuar cálculos massivamente paralelos levando a que as aplicações que tiram partido deste paralelismo sofram um aumento de desempenho. Usando os modernos GPU's e o seu paralelismo, a NVidia desenvolveu um motor de ray tracing, que efetua cálculos de interseções de segmentos de reta com geometria tridimensional, denominado Optix Parker et al. (2010). Este motor permite-nos desenvolver aplicações que fazem uso da técnica de ray tracing, tirando proveito da computação paralela dos GPU's.

Nos dias de hoje, as aplicações de síntese interativa de imagens de alta qualidade, que usam modelos baseados em física procuram migrar os cálculos que as tornam computacionalmente intensivas para os GPU's HP and Nvidia (2014). Estas aplicações colocam a ênfase na visualização de cenários tridimensionais estáticos, não possuindo animações e produzindo algum ruído inicial sempre que a câmara sobre este cenário altera a sua posição, ou colocam a ênfase na produção de imagens finais de alta qualidade deixando a gama de tempo de execução a taxas interativas. O motor de ray tracing Optix possibilita a implementação de aplicações de síntese interativa de imagens de alta qualidade usando modelos baseados em física. Este motor efetua os cálculos de ray tracing no GPU e permite-nos implementar os programas que são executados quando determinados eventos associados ao trajeto dos raios acontecem. O Optix quando integrado com um software como o Autodesk Maya, permitirá visualizar com elevada qualidade, cenários tridimensionais dinâmicos e o comportamento da sua iluminação em tempo útil.

Pretende-se assim com esta dissertação desenvolver uma aplicação de síntese interativa de imagens de alta qualidade para a visualização das interações das fontes de luz e materiais dos objetos num

cenário tridimensional. A aplicação vai utilizar o motor de ray tracing em tempo real Optix e será integrada com o Autodesk Maya no formato de um plugin. Os objetivos desta dissertação visam o desenho de um pipeline de comunicação de dados entre o Maya e o Optix, a implementação de um algoritmo de síntese de imagem baseado em física, tendo em conta a relação entre qualidade de imagem com velocidade de síntese, o desenvolvimento da aplicação, integração com o software Autodesk Maya e por último a avaliação da aplicação.

O principal desafio desta dissertação está focado na avaliação e análise da utilização da técnica de ray tracing, recorrendo ao GPU, para a produção de imagens de alta qualidade a taxas interativas, com o mínimo de artefactos a afetarem o resultado final, a sua implementação usando o Optix e na integração do Optix com outras aplicações. Neste tipo de processo de síntese de imagem usando o GPU é comum usar algoritmos progressivos, que apresentam a imagem inicial percetível mas com algum ruído ou aliasing, que vai reduzindo gradualmente. Este aspeto é aceitável quando queremos produzir uma única imagem, no entanto quando pretendemos visualizar uma sequência de imagens a taxas interativas, o ruído inicial afeta consideravelmente o resultado. Um desafio não menos importante é a integração com o software Autodesk Maya, que vai permitir a extração da informação dos vértices, normais, materiais, texturas e coordenadas de textura que constituem a geometria do cenário.

# 3

# ESTADO DA ARTE

Nesta secção é apresentado o estado da arte em algoritmos de síntese de imagem com foco na técnica de ray tracing, que vai contribuir para a posterior tarefa de seleção do melhor algoritmo para atingir o objetivo proposto por esta Dissertação.

O processo de síntese de imagem requer que seja simulada a distribuição de luz num cenário tridimensional de forma a poder preencher cada pixel da imagem final com uma determinada cor. Esta simulação é denominada de modelo de iluminação e divide-se em duas categorias: modelo de iluminação local e modelo de iluminação global. O modelo de iluminação local é o modelo mais simples e trata dos raios de luz que chegam a uma superfície diretamente de uma fonte de luz e sem obstruções. O modelo de iluminação global é mais complexo pois lida com os raios de luz que chegam a uma superfície de forma indireta, ou seja, que são emitidos a partir da inter-reflexão entre superfícies Comninos (2006).

Os algoritmos de síntese de imagem usados atualmente podem ser classificados como sendo baseados em uma das seguintes técnicas: rasterização ou ray tracing. A técnica de rasterização produz as suas imagens projetando geometricamente os elementos virtuais que constituem o cenário tridimensional num plano bidimensional. A técnica de ray tracing produz as suas imagens baseando-se na física ótica, simulando o trajeto que os raios de luz percorrem desde um ponto de vista, sobre um cenário tridimensional, até as suas interações com os elementos virtuais desse cenário. A rasterização é o processo de síntese usado atualmente pelo hardware gráfico. É um processo mais rápido que o ray tracing e utilizado frequentemente em aplicações de tempo-real. A técnica de ray tracing, por ser baseada na física ótica consegue lidar de forma precisa com modelos de iluminação global e por consequente é capaz de produzir resultados realistas e de elevada qualidade, simulando complexos fenómenos de transporte de luz.

A aplicação da técnica de ray tracing, utilizada de forma recursiva no processo de síntese de imagem foi introduzida pela primeira vez em 1980 por Whitted Whitted (1980). Este algoritmo descreve que o cálculo da visibilidade é efetuado traçando um raio a partir do ponto de vista do observador, passando pelo centro do pixel e verificando se interseta alguma superfície. Em caso de interseção com uma ou várias superfícies, este algoritmo escolhe o ponto de interseção mais próximo do observador e calcula a radiância para esse ponto traçando um raio em direção às fontes de luz, de forma a validar se este se encontra diretamente visível a partir da fonte de luz. A quantidade e direção de luz refletida

ou transmitida é de seguida avaliada traçando um novo raio a partir deste ponto. Este algoritmo vai atuar recursivamente até que uma condição de paragem seja invocada ou uma superfície difusa seja atingida. Neste algoritmo é possível simular reflexões difusas directas, reflexões especulares perfeitas, transmissão de luz e sombras. Esta introdução da técnica de ray tracing no processo de síntese de imagem serviu de base científica para os algoritmos que surgiram posteriormente, permitindo que a síntese de imagem se tornasse cada vez mais fiel à realidade, simulando com cada vez mais precisão e fisicamente plausível os fenómenos de transporte de luz.

Uma evolução do algoritmo de Whitted foi o algoritmo "Distributed ray tracing" Cook et al. (1984). Este algoritmo veio melhorar e resolver alguns problemas presentes no algoritmo clássico de Whitted. As melhorias foram principalmente nas reflexões e transmissões semi-coerentes, introdução de sombras com penumbras, profundidade de visão e motion blur. Neste algoritmo, ao contrário do clássico, eram disparados vários raios primários por pixel, dispersos estocasticamente e escolhidos usando uma função de probabilidade. Este algoritmo permitiu eliminar o aliasing resultante de métodos determinísticos como o de Whitted e substitui-o por ruído que é mais tolerável pelo sistema visual humano.

O algoritmo de "Path tracing" Kajiya (1986) proposto por Kajiya, contribuiu com um método alternativo, aos referidos anteriormente, de resolver a equação de síntese de imagem para a iluminação local e iluminação global. A equação de síntese de imagem, que tem como base a lei da conservação da energia, é um integral que calcula a radiância num ponto e é dado pela soma da radiância emitida mais a radiância refletida. Este algoritmo de Kajiya foca-se na integração da irradiância da iluminação direta de uma área de luz e da iluminação indireta proveniente de outras superfícies. Calculando no ponto de interseção de um raio primário com uma superfície, a intensidade de luz refletida para o observador. Neste algoritmo, ao contrário dos anteriores, não termina o traçado de raios ao atingir uma superfície difusa, sendo sempre escolhida uma nova direção para amostrar. Estas inovações levaram a que este algoritmo fosse o primeiro a conseguir produzir imagens indistinguíveis da realidade, conseguindo simular efeitos como sombras suaves, cáusticas, oclusão de ambiente, motion blur e profundidade de visão. No entanto, para obter imagens com elevada qualidade é necessário amostrar um grande número de raios, caso contrário a imagem final vai conter ruido indesejável. Uma extensão ao algoritmo de "Path tracing" foi o algoritmo "Bidirectional path tracing" Lafortune and Willems (1993) e Veach (1998). Este algoritmo traça os raios a partir do observador e a partir de uma fonte de luz escolhida com uma função de probabilidade, unindo posteriormente ambos os trajetos e pesando as suas contribuições para calcular a intensidade de luz refletida para o observador. Este algoritmo permite um cálculo preciso da irradiância proveniente de iluminação direta e iluminação indireta. Ambos os algoritmos, "Path tracing" e "Bidirectional path tracing", são classificados como algoritmos sem bias por não utilizarem estimativas da radiância que levam a erros sistemáticos.

Os algoritmos descritos anteriormente resolviam a equação de síntese de imagem usando o método de Monte Carlo. Este método é utilizado para obter uma aproximação numérica de integrais recursivos complexos. Jensen propôs um novo algoritmo como sendo uma alternativa as técnicas baseadas puramente no método de Monte Carlo. O algoritmo proposto foi o "Photon mapping" Jensen and

Christensen (1995) e funciona em dois passos. Num primeiro passo o algoritmo dispara fotões a partir das fontes de luz e guarda os seus trajetos até atingir uma superfície difusa. Após atingir uma superfície difusa é guardado o ponto de interseção e a sua intensidade. Esta distribuição de fotões é usada para estimar a radiância das superfícies do cenário. Num segundo passo é efetuada a síntese de imagem recorrendo ao uso de um algoritmo de ray tracing padrão e ao mapa de fotões gerado pelo primeiro passo, para obter a radiância no ponto de intersecção dos raios primários. Este algoritmo é capaz de simular efeitos como cáusticas realistas, transporte de luz em materiais translúcidos e inter-reflexões difusas. No entanto este é um algoritmo que introduz bias porque utiliza uma estimativa da radiância e isso conduz erros sistemáticos que levam a resultados que poderão não ser fisicamente corretos. Similar ao algoritmo de "Photon mapping", o algoritmo "Instant Radiosity" Keller (1997) dispara num primeiro passo um conjunto de fotões a partir das fontes de luz e guarda as suas intensidades. É no segundo passo que este difere do "Photon mapping", tratando os fotões guardados como fontes de luz virtuais (VPL's). O calculo da iluminação indireta neste algoritmo é efetuado verificando se um ponto da superfície esta diretamente visível pelas VPL's.

Um outro algoritmo é o "Metropolis light transport" Veach and Guibas (1997), que é baseado no método de Monte Carlo e na técnica de amostragem de "Metropolis". Este algoritmo calcula um trajeto entre o observador e a uma fonte de luz, usando o algoritmo "Bidirectional path tracing". Após este processo o algoritmo cria progressivamente trajetos alternativos a partir da mutação do trajeto inicial, sendo estes novos trajetos aceites ou rejeitados sob uma função de probabilidade. Sobre os trajetos criados é aplicado o método de "Metropolis" para calcular a distribuição da intensidade de luz da imagem. Este algoritmo explora a vantagem de, após ter descoberto um trajeto entre o observador e uma fonte de luz, os novos trajetos poderem partilhar da informação adjacente ao trajeto inicial.

Parte II

# SÍNTESE INTERATIVA DE IMAGENS DE ALTA QUALIDADE

# 4

# PROCESSO DE SÍNTESE DE IMAGEM

A primeira parte do desenvolvimento desta dissertação tem como objetivo implementar um processo de síntese de imagens de alta qualidade a taxas interativas.

A técnica de ray tracing é usada quando queremos produzir imagens de alta-qualidade ou foto-realista com base na simulação dos trajetos que a luz efetua no mundo real e as suas interações com os materiais Seymour (2013). Simular os mesmos fenómenos usando a técnica de rasterização é mais complexo devido à natureza da técnica e usualmente recorre-se a aproximações não plausíveis em termos físicos. Neste capitulo é apresentado algoritmo de síntese de imagem implementado e a API de suporte escolhida.

Seguindo o estado da arte apresentado anteriormente, existem variados algoritmos de ray tracing, cada uma com vantagens e desvantagens em relação aos fenómenos ópticos simulados. O path tracing é o algoritmo que permite simular todos os fénomenos ópticos expostos no estado da arte, não sendo primariamente optimizado para nenhum fénomeno em particular. No entanto para obter uma imagem de alta qualidade é necessário traçar um elevado número de raios o que aumenta o tempo de computação. A natureza deste algoritmo permite que seja implementado usando métodos progressivos, refinando o resultado a cada imagem produzida. Para implementar este algoritmo tirando partido do poder de processamento do GPU foi utilizado o Nvidia Opix.

Nesta implementação do processo de síntese de imagem foram escolhidos um conjunto de funcionalidades a implementar. Devido a ser baseado numa API que utiliza o GPU este processo vai ser iterativo e progressivo. Este processo de síntese vai simular apenas reflexões entre as superfícies, utilizando a BRDF de Phong Modificada Montes and Ureña (2012), o fenómeno de transmissão de luz não vai ser simulado nesta implementação. As fontes de luz utilizadas vão ser descritas por aéreas luz, devido a ser o tipo que melhor representa as fontes de luz no mundo real. Esta implementação vai ainda utilizar uma técnica de tone mapping, para podermos utilizar intensidades de luz elevadas e mapeamento de ambiente.

## 4.1 PATH TRACING

O algoritmo de Path Tarcing implementado é baseado na implementação fornecida com o Optix. Este algoritmo iterativo e progressivo constrói um caminho para cada pixel da imagem a produzir e acumula o resultado entre as varias iterações.

O caminho inicia a partir da camera e termina ao atinjir uma determinada profundidade. A condição de paragem para cada caminho traçado é dada pela técnica de roleta russa ou quando o caminho atinge uma determinada profundidade máxima. A cada intersecção com a superfície de um objeto é calculada a radiância nesse ponto, escolhendo aleatoriamente uma fonte de luz e calculada a sua contribuição utilizando uma amostragem uniforme sobre a sua aérea. A contribuição da fonte de luz é depois multiplicada pelo resultado da função de distribuição de refletância bidirecional (BRDF) e esta pelo produto do coseno entre a normal da superficie e a direção da fonte de luz. O resultado deste produto é dividido pela probabilidade de distribuição resultante da amostragen da fonte de luz. A BRDF utilizada nesta implementação é o modelo de Phong Montes and Ureña (2012), alterado de forma a introduzir propriedades físicas ao modelo original. Esta BRDF foi escolhida devido a ser simples e rapida de calcular. Após calcular a radiância é escolhida uma nova direção amostrando a brdf e é atenuada a radiância utilizando o valor da brdf para esta nova direção. Após terminar um caminho é aplicado um algoritmo de tone mapping à radiância final e acumulada com o resultado da iteração anterior.

## 4.2 OPTIX

O Optix é uma API genérica dedicada ao calculo de intersecções de segmentos de reta com geometria tridimensional Parker et al. (2010). Esta API utiliza o GPU e o modelo de programação CUDA para efetuar os cálculos. Este modelo permite tirar partido da computação massivamente paralela dos GPU's, levando a uma redução dos tempos de computação para processos complexos.

O modelo de programação do Optix assenta na criação de programas de controlo dos vários estágios de um ray tracing NVIDIA (2014). Estes estágios são a inicialização de raios, a sua intersecção com superfícies, o calculo da cor no ponto de intercepção utilizando o material da superfície e a geração de novos raios. A geração de novos raios pode ser efetuada recursivamente e é configurável de acordo com as variáveis que queremos armazenar.

O Optix permite ao utilizador definir o tipo de primitivas que compõe a geometria e de que forma são calculadas as intersecções, na implementação do Path Tracing são usados triângulos para definir a geometria. Para poder calcular as intersecções é necessário fornecer ao Optix um modelo que define a geometria tridimensional a usar. Este modelo toma a forma de um grafo hierárquico onde podemos definir a estrutura que mais se adequa a nossa implementação. O Optix fornece um conjunto de possíveis tipos de nós a usar e permite usar múltiplas configurações de forma a optimizarmos a

implementação. Na implementação do Path Tracing foi desenhada uma hierarquia optimizada para um cenário dinâmico, onde os objectos tridimensionais podem ser actualizados a qualquer momento.

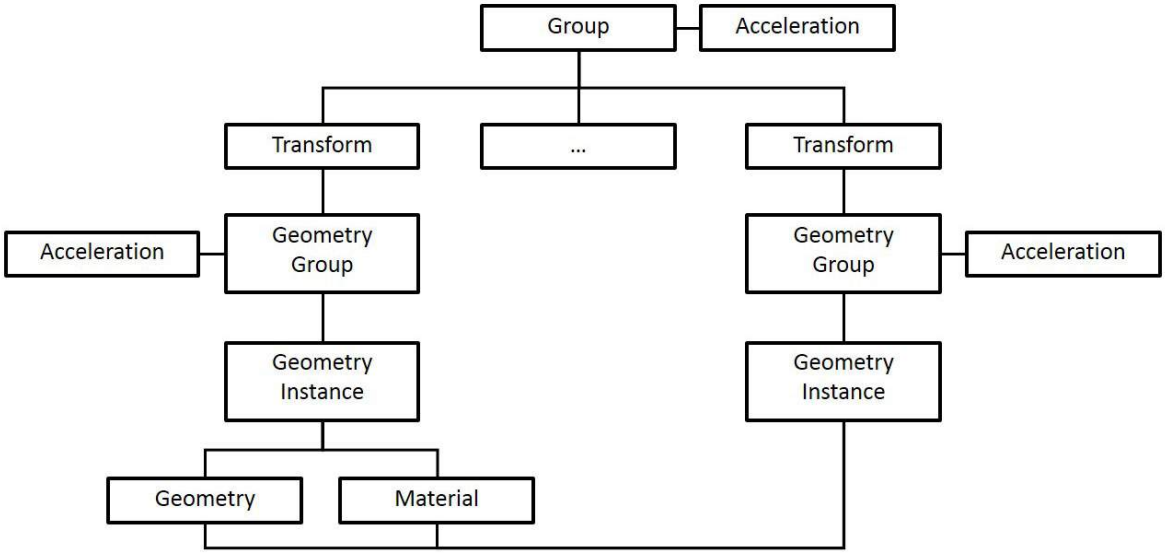



Figura 1: Estrutura do grafo implementado para suporte a geometria e materiais.

Na figura 1 podemos observar um esquema da estrutura desenhada. No topo da hierarquia esta presente um nó que agrupa todos os restantes elementos e associado a este uma estrutura de aceleração. Quando adicionamos uma nova geometria é criado um elemento Geometry que contém a informação que define a geometria e um elemento Material que contém a informação que define o seu material. Associado a estes elementos estão os programas de controlo que tratam da intersecção é do calculo da cor resultante. Num passo seguinte são criados os elementos Geometry Instance, Geometry Group, Transform e associada uma estrutura de aceleração ao elemento Geometry Group. Os elementos Geometry e Material, criados anteriormente, são ligados ao elemento Geometry Instance criando uma instância da geometria fornecida. O elemento Transform contém a informação da transformação desta instância no espaço tridimensional. Se adicionarmos uma segunda instância da mesma geometria com uma transformação tridimensional diferente, são criados os elementos Transform, Geometry Group e Geometry Instance, sendo este ligado ao elementos Geometry e Material já existentes. Esta estrutura permite que varias instâncias utilizem o mesmo conjunto de dados que definem uma geometria. Outra vantagem é a reconstrução das estruturas de aceleração quando a geometria é alterada ou transformada. Quando uma geometria é transformada no espaço tridimensional invocamos a reconstrução apenas da estrutura de aceleração do elemento do topo da hierarquia. Caso os vértices da geometria sejam transformados invocamos apenas a reconstrução da estrutura de aceleração do seu elemento Geometry Group associado. Desta forma minimizamos a reconstrução das estruturas de aceleração e por consequente optimizamos o desempenho.

# 5

# INTEGRAÇÃO COM O MAYA

A segunda parte do desenvolvimento desta dissertação tem como objetivo integrar o processo de síntese de imagem descrito no capitulo anterior com uma aplicação de criação de conteúdos tridimensionais. A aplicação escolhida para a integração foi o Autodesk Maya devido a ser uma aplicação essencial na criação de imagens foto-realistas geradas por computador, usada a nível da industria do cinema, video jogos, visualização e investigação HP and Nvidia (2014). Este software apresenta uma poderosa API que permite extender as suas funcionalidades e implementar novas funcionalidades para diferentes processos de computação gráfica Gould (2003).

Esta integração tem como objetivo disponibilizar uma visualização de alta qualidade a taxas iterativas de cenários tridimensionais desenvolvidos no Maya. Isto permite ao utilizador movimentar a câmara pelo cenário, alterar as fontes de luz e suas propriedades, ou ainda alterar as propriedades dos materiais e visualizar os resultados sem ter de esperar pelo resultado de um processo de síntese offline. No mundo real estas funcionalidades permitem uma rápida iteração do desenvolvimento de cenários tridimensionais, com a vantagem de visualizar em tempo útil os resultados das suas criações.

Para a integração do processo de síntese de imagem foi desenvolvida uma aplicação no formato de um plugin para o Autodesk Maya. O Maya permite implementar o plugin fazendo uso do seu viewport, ver figuras 2 e 3, desta forma podemos tirar partido das funcionalidades que o viewport disponibiliza, como mover a câmara e interagir com os objetos no cenário. Esta aplicação foi desenvolvida de forma modular, permitindo a sua futura utilização com diferentes processos de síntese de imagem. Com este objetivo em mente foi necessário desenvolver um pipeline de comunicação de dados abstracto da aplicação com qual vai integrar. Nesta integração tiramos partido das funcionalidades existentes no Maya para extrair informação sobre geometria, materiais, texturas, luzes e câmaras a usar. O Maya permite criar diferentes tipos de geometria, materiais e luzes. Para a aplicação de integração foi escolhido um subgrupo de elementos suportados, no entanto a sua arquitetura foi desenhada de forma a poder lidar com os diferentes tipos de elementos. O tipo de geometria suportado são malhas poligonais definidos por triângulos. O tipo de materiais suportados é do tipo Phong e o tipo de luzes são áreas. O modelo de Phong foi escolhido devido a ser um modelo computacionalmente eficaz de avaliar. O tipo de luzes baseado em áreas foi escolhido devido a ser o que melhor representa as fontes de luz no mundo real.

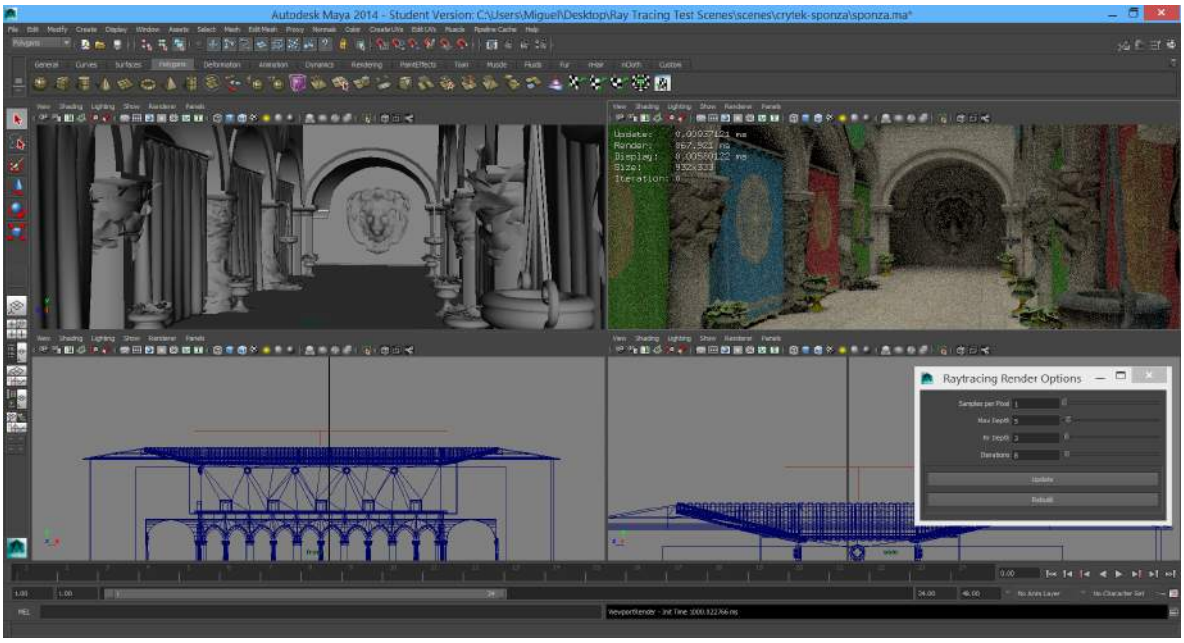



Figura 2: Integração com o Maya - Viewport superior direito.

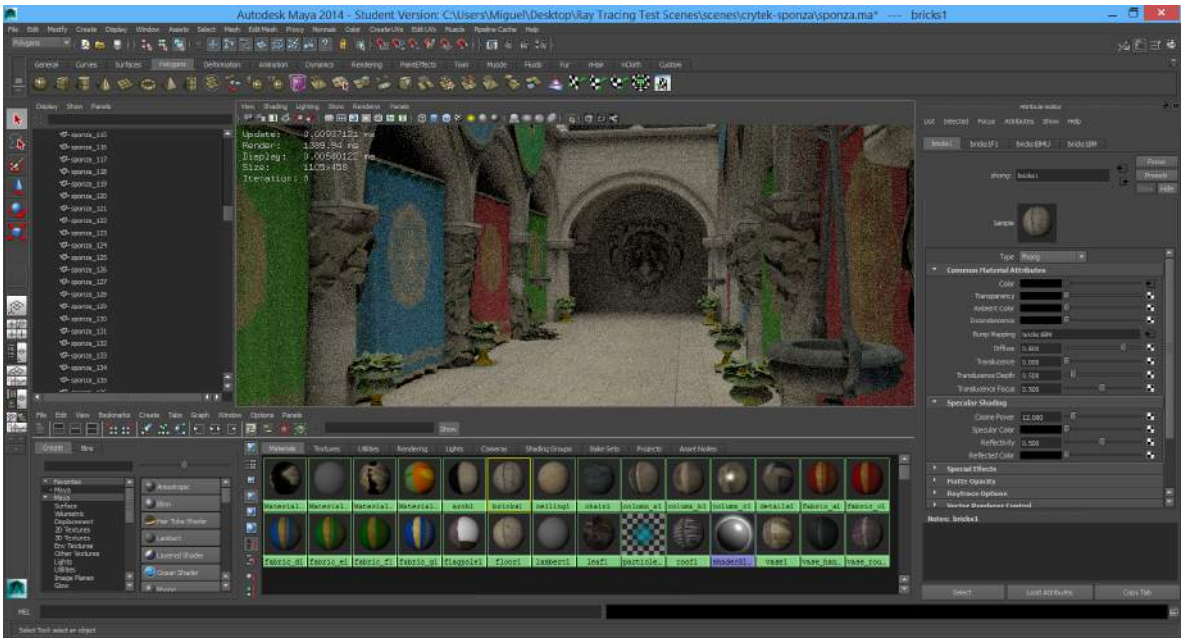



Figura 3: Integração com o Maya - Viewport central.

## 5.1 CONCEPÇÃO DA APLICAÇÃO

O principal objetivo para a concepção da aplicação é ser modular e permitir futuras integrações quer com outras aplicações de criação de conteúdos tridimensionais, quer com diferentes implementações de processos de síntese de imagem. A aplicação consiste em dois módulos principais, como podemos

ver na figura 4: plugin e o processo de síntese de imagem. De forma a estes dois módulos poderem comunicar foi desenvolvido um interface e um conjunto de estruturas de dados para suporte a comunicação da informação entre os dois módulos. O processo de síntese de imagem implementa este interface e o plugin é responsável por criar um objeto do tipo interface.

O módulo Plugin consiste em duas funcionalidades principais: interpretar os dados do Maya e exibir a imagem produzida pelo processo de síntese de imagem. Os dados interpretados pelo Plugin definem a geometria, materiais, texturas, fontes de luz e câmara. A imagem produzida pelo processo de síntese é depois exibida no viewport do Maya. O módulo de síntese de imagem lida com a transformação dos dados fornecidos pelo plugin num formato reconhecido pelo optix. Este módulo implementa toda a lógica de síntese de imagem utilizando a API do optix e fornece ao plugin uma textura com a imagem produzida.

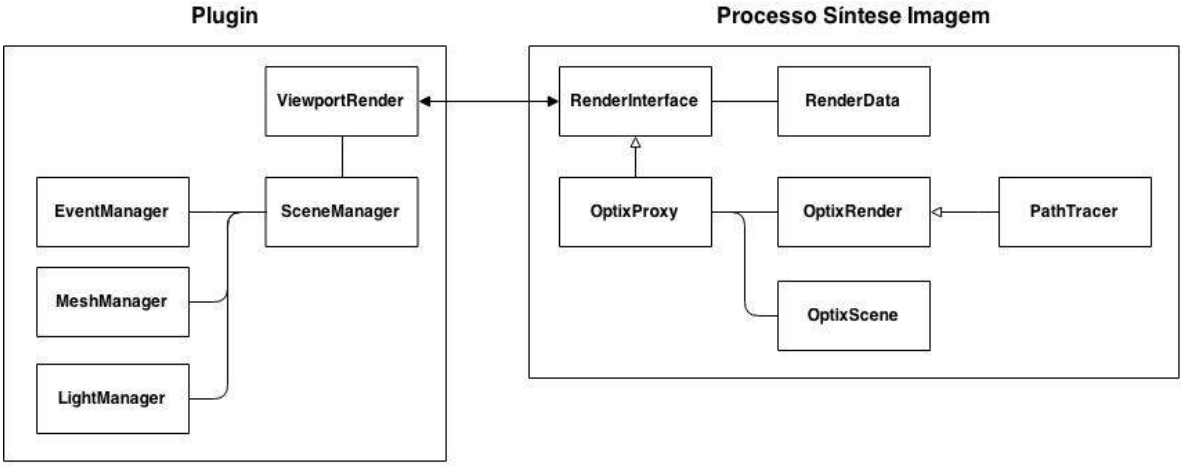



Figura 4: Arquitetura da aplicação.

## 5.2 PIPELINE DE COMUNICAÇÃO DE DADOS

O processo de síntese de imagem, no momento da criação, inicializa uma textura para onde vai ser gerada a imagem. Os dados necessários para a geração desta imagem passam do plugin para o processo de síntese através de um pipeline de comunicação de dados desenvolvido para esta aplicação, exemplificado na figura 5. Este pipeline deve ser abstrato das estruturas de dados utilizadas pelo Maya e pelo Optix, permitindo desta forma a integração com outras aplicações criativas ou outras API's dedicadas à síntese de imagem. Este pipeline funciona no sentido do plugin para o processo de síntese, passando os dados originados do Maya por um conjunto de processsos até serem transformados num formato interpretado pelo Optix. Desta forma a integração com outras aplicações ou processos apenas necessita de implementar um processo que traduza os dados.

Existem três processos principais neste pipeline, funcionando independentemente. Um processo dedica-se a receber os dados da câmara do viewport do Maya e transforma-os em matrizes inter-

pretáveis pelo Optix. Outro dedica-se à atualização das luzes sempre que um área de luz é adiciona ou alterada. Um terceiro dedica-se à atualização dos objetos tridimensionais, tratando da informação que define o objeto, sendo esta: vértices da malha, normais, coordenadas de textura, matrizes de transformação do objeto, propriedades dos materiais e texturas. Esta informação é tratada independentemente e depois utilizada para definir uma instância do objeto, que por sua vez é passada ao optix.

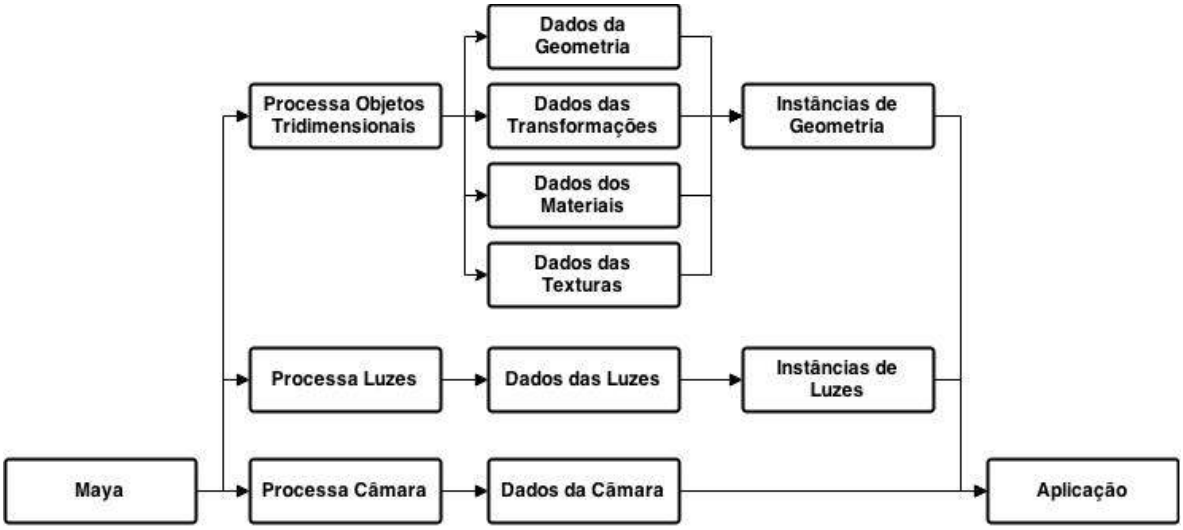



Figura 5: Esquema do pipeline de comunicação de dados.

# 6

# AVALIAÇÃO DE RESULTADOS

Neste capitulo apresentamos os resultados da aplicação desenvolvida e comparamos o processo de síntese implementado com dois processos incorporados no Autodesk Maya, o Maya Software e o Mental Ray.

A avaliação de resultados foi realizada utilizando o modelo tridimensional Sponza, constituindo por 262267 triângulos. Foram criadas 2 configurações para este cenário, uma com 1 fonte de luz e outra com 7 fontes de luz. Para cada configuração foi utilizado 2 pontos de vista sobre o cenário. Os processos de síntese do Maya e da Dissertação estão configurados para utilizar 1 raio sombra e uma profundidade máxima de 5 raios. As imagem foram produzidas a uma resolução de 1024 por 768. A versão do Autodesk Maya utilizada foi a 2014 e do Optix a 3.6.2. As especificações do PC utilizado são as seguintes: Windows 8.1 64 bits, Intel Core i5-4200U @ 1.60Ghz, 8 Gb Ram, Nvidia GeForce GT 735M.

A aplicação de síntese de imagem desenvolvida integra com o Maya no formato de plugin. Activando o plugin, este utiliza o viewport do Maya para exibir a imagem produzida e interagir com o cenário tridimensional. As propriedades da câmara, fontes de luz e geometria são enviadas para a aplicação juntamente com as propriedades dos materiais e texturas. A cada interação com o cenário tridimensional a aplicação de síntese reinicia o processo de síntese e vai refinando a imagem até um determinado numero de iterações seja atingido ou uma nova iteração com o cenário seja efectuada. As interações possiveis com o cenário tridimensional são o movimento de câmara, translação e rotação das fontes de luz, assim como a alteração das suas propriedades, translação e rotação da geometria e alteração das propriedades dos materiais aplicados a geometria.

Um particularidade da implementação da dissertação está na forma como lida com cenários que possuem múltiplas fontes de luz. Quando existe mais do que uma fonte de luz, a implementação da Dissertação ao determinar a iluminação direta para cada raio traçado, escolhe aleatoriamente uma fonte de luz, multiplicando a radiância final pelo numero total de fontes de luz. Este detalhe permite ao utilizador obter um feedback muito mais rápido do que as alternativas. Como podemos verificar na tabela de comparação de tempos 1 apresentada neste capitulo, o incremento de fontes de luz de 1 para 7, no mesmo cenário, com a mesma vista, toma tempos de produção aproximadamente equivalentes. Esta particularidade mostra-se bastante importante quando pretendemos ter um feedback rápido da distribuição da iluminação no cenário. Os resultados mostram que a nível de interatividade, esta

varia consoante o tamanho do viewport, podendo a produção de uma imagem levar desde alguns milisegundos até alguns segundos.

De seguida apresentamos a comparação de tempos de produção das imagens entre a implementação da Dissertação e os processos de síntese incorporados no Maya.

| | **Implementação Dissertação** | | | | **Maya Software** | **Mental Ray** |
|---|---|---|---|---|---|---|
| | **1 Iteração** | **8 Iterações** | **32 Iterações** | **100 Iterações** | **Total** | **Total** |
| **Vista 1 - 1 Fonte de Luz** | 1995 | 15976 | 63912 | 200169 | 58000 | 69000 |
| **Vista 1 - 7 Fontes de Luz** | 2054 | 16449 | 65930 | 205902 | 181000 | 171000 |
| **Vista 2 - 1 Fonte de Luz** | 2827 | 22634 | 90636 | 283538 | 62000 | 75000 |
| **Vista 2 - 7 Fontes de Luz** | 2806 | 22619 | 89969 | 280944 | 208000 | 240000 |

Tabela 1: Comparação dos tempos de produção das imagens (tempos em milisegundos)

Como podemos observar pela tabela 1 a implementação da dissertação produz resultados mais rápido do que alternativas e sendo um processo iterativo os tempos de produção aproximam-se das alternativas as 32 iterações, para o caso de 1 fonte de luz e as 100 iteração, para o caso de 7 fontes de luz. Os tempos apresentados mostram que, para uma resolução de 1024 por 768, perdemos interatividade ao chegar as 8 iterações.

Para concluir o capitulo, apresentamos as imagens produzidas por cada processo seguido de uma analise comparativa. As imagens produzidas pelo Maya Software 6 e Mental Ray 7 apresentam nitidez e bom detalhe, no entanto verificamos algum ruído devido a utilização de apenas 1 raio sombra. As imagens da implementação da Dissertação 4 apresentam bastante ruído na iteração 1, no entanto podemos observar que na iteração 100 o ruído já é bastante reduzido. Apesar do ruído presente na iteração 1, podemos observar que existe algum detalhe presente o que leva o utilizador a ter um feedback bastante rápido da iluminação do cenário. Na implementação da Dissertação por cada iteração é traçado 1 caminho por pixel, isto significa que na iteração 100 traçamos 100 caminhos por pixel. Num algoritmo de traçado de raios o ruído é reduzido aumentando o numero de caminhos por pixel, sendo necessário um elevado numero para eliminar o ruído por completo. Outro detalhe importante de observar é a qualidade da iluminação indireta presenta na implementação da Dissertação, sendo esta fisicamente plausível. Nas imagens produzidas pelo Maya Software, a iluminação indireta é aproximada por uma constante de ambiente. Nas imagens produzidas pelo Mental Ray, a iluminação indireta é calculada usando traçado de raios em conjunto com uma constante de ambiente.

(a) Maya Software - Vista 1

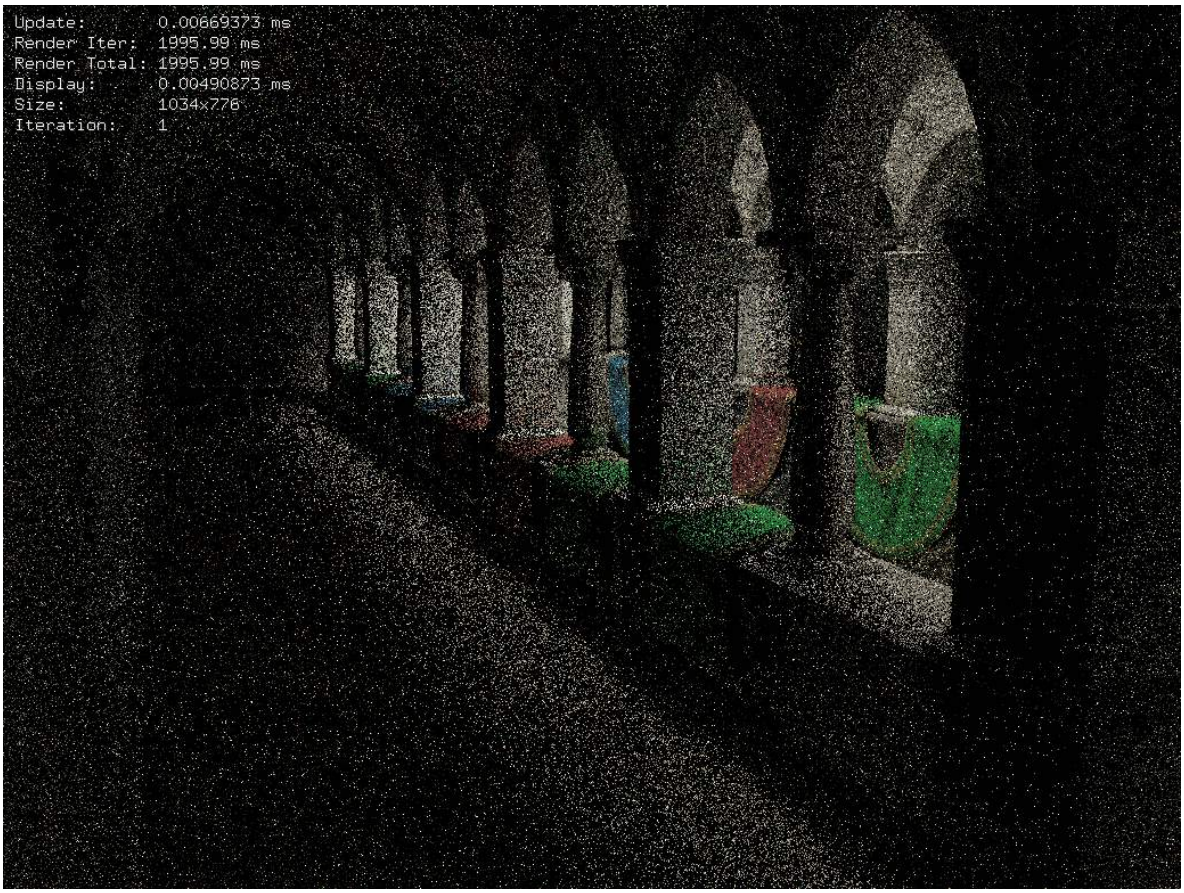
(b) Maya Software - Vista 2

Figura 6: Maya Software - Cenário com 1 fonte de luz.

(a) Mental Ray - Vista 1

(b) Mental Ray - Vista 2

Figura 7: Mental Ray - Cenário com 1 fonte de luz.




(a) Vista 1 - 1 iteração

(b) Vista 1 - 8 iterações

(c) Vista 1 - 32 iterações

(d) Vista 1 - 100 iterações

(e) Vista 2 - 1 iteração

(f) Vista 2 - 8 iterações

(g) Vista 2 - 32 iterações

(h) Vista 2 - 100 iterações

Figura 4: Implementação Dissertação - Cenário com 1 fonte de luz.

(a) Maya Software - Vista 1

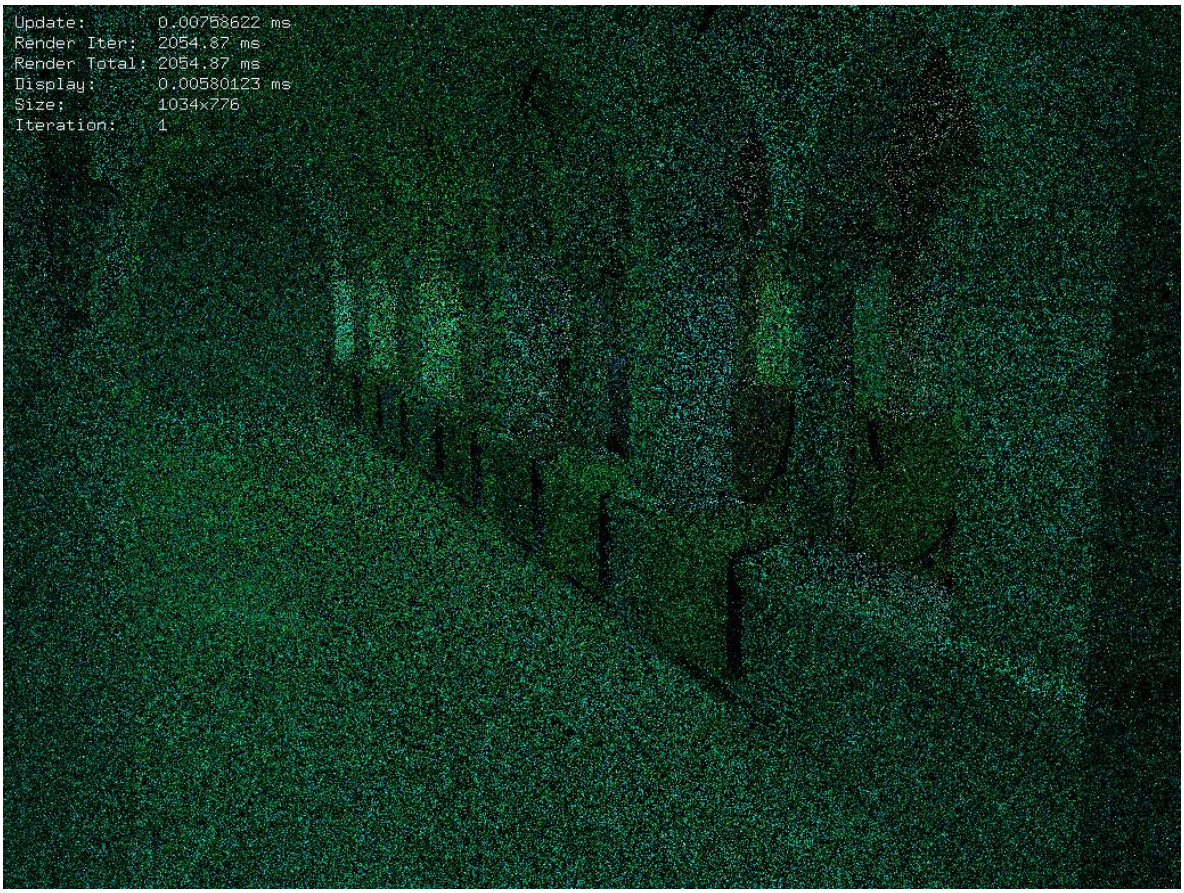
(b) Maya Software - Vista 2

Figura 5: Maya Software - Cenário com 7 fontes de luz.

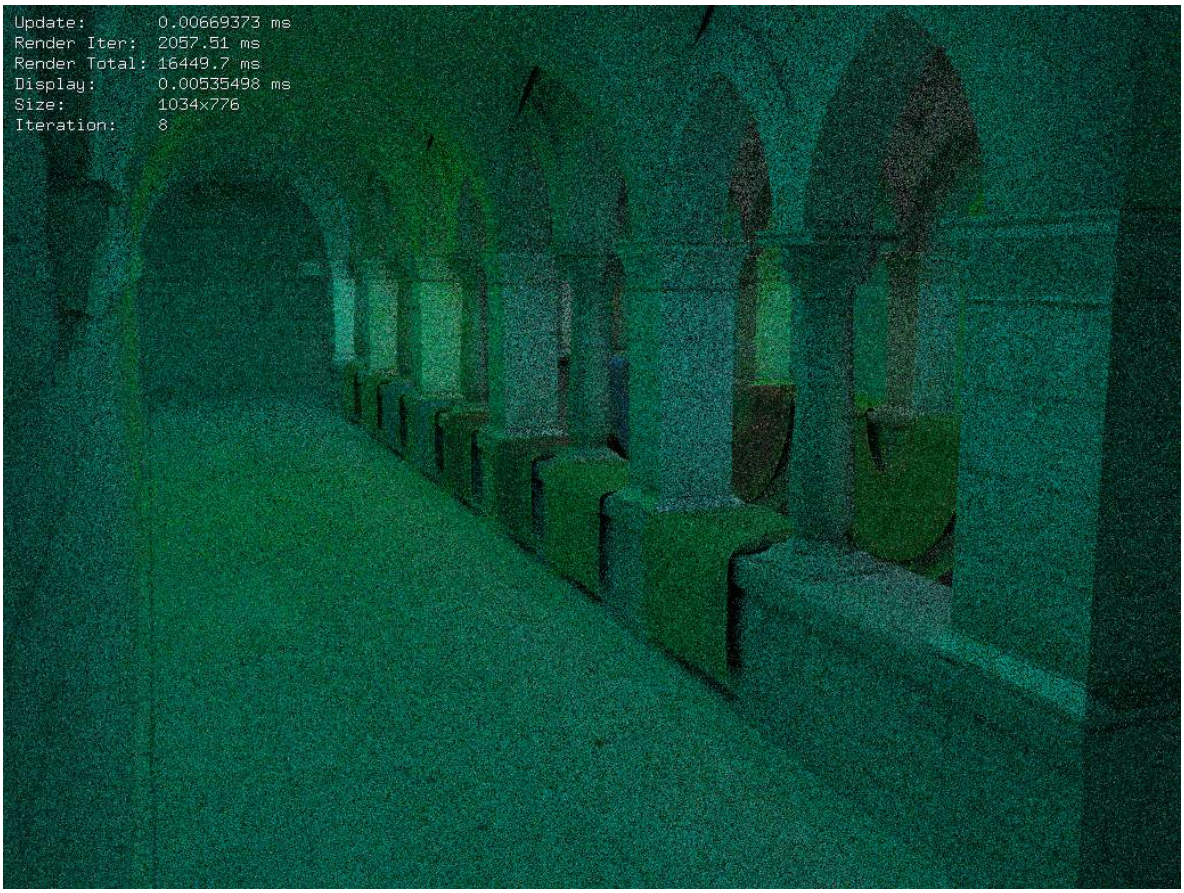
(a) Mental Ray - Vista 1

(b) Mental Ray - Vista 2

Figura 6: Mental Ray - Cenário com 7 fontes de luz.



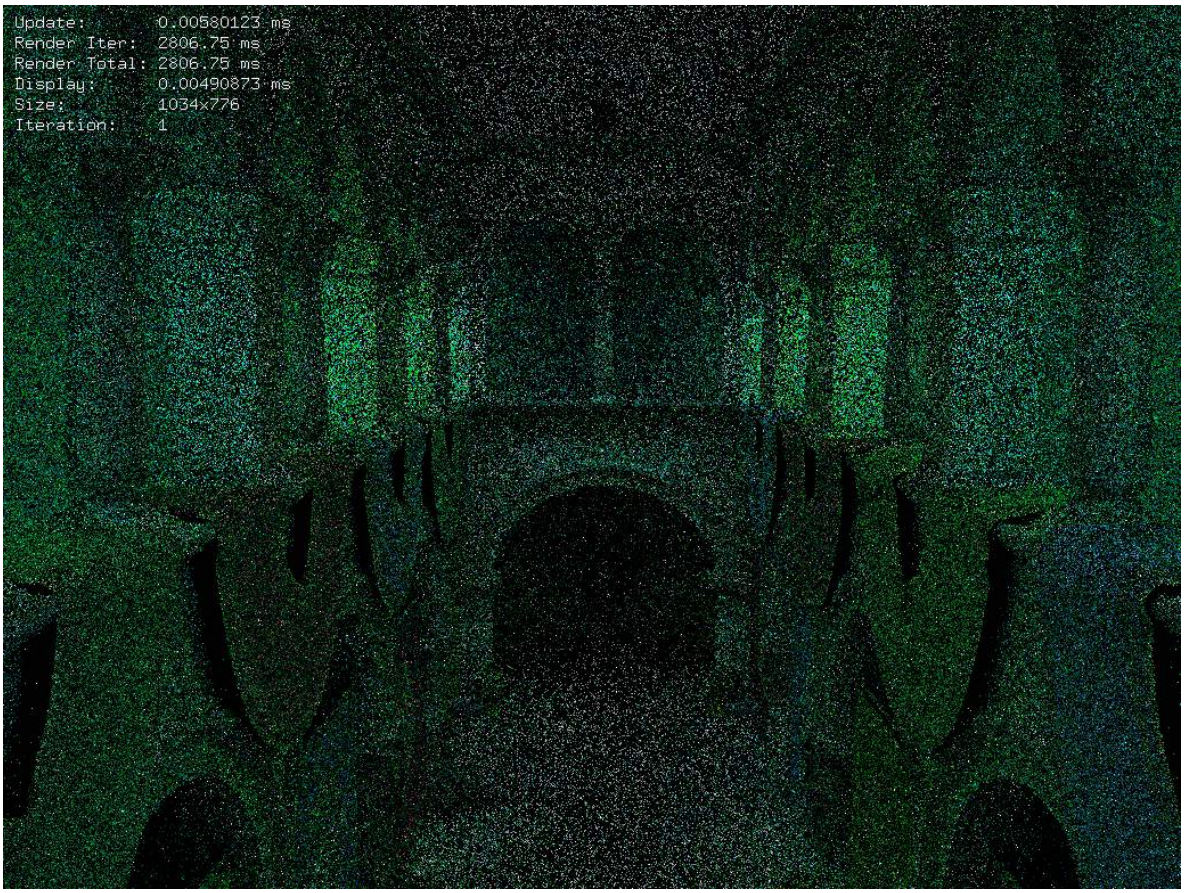

(a) Vista 1 - 1 iteração




(b) Vista 1 - 8 iterações

(c) Vista 1 - 32 iterações

(d) Vista 1 - 100 iterações




(e) Vista 2 - 1 iteração

(f) Vista 2 - 8 iterações

(g) Vista 2 - 32 iterações

(h) Vista 2 - 100 iterações

Figura 3: Implementação Dissertação - Cenário com 7 fontes de luz.

# 7

# CONCLUSÃO E TRABALHO FUTURO

Neste capitulo são apresentadas as conclusão do desenvolvimento desta dissertação e a perspectiva sobre trabalho futuro.

## 7.1 CONCLUSÃO

O principal objetivo desta dissertação é desenvolver uma aplicação de síntese interativa de imagens de alta qualidade e integrar com uma aplicação de criação de conteúdos tridimensionais, ativando para os utilizadores deste tipo de aplicações uma visualização de alta qualidade a taxas interativas dos conteúdos criados.

Para a implementação do processo de síntese foi utilizado o Nvidia Optix, esta API permite tirar partido das capacidades do GPU para efetuar cálculos intensivos. A utilização desta API mostrou ser bastante satisfatória na implementação de processos de síntese de imagem baseados em ray tracing a taxas interativas. Apesar de os resultados visuais possuírem ruído indesejável os tempos obtidos são satisfatórios quando falamos em imagens de alta resolução produzidas utilizando técnicas de iluminação global. A arquitectura do Optix mostrou ser bastante flexível quando queremos transformar um conjunto de dados num formato interpretável pelo Optix. No entanto foram verificados alguns problemas relacionados com a gestão da memória alocada pelo Optix, não fornecendo a API muitas ferramentas para lidar com situações em que a memória atinje o limite.

A integração com o Maya mostrou ser bastante acessível, muito pelo fato de este possuir uma poderosa API permitindo elevada flexibilidade na implementação. A leitura dos dados de geometria, materiais, texturas, fontes de luz e câmaras é de fácil acesso o que facilitou a interpretação dos mesmos e a utilização de cenários ja modelados existentes gratuitamente na internet. O Maya permite a implementação de novos materiais e é uma vantagem se queremos utilizar outros modelos de BRDF, evitando a tradução dos materiais existentes para os novos modelos. No entanto isto leva ao utilizador a ter de substituir os atuais materiais das geometrias e estes não vão ser interpretados pelos processos incorporados no Maya ou outros externos. A utilização do viewport do Maya revelou ser de fácil utilização e facilita a movimentação da câmara pelo cenário e interação com os elementos tridimensionais, no entanto se o tempo por iteração do processo de síntese aumentar consideravelmente, a utilização torna-se ineficiente ou nula pois o viewport do Maya fica bloqueado enquanto o processo de

síntese está a executar uma iteração. Tempos por iteração acima dos dois segundos mostraram tornar a utilização ineficiente e acima dos 5 segundos nula, pois elimina a interactividade com o software.

Em suma, a integração com o Maya mostrou bons resultados de interatividade e visualização quando não utilizamos elevadas resoluções. O processo de síntese de imagens implementado produz imagens com qualidade mas ainda com muito ruído, sendo aceitável para a pré-visualização de cenários tridimensionais. A arquitectura desenhada para esta aplicação de síntese de imagem revela ser eficiente e permite a integração com outras aplicação de criação de conteúdos tridimensionais ou outros processos de síntese de imagem.

## 7.2 TRABALHO FUTURO

O trabalho desenvolvido fornece uma base sólida para a implementação de diferentes processos de síntese de imagem ou extensão da atual implementação. O atual processo de síntese de imagem implementado pode ser melhorado ou extendido de várias formas incrementado a qualidade de imagem. Alguns pontos de trabalho futuro relacionados com o processo de síntese de imagem são a amostragem de importância, redução do ruído na imagem e utilização de outros modelos de BRDF. A utilização de um processo de síntese híbrido, entre ray tracing e rasterização é uma alternativa que pode produzir bons resultados. Em relação a integração com o Maya, a forma como é utilizado o viewport pode ser revista e a aplicação em geral pode melhorar suportando diferentes tipos de geometria, fontes de luz e materiais.

# BIBLIOGRAFIA


Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., 2008.

Mark Colbert, S Premoze, and G François. Importance Sampling for Production Rendering. *SIGGRAPH 2010 Course . . .*, 2010.

Peter Comninos. Physically based lighting and shading models and rendering algorithms. In *Mathematical and Computer Programming Techniques for Computer Graphics*, pages 317 – 421. Springer-Verlag London, 2006.

Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *ACM SIGGRAPH Computer Graphics*, 1984.

Philip Dutré. Global Illumination Compendium, Version k.4, 2001. URL http://www.graphics.cornell.edu/~{}phil/GI/.

David Gould. *Complete Maya Programming: An extensive guide to mel and the C++ API*. Morgan Kaufmann, 2003.

David Gould. *Complete Maya Programming Volume II: An in-depth guide to 3D fundamentals, geometry, and modeling*. Morgan Kaufmann, 2005.

HP and Nvidia. *Computer Graphics Masters*. Future Plc, 2014.

Henrik Wann Jensen and Niels Jø rgen Christensen. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers & Graphics*, 1995.

James T. Kajiya. The rendering equation. *ACM SIGGRAPH Computer Graphics*, 1986.

Alexander Keller. Instant radiosity. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 1997.

Daniël Jimenez Kwast. An Introduction to BRDF Models. *hmi.ewi.utwente.nl*, 2014.

Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. *ACM SIGGRAPH Computer Graphics*, 1993.

Rosana Montes and Carlos Ureña. An Overview of BRDF Models. 2012.

NVIDIA. *NVIDIA® OptiX™ Ray Tracing Engine, Programming Guide, Version 3.6*, 2014.

Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David Mcallister, and Martin Stich. OptiX: A general purpose ray tracing engine. *ACM Transactions on Graphics TOG*, 2010.

Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2010.

Tobias Ritschel and C Dachsbacher. The state of the art in interactive global illumination. *Computer Graphics Forum*, 2012.

Mike Seymour. The state of rendering, 2013. URL http://www.fxguide.com/featured/the-state-of-rendering.

Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, 1998.

Eric Veach and Leonidas J. Guibas. Metropolis light transport. *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, 1997.

Turner Whitted. An improved illumination model for shaded display. *ACM SIGGRAPH Computer Graphics*, 1980.

Chris Wynn. An Introduction to BRDF-Based Lighting. *Nvidia Corporation*, 2000.