

# **Critérios de Refinamento Progressivo para Navegação Interactiva em Ambientes Virtuais Complexos**

por

**António Araújo de Oliveira**

Dissertação submetida à Universidade do Minho  
para obtenção do grau de Mestre  
em Computação Gráfica e Ambientes Virtuais,  
elaborada sob a orientação científica do Professor:

**Luís Paulo Santos**

Departamento de Informática

Escola de Engenharia

Universidade do Minho

Braga, Setembro de 2006

É autorizada a reprodução integral desta tese apenas para efeitos de investigação, mediante declaração escrita do interessado, que a tal se compromete.

The reproduction of this thesis, in its whole, is authorized only for research purposes, upon written declaration from the interested part, which compromises itself to do so.

*non nova, sed nove*



# Agradecimentos

Tendo sido grande parte do trabalho desenvolvido no âmbito desta tese de carácter mais ou menos solitário, não cheguei a conhecer essa donzela, a solidão. Tecnicamente, tive os melhores conselheiros que se pode desejar; socialmente, tive os melhores parceiros que a sociedade alberga nas suas modestas instalações.

Em primeiro lugar, quero e devo agradecer profundamente ao meu orientador e supervisor, Professor Luís Paulo Santos, pela sua disponibilidade e envolvimento contínuo e pleno no evoluir do trabalho e pela compreensão demonstrada relativamente aos afazeres do dia-a-dia que me distraíam do normal fluir do mesmo. Sem a sua orientação, sem as suas contribuições, sem a sua visão “iluminada” e abrangente dos temas subjacentes a este projecto sentir-me-ia, concerteza, perdido neste mundo imenso, contudo maravilhoso, da computação gráfica. Quero agradecer ainda ao Professor Alberto Proença que, nos poucos encontros que tivemos durante este período, não deixou de partilhar connosco as suas ideias, muito oportunas, e conselhos úteis, particularmente no que diz respeito à avaliação e análise de resultados.

Uma parte significativa das minhas horas de trabalho no projecto tiveram lugar no Laboratório de Arquiteturas de Computadores, do Departamento de Informática da Universidade do Minho. Quero agradecer aos responsáveis pelo mesmo terem-me proporcionado livre acesso a este espaço onde sempre me senti confortável e bem-vindo por todos os que usufruíam do mesmo. Dos que partilharam comigo este espaço, quero referir os Professores João Miguel Fernandes e João Luís Sobral pela sua simpatia e por alguns bons momentos de descontração e alegre cavaqueira. Quero agradecer ainda aos colegas de labuta, Óscar e Sérgio, que, além dos bons momentos que partilharam comigo, sempre se dignaram convidar-me para o cafezinho da praxe.

A todos os professores do curso de mestrado de Computação Gráfica e Ambientes Virtuais, agradeço os preciosos ensinamentos que me transmitiram e com os quais pude entrar num novo mundo das Ciências da Computação, relativamente ao qual me sentia um pouco alheado nos últimos anos. Não quero ainda esquecer os meus colegas de curso. Pela sua

camaradagem, amizade e apoio, o meus cordiais agradecimentos e desejos de muitos sucessos nas suas carreiras profissionais. *Audaces fortuna juvat.*

Ao Francisco Gaitto, agradeço os muitos momentos de alegria e boa disposição que partilhou comigo nos “grandes almoços” de sexta-feira. Desejo-te o melhor que a vida pode dar e que os almoços não se acabem.

Aos meus colegas de trabalho, agradeço a preocupação demonstrada para comigo durante este período de muito trabalho, especialmente o João, a Carmen, a Filomena, a Luciana e toda a equipa do grupo nuclear Arboreto de Barcelos.

Aos pequenos cantores do Côro Juvenil de Oliveira com quem tenho passado alguns momentos de especial prazer e deleite musical, ao João Carlos e seus, e a todos os meus conterrâneos que souberam preservar os laços de amizade que nos ligam e que souberam manter a verdade e a honra acima de todas as coisas.

À minha família, agradeço as oportunidades e o bem-estar que me souberam proporcionar, desde sempre; ao meu pai, em especial, que me transmitiu o gosto pelo saber e que sempre me disse que o saber não ocupa lugar.

*Alea jacta est!*

# Resumo

Os algoritmos de iluminação global exigem tempos de computação elevados, resultando em grandes tempos de espera entre o momento em que o observador se posiciona num mundo virtual e o momento em que a imagem correspondente pode ser apresentada. Estes tempos de espera são um dos maiores obstáculos ao desenvolvimento de sistemas de navegação interactivos que recorram a modelos de iluminação de alta-fidelidade.

Este problema pode ser ultrapassado explorando a coerência espacio-temporal inerente a estas aplicações. A coerência espacial é explorada diminuindo a frequência de amostragem do mundo virtual, enquanto a exploração da coerência temporal é conseguida reutilizando amostras calculadas para imagens anteriores.

A subamostragem do mundo virtual é conseguida calculando a radiância reflectida na direcção do observador apenas nos vértices dos triângulos que constituem o modelo. Para efeitos de projecção no plano da imagem estes valores são interpolados, recorrendo ao equipamento gráfico do terminal de visualização. A interpolação resulta numa imagem de qualidade inferior à que seria obtida utilizando uma frequência de amostragem superior; fenómenos de iluminação importantes como sombras e texturas, que se caracterizam por altas frequências espaciais, podem não ser capturados. No entanto, este problema é ultrapassado aumentando a frequência de amostragem sempre que exista disponibilidade de poder de cálculo. Este aumento da qualidade da imagem ao longo do tempo é designado por refinamento progressivo e constitui o objecto de estudo da presente dissertação.

A selecção dos pontos do mundo virtual correspondentes a novas amostras obedece a algum critério de refinamento, que deve assegurar uma convergência rápida para uma solução de alta qualidade. A contribuição deste trabalho consiste na proposta e avaliação de três critérios de refinamento progressivo: um aleatório, um que opera no espaço da imagem e outro que opera no espaço dos objectos. As experiências realizadas e respectivos resultados permitem concluir que este último critério apresenta uma taxa de convergência mais elevada, consumindo simultaneamente menos recursos.



# Abstract

## Progressive Refinement Criteria for Interactive Navigation within Complex Virtual Worlds

Global illumination algorithms require huge computation requirements, resulting in large waiting times between the moment when the view point is set and the moment when the final image is ready to be displayed. These large waiting times prevent the development of interactive navigation systems that use high fidelity physically based illumination models.

This limitation can be overcome by exploring both spatial and temporal coherence, typical of such applications. Spatial coherence is exploited by subsampling the virtual world, while temporal coherence is exploited by reusing samples computed for previous frames.

Subsampling is achieved by computing the reflected radiance only at triangles' vertices. To project onto the image plane, these values are interpolated by the visualization computer graphics hardware. Interpolation results on a lower quality image than what would be obtained with a higher sampling rate; illumination effects such as shadows and textures, characterized by high spatial frequencies, are often missed. This problem can be solved by increasing the sampling frequency whenever there is available computing power surplus. Increasing the image quality with time is referred to as progressive refinement and is this thesis main motivation.

The selection of where to compute new radiance samples within the virtual world is guided by a refinement criterion, which must assure a fast convergence rate to a high quality solution. This thesis main contribution is the design and evaluation of three such criteria: a random one, an image space one and an object space criterion. Experiments and respective results show that the latter exhibits the best convergence rate, while simultaneously consuming less resources.



# Conteúdo

<b>Lista de Figuras</b>	<b>iv</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema a atacar . . . . .	2
1.2 Objectivos . . . . .	2
1.3 Abordagem do problema . . . . .	2
1.4 Contribuição do trabalho . . . . .	4
1.5 Estrutura da tese . . . . .	4
<b>2 O sistema de navegação interactiva</b>	<b>7</b>
2.1 <i>Renderer</i> . . . . .	8
2.2 SMA ( <i>Shading Management Agent</i> ) . . . . .	8
2.3 Visualizador . . . . .	9
2.4 O modelo arquitectónico . . . . .	10
2.4.1 Resenha histórica da Bracara Augusta . . . . .	10
2.4.2 O modelo virtual . . . . .	10
<b>3 Enquadramento</b>	<b>13</b>
3.1 Critérios de refinamento não perceptuais . . . . .	14
3.1.1 Interpoladores de Radiância . . . . .	14
3.1.2 Render Cache . . . . .	15
3.1.3 Tapestry . . . . .	17
3.1.4 Shading Cache . . . . .	18
3.1.5 Combinação de Arestas e Pontos . . . . .	20
3.2 Critérios de refinamento perceptuais . . . . .	21

3.2.1	Saliency Maps . . . . .	22
3.2.2	Aleph Maps . . . . .	24
3.2.3	Métrica perceptual adaptativa . . . . .	25
3.2.4	Snapshot . . . . .	26
<b>4</b>	<b>Abordagem</b>	<b>29</b>
4.1	Optimização do Sistema de Navegação . . . . .	29
4.2	Métodos de subdivisão . . . . .	31
4.3	Critérios de refinamento . . . . .	34
4.3.1	Critério aleatório . . . . .	35
4.3.2	A métrica NLD . . . . .	35
4.3.3	O critério NLD_IS . . . . .	36
4.3.4	O critério NLD_OS . . . . .	37
4.3.5	Optimizações . . . . .	38
<b>5</b>	<b>Análise de resultados</b>	<b>41</b>
5.1	Recursos utilizados . . . . .	41
5.2	Taxa de convergência . . . . .	41
5.2.1	Muralha da cidade . . . . .	43
5.2.2	Vista de rua . . . . .	45
5.2.3	Vista de templo . . . . .	47
5.3	Desempenho do critério NLD_OS . . . . .	48
5.4	Desempenho do sistema . . . . .	50
<b>6</b>	<b>Conclusão e trabalho futuro</b>	<b>55</b>
<b>A</b>	<b>Construção da métrica NLD</b>	<b>59</b>
<b>B</b>	<b>Imagens do processo de refinamento</b>	<b>61</b>

# Lista de Figuras

2.1	Sistema de Navegação Interactiva . . . . .	8
3.1	Render Cache . . . . .	16
3.2	Tapestry . . . . .	18
3.3	Técnica de arestas e pontos . . . . .	22
4.1	Tempos de resposta renderer versus cache . . . . .	30
4.2	Exemplo: operador de mapeamento de tons anterior . . . . .	31
4.3	Exemplo: operador de mapeamento de tons actual . . . . .	32
4.4	Subdivisão no ponto de intersecção . . . . .	32
4.5	Exemplo: subdivisão no ponto de intersecção . . . . .	33
4.6	Subdivisão binária de triângulos . . . . .	33
4.7	Propagação da subdivisão de triângulos . . . . .	34
4.8	Comparação das métricas NLD_IS e NLD_MM . . . . .	36
4.9	Comparação das métricas NLD_OS e NLD_MM . . . . .	38
4.10	Problema do critério NLD_OS . . . . .	39
4.11	Actuação do critério NLD_OS . . . . .	40
5.1	Vista da Muralha da Cidade - MC . . . . .	43
5.2	Vista de Rua - VR . . . . .	43
5.3	Vista de Templo - VT . . . . .	43
5.4	Posição MC: comparação de qualidade . . . . .	44
5.5	Posição VR: comparação de qualidade . . . . .	46
5.6	Critérios para a vista VR após cerca de 20s . . . . .	46
5.7	Posição VT: comparação de qualidade . . . . .	48
5.8	NLD_OS: optimizações . . . . .	48
5.9	NLD_OS: Potencialidades do critério NLD_OS . . . . .	50

5.10	Tempo de resposta do renderer . . . . .	51
5.11	Tempo de cálculo da visibilidade . . . . .	51
5.12	Tempo de aplicação do critério de refinamento . . . . .	52
B.1	Refinamento para a posição MC . . . . .	62
B.2	Refinamento para a posição VR . . . . .	63
B.3	Refinamento para a posição VT . . . . .	64

# Lista de Tabelas

5.1	Tempos do Radiance . . . . .	43
5.2	Resultados no visualizador para a Muralha da Cidade . . . . .	44
5.3	Resultados no visualizador para a Vista de Rua . . . . .	45
5.4	Resultados no visualizador para a Vista de Templo . . . . .	47



# Capítulo 1

## Introdução

Um dos principais desafios na área da Iluminação Global é a interactividade, isto é, a síntese de imagens fisicamente correctas e de alta qualidade a uma taxa mínima de 6 a 10 imagens por segundo. No entanto, a complexidade computacional dos algoritmos utilizados torna este objectivo muito difícil de atingir.

Uma das abordagens que tem apresentado algum sucesso é a utilização de esquemas de *caching* juntamente com o princípio de subamostragem e interpolação:

- A *cache* permite armazenar resultados que podem ser utilizados em imagens posteriores, tirando assim partido da coerência temporal que se observa na maior parte das animações.
- A subamostragem consiste em não calcular todos os pixels da imagem, mas apenas um subconjunto escolhido baseado em vários critérios: velocidade do *renderer*, detalhes e discontinuidades na imagem (reflexos, sombras, etc.). Os restantes pixels são calculados por interpolação, recorrendo frequentemente ao *hardware* gráfico.

A subamostragem resulta numa imagem com qualidade inferior à que seria obtida se a frequência de amostragem fosse mais elevada; este resultado é bastante óbvio pois muita informação de alta frequência pura e simplesmente não é capturada (incluem-se aqui, por exemplo, discontinuidades devidas a sombras, fenómenos especulares e mapeamento de texturas).

## 1.1 Problema a atacar

Aumentando a frequência de amostragem no plano da imagem, pode-se aumentar progressivamente a qualidade desta. Este processo é designado por refinamento e o problema a atacar é a selecção dos pontos do domínio da função onde o realizar. Esta selecção pode fazer-se no espaço da imagem ou no espaço dos objectos.

## 1.2 Objectivos

Pretende-se com este trabalho definir e testar critérios de refinamento progressivo da qualidade das imagens sintetizadas pelo aumento do número de amostras por *frame*. Os critérios a definir deverão ter em consideração:

- O impacto das amostras solicitadas na percepção que o utilizador tem da qualidade da imagem;
- O poder computacional disponível em cada instante para gerar as amostras seleccionadas, permitindo sempre uma visualização interactiva da cena.

O projecto implica a definição de um plano de ataque ao problema proposto, e a implementação de um protótipo que valide a estratégia adoptada.

## 1.3 Abordagem do problema

A observação do comportamento típico de um utilizador que navega num mundo virtual rapidamente permite concluir da elevada coerência entre frames consecutivas, isto é, muita da informação necessária para sintetizar a *frame*  $t$  é reutilizada para a *frame*  $t+1$ . A reutilização da informação permite que a frequência de amostragem seja aumentada, possibilitando um refinamento progressivo da qualidade da imagem. Adicionalmente, a própria qualidade das amostras pode aumentar progressivamente, partindo de amostras que apenas incluem iluminação directa e progredindo no sentido de aumentar as componentes de cada amostra (reflexão e transmissão especular, inter-reflexões difusas, etc.).

Neste trabalho, é usada uma arquitetura em três níveis [Coe05] que consiste num sistema de *rendering*, um sistema de gestão e armazenamento de amostras (em *cache*) e um sistema de visualização. Estes três sistemas trabalham em parceria, sendo o papel principal representado pelo segundo. Neste, para cada *frame*, é calculada a geometria visível com recurso

ao *hardware* gráfico. Para cada elemento geométrico (triângulo), se este já foi processado e a respectiva iluminação já está armazenada na *cache*, são utilizadas as respectivas amostras. Caso contrário, estas devem ser pedidas ao *renderer* e logo que estejam disponíveis devem ser guardadas em *cache* e enviadas para o sistema de visualização (ver secção 2.3). Terminada esta tarefa e não havendo mudança de estado, ou seja, o utilizador não tenha interagido entretanto com o sistema, começa o processo de refinamento. Este procura identificar os triângulos que ocupam zonas importantes da imagem (descontinuidades, sombras, reflexos, etc.) e procede à sua subdivisão de modo a aumentar a frequência de amostragem na zona correspondente da imagem. Após cada subdivisão são pedidas amostras ao *renderer* para os novos vértices criados. Logo que as amostras correspondentes aos vértices dos subtriângulos de um dado triângulo estão disponíveis, além de serem guardadas em *cache*, são enviadas para o sistema de visualização com o pedido consequente da remoção do triângulo inicial de cuja subdivisão estes derivam.

A selecção dos triângulos a subdividir pode ser feita com base na imagem actual ou com base na geometria visível subjacente à mesma imagem. Se esta é feita com base na imagem, então são seleccionados pontos da imagem segundo algum critério<sup>1</sup> e a partir de cada um desses pontos obtêm-se o triângulo que se projecta nesse ponto, procedendo-se à respectiva subdivisão. Se a selecção é feita com base na geometria visível, o triângulo a subdividir é directamente seleccionado com base em características deste tais como o tamanho (número de píxeis da imagem em que se projecta), a idade das respectivas amostras de *shading* ou com base na comparação dos valores individuais do *shading* de cada um dos seus vértices.

O processo de subdivisão não se limita a subdividir o triângulo seleccionado. De modo a manter a coerência da malha de triângulos do modelo, sempre que uma aresta de um triângulo é subdividida, se esta é partilhada por um triângulo vizinho, então também o vizinho é subdividido. Além de manter a coerência da malha, a subdivisão dos vizinhos acelera significativamente a taxa de convergência da qualidade da imagem obtida pelo facto de que a probabilidade de que os aspectos que levaram à selecção do triângulo inicial também estejam presentes nos seus vizinhos seja elevada. É através deste processo que quando, por exemplo, é encontrado o contorno de uma sombra ao subdividir um triângulo, este pode ser seguido através dos triângulos vizinhos.

---

<sup>1</sup>Critérios de carácter aleatório ou perceptual são passíveis de serem aplicados.

## 1.4 Contribuição do trabalho

Este trabalho apresenta um algoritmo eficiente para a identificação de descontinuidades geométricas e outras, tais como sombras e silhuetas. Este algoritmo tem como base um critério de refinamento que se baseia na diferença de luminâncias entre diferentes amostras de uma determinada área de uma imagem ou dos vértices de um determinado polígono. Verifica-se que a aplicação do critério no espaço dos objectos dá resultados bastante mais satisfatórios. Isto deve-se ao facto de que sendo o refinamento implementado com base na subdivisão de triângulos, ao aplicar directamente o critério aos triângulos visíveis, há uma maior objectividade na escolha dos triângulos a subdividir visto que estes, quando escolhidos, são necessariamente triângulos que interessa subdividir. Todavia, há a possibilidade de não seleccionar triângulos importantes só pelo facto de não haver diferenças significativas de luminância entre os seus vértices.

É apresentado um segundo algoritmo baseado no mesmo critério mas aplicado no espaço da imagem. A imagem é dividida sucessivamente em quadrantes cada vez mais pequenos (em cada nível divide-se cada quadrante em quatro subquadrantes) e para cada subdivisão é seleccionado o ponto central do quadrante que apresenta a maior diferença entre as luminâncias dos respectivos cantos na imagem. Com este método pretende-se refinar a imagem nas zonas com maiores variações de luminância. Notar que, mesmo que um quadrante não seja seleccionado para refinamento este é subdividido e os seus subquadrantes são também testados no sentido de serem ou não escolhidos. Tal assegura que nenhuma zona da imagem deixa de ser testada.

## 1.5 Estrutura da tese

O conteúdo restante desta tese está organizado do seguinte modo:

**Capítulo 2** - Faz-se uma descrição do sistema de navegação interactiva sobre o qual os critérios aqui apresentados foram implementados e testados.

**Capítulo 3** - Faz-se o levantamento do trabalho relacionado, destacando aqueles trabalhos cuja contribuição seja objectivamente aplicável neste projecto.

**Capítulo 4** - Descrevem-se os critérios de refinamento propostos neste documento e os métodos de subdivisão utilizados. Descreve-se ainda o conjunto de alterações complementares feitas ao sistema de navegação.

**Capítulo 5** - Apresentam-se os resultados obtidos em termos de eficiência, qualidade, e taxa de convergência.

**Capítulo 6** - São apresentadas as conclusões a este trabalho e apresenta-se um conjunto de ideias para trabalho a realizar no futuro.

**Anexo A** - Descreve-se o processo de desenvolvimento da métrica NLD utilizada na implementação de alguns dos critérios aqui descritos e sugeridos.

**Anexo B** - Apresenta-se a sequência de imagens relativas ao processo de refinamento para cada uma das três vistas consideradas na apresentação dos resultados. São apresentadas imagens relativas aos instantes 0, 1, 2, 3, 4 e 5 segundos de refinamento, e a imagem de referência obtida directamente pelo *renderer* utilizado neste projecto.



## Capítulo 2

# O sistema de navegação interactiva

O sistema de refinamento aqui especificado serviu de base ao sistema de refinamento implementado, integrado num sistema de navegação interactiva já existente [Coe05]. O referido sistema, inserido no projecto ViAr (Arqueologia Virtual), pretende comportar uma infra-estrutura robusta e eficiente para a navegação interactiva em ambientes virtuais complexos, particularmente modelos arqueológicos. Este sistema, com vista a suportar altas taxas de imagens por segundo, explora a coerência espacial e temporal baseando-se num esquema de sub-amostragem e de reaproveitamento de amostras guardadas em *cache*. A informação relativa às amostras é guardada no espaço dos objectos.

O sistema apresenta uma arquitectura em três níveis (figura 2.1):

**Visualizador** - envia alterações da posição do observador e recebe dados correspondentes a triângulos e respectiva iluminação, vértice a vértice. Os triângulos recebidos são apresentados ao utilizador recorrendo ao sistema gráfico da plataforma de visualização;

**SMA (*Shading Management Agent*)** - é o cérebro do sistema. Recebe indicações de mudança da posição do observador a partir do visualizador, procede ao cálculo da visibilidade e, em função dos resultados, envia as amostras que possui em *cache* para o visualizador e pede as restantes amostras ao *renderer*;

**Renderer** - recebe pedidos a partir do SMA e calcula radiâncias para esses pedidos.

O *renderer*, o SMA e o visualizador podem ser implementados em diferentes tipos de máquinas facilitando a visualização remota e o processamento paralelo.

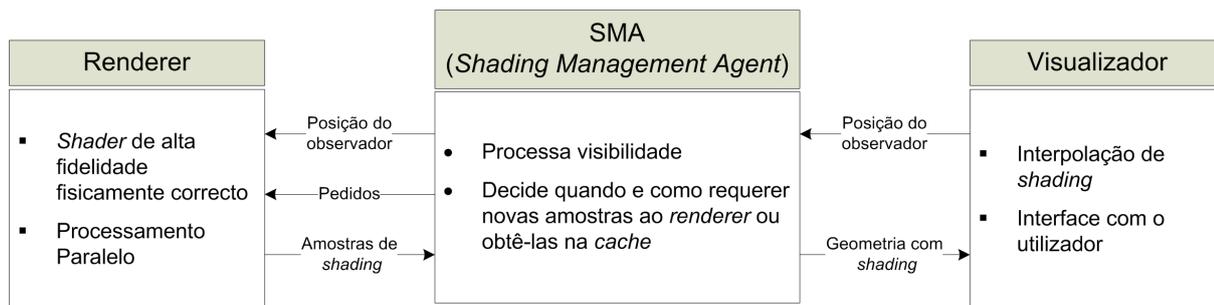


Figura 2.1: *Estrutura do Sistema de Navegação Interactiva.*

## 2.1 *Renderer*

O *renderer* utilizado no sistema descrito deve ser orientado ao ponto (*path tracing* ou *ray tracing*) e deve poder aceitar pedidos relativos a pontos distribuídos pelo plano da imagem ou pelo espaço dos objectos. Nesta implementação é usada uma versão alterada do *Radiance*<sup>1</sup> [War94, WS98] a qual aceita pedidos através de *sockets POSIX* e lança vários processos num *cluster* de estações de trabalho de modo a incrementar a taxa de *rendering*.

## 2.2 SMA (*Shading Management Agent*)

Duas funcionalidades básicas de um sistema de navegação interactiva são a visualização da imagem e o *rendering*. Neste sistema estas duas funcionalidades são executadas assincronamente, cada uma sendo executada à sua própria velocidade. Um processo adicional foi identificado e inserido entre os dois anteriores: o *Shading Management Agent* (SMA) que é um programa multiprocesso que se liga ao *renderer* e ao visualizador. Este, faz o cálculo da visibilidade e decide quando e quais as amostras que devem ser pedidas ao *renderer* ou obtidas da *cache* local onde estão armazenadas amostras previamente calculadas (exploração da coerência temporal). Estas amostras são guardadas no espaço dos objectos de maneira a

<sup>1</sup>O *Radiance* é um sistema de software para *ray tracing* com alto grau de exactidão. Este pretende ajudar projectistas e arquitectos de iluminação a predizer os níveis de luz e a aparência de um espaço antes da sua construção. O pacote de *software* inclui programas para modelar e transladar geometria na cena, dados sobre iluminação e propriedades dos materiais, os quais são necessários para a simulação. A própria simulação da iluminação usa técnicas de *ray tracing* para calcular os valores da radiância (isto é, a quantidade de luz que passa num determinado ponto, numa determinada direcção). Estes valores são combinados de forma a obter uma imagem fotográfica de qualidade. A imagem resultante pode ser analisada, exibida e manipulada dentro do próprio pacote de *software*, e convertida para outros formatos de imagem para exportação para outros pacotes de *software*.

evitar artefactos na reprojecção dos mesmos [WDP99, WDG02].

Ao mesmo tempo, as amostras correspondentes à geometria visível são enviadas para visualização. A versão actual do SMA só suporta um cliente de visualização; no futuro, deverá suportar múltiplos clientes que partilhem o mesmo modelo geométrico e as mesmas amostras em *cache*. Tirando partido da coerência temporal entre *frames* sucessivas, o SMA reduz o tráfego com o visualizador recorrendo a um protocolo de comunicação incremental: em vez de enviar toda a geometria visível para cada *frame*, limita-se a enviar os novos triângulos e a indicar quais os que deixaram de estar visíveis e que podem ser removidos.

Para além de fazer pedidos de novas amostras e de enviar a geometria visível para o visualizador, pretende-se ainda que o SMA tome decisões relativas ao aproveitamento dos tempos não utilizados nas tarefas anteriores, no sentido de utilizar este tempo para melhorar a qualidade da imagem exibida ao utilizador. É aqui que entra a principal contribuição deste trabalho. Recorrendo a um conjunto de critérios de refinamento, são tomadas decisões sobre os triângulos a subdividir com vista à obtenção de uma maior frequência de amostragem em determinada zona da imagem, a qual, segundo os critérios, é tida como mais importante em termos perceptuais ou apresenta boas probabilidades de o ser. Por exemplo, um triângulo que se projecte num número significativo de pixels da imagem apresenta boas probabilidades de conter algum tipo de descontinuidade no seu interior.

## 2.3 Visualizador

Fornecendo ao visualizador só a parte da geometria visível, o volume de tráfego entre o SMA e o visualizador é reduzido e a carga imposta ao visualizador é mantida no mínimo. Este limita-se a receber informação sobre triângulos, respectivos vértices e shading, a partir do SMA, e apresenta-os ao utilizador, no ecrã, recorrendo para isso ao equipamento gráfico disponível. Deste modo é explorada a coerência espacial por subamostragem no espaço da imagem: as amostras correspondentes aos vértices dos triângulos são rigorosas; o *shading* correspondente ao interior dos triângulos é obtido por interpolação de *hardware* [TPWG02].

Tendo em conta que a identidade e o número dos triângulos a exibir varia com o tempo, o visualizador necessita de fazer uma gestão eficiente destes. Para isso, estes são armazenados numa estrutura de dados de modo a que possam estar disponíveis para visualização até que seja recebida ordem de remoção dos mesmos.

Além da informação relativa à geometria, o visualizador recebe ainda informação sobre

o máximo valor de luminância sempre que este muda. Este valor é utilizado no algoritmo de mapeamento de tons (*tone mapping*) de modo a garantir que a imagem seja exibida dentro da gama de luminâncias admitidas pelo ecrã utilizado.

Pelo facto de o visualizador receber a geometria, esta é sempre rigorosamente reproduzida mesmo quando há mudança na posição do observador.

## 2.4 O modelo arquitectónico

### 2.4.1 Resenha histórica da Bracara Augusta

*Bracara Augusta* foi fundada pelo imperador Augusto, provavelmente no ano 16 a.C., no fim das guerras cantábricas. É possível que tenha sido criada já com direito latino. De qualquer modo, ou por privilégio de fundação ou, mais tarde, por promoção municipal, *Bracara Augusta* terá possuído estruturas de governo autónomas, com senado e magistrados.

Como capital de convento, exercia funções jurídicas, religiosas e económicas sobre um extenso território. Durante a primeira metade do século I, a cidade conheceu um povoamento sistemático, tendo então sido erguidos alguns dos seus edifícios públicos, ao mesmo tempo que iam sendo ocupados os seus quarteirões residenciais.

A cidade deve ter atingido a sua máxima expansão no séc. II.

A ocupação suévica não determinou o imediato declínio da cidade. Se algumas construções foram abandonadas, outras foram erguidas de novo, registando-se uma perfeita continuidade da vida urbana ao longo do século V.

Com o domínio visigodo, no séc. VI (585), a cidade persistiu ainda como metrópole eclesiástica. É possível que o centro de gravidade administrativo e religioso dos sécs. V/VI tenha privilegiado a zona da actual Sé-Catedral, facto que justificará a persistência dos quarteirões do quadrante nordeste da cidade romana, enquanto que as áreas periféricas parecem ter sido paulatinamente abandonadas.

### 2.4.2 O modelo virtual

O modelo virtual da Bracara Augusta foi desenvolvido no âmbito de um projecto de mestrado [Ber02], realizado no Instituto de Ciências Sociais da Universidade do Minho, em Braga. Este projecto surgiu como consequência de todo um trabalho que tem vindo a ser desenvolvido desde 1976, ano de criação do Projecto Bracara Augusta, projecto este

que visava conhecer melhor as diferentes fases da história da cidade bem como preservar os testemunhos arqueológicos ainda existentes. Posteriormente, o modelo geométrico foi exportado para o formato OBJ e a partir deste fez-se uma conversão para o formato do *Radiance* com os devidos ajustamentos tendo em conta as particularidades de cada formato, concretamente no que diz respeito a materiais e respectivas propriedades.

O modelo sobre o qual está a ser desenvolvido este projecto apresenta algumas lacunas que de algum modo condicionam o desenvolvimento do mesmo projecto tendo dado origem a algumas acções de compensação. Globalmente, o modelo apresenta os seguintes problemas:

- A geometria do modelo não está otimizada. Há muitos vértices ocultos cuja contribuição é altamente negativa pelo facto de a sua luminância ser sempre nula o que gera zonas pretas durante a navegação.
- Pelo facto de os vértices de cada triângulo não estarem ordenados de modo coerente, ou seja, segundo o sentido dos ponteiros do relógio ou segundo o sentido inverso dos mesmos, não é possível aplicar *culling* (em particular *backface culling*) ao modelo<sup>2</sup>. São óbvias as vantagens que se poderiam tirar da eliminação de geometria que nunca deve ser vista (por exemplo, o interior das paredes). Além das melhorias de desempenho haveria também melhoria na qualidade das imagens pelo facto de se evitar considerar as faces ocultas para questões de iluminação e refinamento.
- Há zonas incompletas no modelo como por exemplo colunas sem base de apoio, ou seja, penduradas no telhado ou suspensas no ar.

É possível que algumas destas lacunas tenham surgido durante o processo de conversão para os formatos actuais.

O modelo virtual da Bracara Augusta é constituído por um total de 1 140 583 triângulos e 671 261 vértices.

---

<sup>2</sup>Pelo mesmo motivo não é possível aplicar ao modelo algoritmos que recorram às normais dos triângulos. Tal facto, como veremos mais à frente, condicionará a utilização de determinados critérios de refinamento.



# Capítulo 3

## Enquadramento

Recentemente têm sido desenvolvidos esforços no sentido de desenvolver algoritmos de iluminação global orientados para aplicação em ambientes dinâmicos, quer explorando a coerência temporal pela redução ou eliminação dos cálculos redundantes e pela redução do *aliasing* temporal<sup>1</sup>, quer pela definição de métricas de erro que consideram aspectos temporais. A coerência temporal têm sido aplicada em *Ray Tracing* no domínio dos pixels para os quais a iluminação já tenha sido calculada previamente, em estruturas de dados para aceleração, e à iluminação global propriamente dita (selecção de amostras, cálculo de visibilidade, ou geração de números aleatórios). A aplicação da coerência temporal à iluminação global pode ser encontrado em vários trabalhos. Destes, destaca-se devido à sua relevância para este projecto, o *Render Cache* [WDP99, WDG02] e o *Shading Cache* [TPWG02].

Outras técnicas orientadas para aplicação em ambientes dinâmicos têm sido desenvolvidas. Entre estas destacam-se as técnicas orientadas à percepção. Estas técnicas procuram orientar o processamento para zonas importantes em termos de percepção por parte do observador humano. Têm em conta as características do sistema visual humano, quer físicas, quer psicológicas, de modo a que a imagem obtida seja o mais possível perceptualmente indistinta da imagem real correspondente, com um consumo de tempo e recursos bastante abaixo dos necessários para a obtenção da imagem total e perfeitamente manuseáveis em tempo real.

As técnicas orientadas à percepção concentram-se especialmente num aspecto importante designado por atenção visual. Ao observar uma imagem, o observador humano tende

---

<sup>1</sup>Variações bruscas entre imagens que se sucedem no tempo.

a ser atraído por determinadas zonas desta que apresentam certas características que as distinguem das zonas vizinhas. A atenção visual pode ser dividida em dois tipos de componentes:

- componentes *bottom-up*, orientadas a estímulos tais como a velocidade, a pré-atenção, e os mecanismos primitivos de resposta ao contraste de cor, contraste de intensidade, orientação e outros (mecanismos inconscientes);
- componentes *top-down* tais como a utilização de controlo cognitivo e a orientação à tarefa (mecanismos conscientes).

O modelo de atenção visual definido por Itti [IKN98] é uma das opções mais populares tendo sido utilizado em várias abordagens posteriores no campo da iluminação global interactiva [YPG01, LDC05, LDC06].

Tendo em conta a problemática abordada neste documento e a estrutura do sistema de navegação interactiva usado como plataforma de implementação e teste, faz-se um levantamento e análise de um conjunto seleccionado de documentos cujo conteúdo aborda o tema da iluminação global interactiva. Esta análise é orientada aos aspectos relacionados com os critérios de refinamento utilizados e respectiva implementação prática.

## 3.1 Critérios de refinamento não perceptuais

Nesta secção faz-se o levantamento de um conjunto de critérios apresentados em diversos trabalhos realizados na área da navegação interactiva e iluminação global e que exploram heurísticas baseadas em aspectos tais como a variação da radiância, a diferença entre frames sucessivas e outros aspectos de carácter estatístico ou aleatório.

### 3.1.1 Interpoladores de Radiância para Aceleração de Ray Tracing com Limitação de Erro

Em 1999, Bala *et al.* [BDT99] apresentaram um trabalho entitulado “*Radiance Interpolants for Accelerated Bounded-Error Ray Tracing*”, o qual visa permitir a interactividade através da aceleração do processo de *Ray Tracing* por exploração da coerência espacial e temporal. Para atingir este objectivo, são separadas as duas principais tarefas de um *ray tracer*: o cálculo da visibilidade e o cálculo da radiância em cada ponto visível.

Para acelerar o cálculo da radiância é explorada a coerência espacial dentro de cada *frame*. Como o *ray tracer*, a partir de uma determinada posição do observador, dispara pelo menos um raio através de cada pixel da imagem, se os valores de radiância variam suavemente então é possível obter novos valores, aproximados, a partir de um conjunto disperso de amostras calculadas anteriormente, por interpolação. Este conjunto de amostras de radiância é designado por *interpolant*. Quando a interpolação é passível de ser aplicada, a aproximação obtida evita a necessidade do cálculo da radiância pelo *ray tracer*, o que, em termos de tempo gasto no processo, representa uma poupança significativa. Obviamente, a interpolação introduz erros nos valores obtidos. Sendo assim, esta só é aplicada quando há garantias de que o valor de radiância obtido por interpolação está dentro dos limites de uma margem de erro  $\epsilon$  especificada pelo utilizador relativamente ao valor de radiância calculado pelo *ray tracer*. Variando o valor de  $\epsilon$ , o utilizador pode ajustar a relação qualidade/desempenho do sistema.

O cálculo da visibilidade é acelerado por exploração da coerência entre *frames* sucessivas, ou seja, quando muda a posição do observador, tipicamente, os objectos visíveis na *frame* anterior continuam visíveis na actual. Portanto, os interpoladores da *frame* anterior são reprojectados para a nova posição do observador de modo a obter rapidamente a visibilidade e radiância para os pixels da *frame* actual.

Os erros de interpolação advêm de descontinuidades ou não-linearidades na função de radiância. O algoritmo de limitação do erro previne a aplicação de interpolação nestes dois casos. A taxa de amostragem é aumentada onde o algoritmo de limitação do erro indica variações rápidas da radiância. Sendo a margem de erro indicada pelo utilizador um valor máximo, sempre que o sistema detecta que a posição do observador está estacionária é lançado um processo de refinamento que gradualmente vai diminuindo o valor de erro de interpolação permitido para produzir imagens de maior qualidade até que o limite de erro definido pelo utilizador seja atingido. Logo que a posição do utilizador mude, o sistema passa a processar a cena à maior velocidade possível, evitando a interpolação entre descontinuidades.

### 3.1.2 Render Cache

Resumidamente, a *Render Cache* [WDP99, WDG02] guarda em *cache* valores de *shading* calculados para pontos 3D, os quais, para cada *frame*, são reprojectados no espaço da imagem. Os casos em que há projecções coincidentes num mesmo pixel resolvem-se através de uma heurística de profundidade. Para os pixels em que não se projecta nenhum ponto os

respectivos valores são interpolados a partir dos valores vizinhos.

A *Render Cache* introduz uma nova estratégia de amostragem que combina vários esquemas de prioridade de actualização de diferentes pixels e usa um operador da difusão do erro (*error diffusion dither*) para mediar entre os objectivos conflitantes de obter uma distribuição uniforme para refinar a imagem suavemente no seu todo, e de concentrar as amostras em regiões mais importantes para obter uma convergência mais rápida da imagem. Tendo como base várias heurísticas que pretendem destacar os pixels que se suspeita serem propícios a conter artefactos visuais é gerada uma imagem de prioridades que orienta o pedido de novas amostras ao *renderer*. A prioridade de cada pixel é estabelecida com base na sua idade<sup>2</sup>

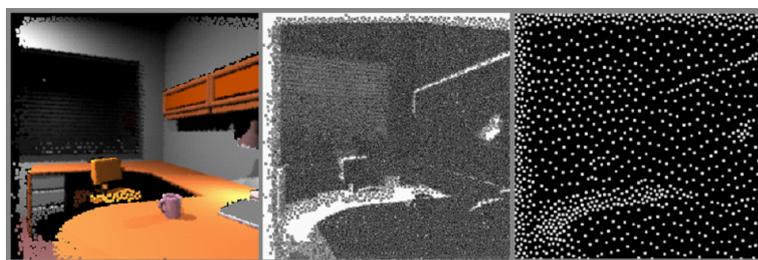


Figura 3.1: A imagem exibida, a correspondente imagem de prioridades e a localização das amostras a pedir.

(as amostras antigas têm maior probabilidade de terem mudado) e na quantidade de pixels vizinhos que possuem pontos em si mapeados (se não há vizinhos válidos, tem-se prioridade máxima; se há, tem-se prioridade média). Dada esta imagem de prioridades, faz-se uma escolha das amostras a pedir pela aplicação do algoritmo de difusão do erro atrás referido, o qual faz uma boa mediação entre os dois objectivos contraditórios de obter uma distribuição uniforme das amostras e de concentrar as amostras em regiões mais importantes à custa de ocasionalmente pedir o processamento de um pixel de baixa prioridade. Este algoritmo percorre os pixels da imagem, linha a linha em direcções alternadas, e a prioridade em cada pixel é comparada com um valor de fronteira  $t$  que depende da prioridade total e do número de amostras a pedir. Se a prioridade do pixel é superior a  $t$  este é seleccionado e o valor de fronteira é subtraído à sua prioridade. A prioridade que resta é propagada, metade para o pixel seguinte na linha de *scan* actual, e outra metade para o pixel equivalente ao actual na próxima linha de *scan*.

Com vista a detectar regiões em mudança e concentrar aí as novas amostras, acelerando

---

<sup>2</sup>A cada ponto em *cache* está associada uma idade que se inicia a zero e é incrementada em cada *frame*.

o processo de envelhecimento usado por defeito que se realiza a uma taxa constante, sempre que um ponto apresenta uma probabilidade significativa de estar fora de prazo é envelhecido prematuramente. Deste modo, o sistema acelera o novo cálculo ou a rejeição de tais pontos. Por exemplo, sempre que há uma nova amostra para um ponto que já contém uma, os respectivos valores são comparados com vista a detectar alterações. Se há alterações significativas então é provável que os pixels vizinhos também tenham mudado, procedendo-se então ao envelhecimento prematuro dos pontos mapeados nos pixels vizinhos deste. Para os novos pontos que não resultem da reamostragem de pontos antigos, a solução adoptada foi examinar em cada caso um subgrupo da *cache* e substituir o ponto mais antigo desse subgrupo. Esses subgrupos são organizados em forma circular (ordem *round robin*) de modo a que todos sejam considerados.

Além dos aspectos referidos, o envelhecimento prematuro também se aplica a pontos não visíveis na *frame* actual visto ser muito provável que estes já não sejam necessários. Por outro lado, podendo haver pontos com valores incorrectos ou fora de prazo, é usado um mecanismo que permite remover pontos da *cache* sem que haja um novo para o substituir. Esta decisão é tomada para um ponto quando este atinge a idade máxima (255).

### 3.1.3 Tapestry

Esta técnica [SS00] baseia-se na reconstrução de uma malha de triângulos 3D, dinâmica, cujos vértices correspondem aos pontos de amostragem, denominada *tapestry*, que serve como saída para o dispositivo de visualização e como *cache* que suporta a reutilização de amostras entre pontos de vista diferentes. A projecção da malha relativamente à posição do observador é gerida de modo a verificar a condição de *Delaunay* [Del34] com vista a melhorar a qualidade da imagem e assegurar uma maior robustez do código de manuseamento da malha. Com a navegação, a malha evolui continuamente de modo a providenciar uma reconstrução apropriada para o ponto de vista actual, dadas as amostras disponíveis. Este processo de reconstrução disponibiliza *feedback* para o *renderer* de modo a orientá-lo na selecção dos pontos mais necessários à obtenção de uma maior resolução. A cada triângulo da malha é atribuída uma prioridade. Para cada *frame*, os triângulos são processados num *back buffer*, calculando a respectiva prioridade a partir da componente de cor. A heurística baseia-se na diferença de cor e de profundidade dos vértices de cada triângulo. É utilizada uma medida do contraste para as diferenças de cor. As componentes individuais da cor são também pesadas em função do respectivo valor médio de modo a evitar sobre amostragem quando



Figura 3.2: *Componentes Tapestry: malha, imagem de prioridades (as áreas mais escuras indicam amostras de maior prioridade), conjunto de amostras, e imagem reconstruída.*

os valores de cor forem baixos. A prioridade resultante é calculada para os três vértices de cada triângulo:

$$p_c = 0.4 \frac{r_{max} - r_{min}}{r_{max} + r_{min}} r_{med} + 0.3 \frac{g_{max} - g_{min}}{g_{max} + g_{min}} g_{med} + 0.6 \frac{b_{max} - b_{min}}{b_{max} + b_{min}} b_{med} \quad (3.1)$$

Também é incorporado um termo baseado na diferença de profundidade dos vértices relativamente à posição actual do observador:

$$p_d = 1.0 - \frac{d_{min}}{d_{max}} \quad (3.2)$$

A prioridade final é obtida através de uma soma pesada<sup>3</sup> de  $p_c$  e  $p_d$  normalizada para o domínio  $[0.0, 1.0]$ :

$$p = w_c p_c + w_d p_d \quad (3.3)$$

### 3.1.4 Shading Cache

A *Shading Cache* [TPWG02] é uma malha de subdivisão hierárquica ligada a cada polígono da cena, com valores de *shading* para os respectivos vértices obtidos à velocidade possível (gerados por algoritmos de *rendering* orientados ao ponto). Esta malha é refinada pela escolha de localizações no espaço da imagem onde é necessária uma maior exactidão. A escolha destas localizações é feita através de um mapa de prioridades, que regista a probabilidade de erro devido a interpolação, à dependência da posição do observador e ao movimento dos objectos.

Durante a inicialização do sistema, a malha, para cada primitiva geométrica, é definida como sendo um polígono simples, não subdividido. Os valores de *shading* para a malha inicial

---

<sup>3</sup>Os autores propõem valores de  $w_c = 0,9$  para o peso do contraste e  $w_d = 0,1$  para o peso da razão de profundidade.

não são pré-calculados mas sim obtidos a pedido, de acordo com o evoluir da situação. A partir do momento que vão existindo valores de *shading* armazenados, cada polígono na malha armazena sempre os últimos valores de *shading* calculados para os seus vértices. Quando um polígono é seleccionado para actualização, são calculados os valores de *shading* para os seus vértices ou este é dividido em quatro sub-polígonos para cujos vértices são calculados os respectivos valores de *shading*.

Os valores na *Shading Cache* são actualizados de maneira a refinar progressivamente a imagem exibida. A actualização é feita através de um processo de refinamento orientado ao observador (*view-driven*).

O mapa de prioridades atrás referido é obtido através da obtenção de um valor de prioridade individual para cada polígono baseado no erro estimado para os valores de *shading* dos pixels onde este é projectado. Este erro é estimado como sendo a diferença entre os valores máximos e mínimos das cores dos vértices do polígono. Antes de calcular a diferença entre os valores máximo e mínimo das cores, é-lhes aplicado *tone-mapping* com o operador de Tumblin [THG99] com os seguintes pesos:

$$\frac{5}{16}R, \frac{9}{16}G \text{ e } \frac{2}{16}B$$

onde R, G e B são os canais de cor vermelho, verde e azul, respectivamente<sup>4</sup>. Após calcular a prioridade de cada polígono é feita uma normalização destes valores de modo que estas passam a ser tratadas como a probabilidade de seleccionar o respectivo polígono para refinamento. Na eventualidade de haver problemas no processo de normalização, calcula-se a média e a variância das prioridades e mapeia-se os valores 4 vezes o desvio padrão acima da média para 1 e os valores 4 vezes o desvio padrão abaixo para 0, isto em vez de mapear linearmente todas as prioridades entre os valores mínimo e máximo para a gama [0,1].

A selecção dos polígonos a refinar é feita através de uma abordagem pegar-e-testar (*hit-and-test*), ou seja, aleatoriamente selecciona-se um pixel do mapa de identificadores, gera-se um número aleatório entre 0 e 1 e, caso a prioridade normalizada do polígono correspondente seja superior a este valor, o polígono é seleccionado para refinamento. Assim, a probabilidade total de seleccionar um polígono é proporcional à sua prioridade e à área por ele projectada em pixels, logo, todos os pixels podem ser eventualmente seleccionados.

De modo a evitar tendências na amostragem pelo facto de a diferença entre os valores

---

<sup>4</sup>As diferenças visuais entre dois valores de radiância são linearmente proporcionais à diferença entre os valores *tone-mapped*.

dos vértices de alguns polígonos poderem ser praticamente iguais sem que o interior do polígono o seja, as amostras são aceites com uma determinada pequena probabilidade,  $\varepsilon^5$ , independentemente da sua prioridade. Assim, a prioridade total de seleccionar um polígono é dada por

$$p_{total} = \varepsilon + (1 - \varepsilon) \times p_{normalizada} \quad (3.4)$$

De modo a evitar que polígonos pequenos deixem de ser seleccionados, é executada uma segunda fase de selecção, no plano da imagem, que testa as localizações vizinhas das amostras de alta prioridade já seleccionadas e aceita-as caso a respectiva prioridade seja superior à da respectiva amostra de referência. Isto é realizado por um algoritmo de preenchimento (*flood filling*) de maneira que as regiões já visitadas não voltam a sê-lo. Este algoritmo termina quando a prioridade de todos os vizinhos estiver abaixo do valor de referência.

Relativamente aos efeitos resultantes da mudança da posição do observador, que só são considerados para objectos com superfícies não difusas, estes são resolvidos através do aumento da prioridade dos polígonos que constituem tais objectos. Associado a cada polígono é guardado um valor correspondente à idade das respectivas amostras em *cache*, sendo atribuída ao polígono uma prioridade proporcional a esse valor. Quanto ao problema do movimento dos objectos da cena, este é resolvido automaticamente pelo processo incremental de aumento da prioridade dos polígonos. Os polígonos mais prioritários acabam por ser calculados rapidamente o que faz com as mudanças de *shading* associadas a movimento, quer dos objectos quer do observador, se manifestem rapidamente.

Nos casos em que uma grande parte da cena muda devido ao movimento do observador ou de objectos, não podendo o *Renderer* responder com rapidez suficiente, limita-se o número de amostras pedidas baixando a resolução da *Shading Cache* nos objectos cujo *shading* está a mudar. A nova resolução é determinada com base num tempo de resposta especificado pelo utilizador. A um maior tempo de resposta corresponde uma maior resolução e vice-versa.

### 3.1.5 Combinação de Arestas e Pontos

A técnica de combinação de arestas e pontos [BWG02, BWG03] combina amostras de *shading* dispersas pela imagem (pontos) e descontinuidades calculadas analiticamente (arestas) para geração interactiva de imagens de alta qualidade. O *shading* em cada pixel

---

<sup>5</sup>São aconselhados para  $\varepsilon$  valores entre 0,01 e 0,05 (os autores usaram 0,03 em todos os exemplos).

é calculado por interpolação a partir das amostras de pontos vizinhos tendo em conta e respeitando as descontinuidades.

Essencialmente, o sistema consiste em dois processos: um processo externo responsável pelo *shading* e o processo responsável pelo processamento das arestas e pontos. O primeiro processo identifica os pontos visíveis e calcula a sua cor, assincronamente, sendo estes armazenados em *cache* e reprojectados no plano da imagem para cada *frame*. O segundo processo (*edge finder*), para cada *frame*, detecta as arestas de silhuetas e sombras para cada objecto visível. Em seguida, estas arestas 3D são percorridas e representadas a uma precisão superior ao pixel sendo, adicionalmente, usada informação de profundidade das amostras reprojectadas para remoção das arestas ocluídas. As arestas e as amostras dos pontos, representadas a uma precisão superior ao pixel, são combinadas para construir a imagem de arestas e pontos. A partir desta representação compacta é feita uma interpolação rápida e de alta qualidade dos valores de *shading* recorrendo a algoritmos de filtragem. Nestes algoritmos inclui-se a determinação de acessibilidade, que identifica as amostras que não são separadas por uma aresta de descontinuidade; a interpolação, que reconstrói o *shading* de cada pixel usando a acessibilidade calculada; e *anti-aliasing* baseados em tabelas.

A decisão relativa a quais as novas amostras de pontos que devem ser pedidas ao *shader* é tomada com base no retorno obtido da interpolação. A estratégia é essencialmente a mesma que é usada na *Render Cache*, excepto o facto das arestas influenciarem a distribuição das amostras. Tal como nesta, a prioridade associada aos pixels sem pontos em si projectados é inversamente proporcional ao peso total do filtro de interpolação antes da renormalização. Um maior número de amostras é concentrado em regiões onde o filtro de interpolação encontra uma densidade baixa de pontos e aos pixels sem amostras acessíveis é dada a prioridade máxima o que, obviamente, aumenta a amostragem na proximidade das arestas.

Esta técnica é uma forma bastante eficiente de, a partir do conjunto de amostras para pontos 3D existentes em *cache* e de uma estrutura hierárquica em árvore baseada em intervalos que armazena a informação relativa a arestas, obter *rendering* de alta qualidade a taxas interactivas.

## 3.2 Critérios de refinamento perceptuais

A generalidade dos critérios perceptuais utilizados nos últimos anos têm como base o trabalho de Itti *et al.*. Este apresentou um modelo eficiente para análise de imagens e identi-



Figura 3.3: *Arestas e pontos: a imagem de saída é calculada por interpolação das amostras dos pontos respeitando as arestas de descontinuidade.*

ificação das regiões da imagem visualmente mais importantes. Em seguida, são apresentados alguns trabalhos, incluindo o atrás referido e outros neste baseados, que apresentam um conjunto de critérios baseados em percepção visual.

### 3.2.1 Saliency Maps

Em 1998 foi apresentado um modelo de atenção visual<sup>6</sup> [IKN98] para análise rápida de cenas que se baseia na identificação de regiões destas que se destaquem em termos sensoriais do ponto de vista da percepção visual. Este modelo apoia-se no comportamento e arquitectura neuronal do sistema visual dos primeiros primatas. Este combina vários aspectos multi-dimensionais num simples mapa topográfico das zonas mais salientes<sup>7</sup> da cena. A cena é percorrida numa estratégia *bottom-up*, orientada pela maior ou menor proeminência de cada zona, e de um modo independente de qualquer tarefa. As estratégias de pesquisa visual dos seres humanos passam por decompor a entrada visual num conjunto de mapas de características topográficas. Diferentes localizações no espaço competem entre si para se

<sup>6</sup>Atenção visual (*visual attention*) é o processo de selecção de uma parte da informação visual disponível para localização, identificação e reconhecimento de objectos em determinado ambiente.

<sup>7</sup>Por saliência subentende-se, no contexto deste documento, uma zona de uma imagem que se destaca em termos visuais relativamente às zonas vizinhas e ao resto da imagem, ou seja, zona que capta a atenção do observador por estimulação visual.

destacarem dentro de cada mapa tal que só localizações que sobressaem relativamente às localizações vizinhas persistem. O conjunto de todos os diferentes mapas dá origem a um mapa geral, de forma puramente *bottom-up*, o qual codifica topograficamente a proeminência local ao longo de toda a cena visível. Em resumo, esta estrutura disponibiliza um método paralelo massivo para a selecção rápida de um pequeno número de localizações da imagem susceptíveis de atrair especialmente a atenção visual do observador, logo prioritárias em termos de *rendering*.

A entrada para o modelo consiste em imagens a cores, estáticas. Para cada imagem são criadas nove escalas espaciais usando pirâmides diádicas de *Gauss* que progressivamente, através de filtros passa baixo e subamostragem, produzem factores de redução horizontal e vertical da imagem desde 1:1 (escala 0) até 1:256 (escala 8) em oito oitavas. Cada índice da escala corresponde ao número de divisões por 2 feito à dimensão da imagem original.

Cada característica é calculada através de um conjunto de operações lineares “centro/arredores” afins aos campos visuais de recepção. Sendo os neurónios visuais típicos mais sensíveis numa pequena região do espaço visual (o centro), enquanto que os estímulos apresentados numa região mais ampla, concêntrica ao centro (arredores) e menos contrastante, inibem a resposta neuronal, a diferença entre escalas mais finas e as menos finas permite a extracção de características multiescalares pela inclusão de diferentes relações de tamanho entre o centro e as regiões circundantes.

O efeito “centro/arredores” torna o sistema visual altamente sensível a características tais como arestas, mudanças abruptas de cor e movimentos súbitos. Este modelo gera mapas característicos<sup>8</sup>, recorrendo aos mecanismos de “centro/arredores”, para canais importantes tais como a intensidade, a cor e a orientação. Cada uma destas características para cada um destes canais é calculada em múltiplas escalas e depois processada com um operador,  $\mathcal{N}(\cdot)$ , que imita o efeito da inibição lateral, ou seja, características semelhantes e vizinhas anulam-se mutuamente. Mapas característicos que possuam características salientes são realçados enquanto que mapas que possuam características semelhantes, não destacadas, são suprimidos. Por exemplo, um quadrado branco sobre um fundo preto seria destacado enquanto que um padrão de xadrez seria suprimido.

A soma dos mapas característicos para cada canal (intensidade, cor e orientação) dá

---

<sup>8</sup>Pode considerar-se que um mapa característico representa as proeminências numa imagem a diferentes escalas espaciais.

origem a um mapa de proeminências. Os mapas de proeminências, por seu lado, são também processados com o operador  $\mathcal{N}(\cdot)$  e adicionados de modo a obter um mapa de saliências único que quantifica a atenção visual. No total, são calculados 42 mapas característicos: seis para a intensidade, doze para a cor, e 24 para orientação.

A cada instante, o máximo no mapa de saliências define a localização mais proeminente na imagem, para a qual deve ser direccionada a atenção.

### 3.2.2 Aleph Maps

Yee *et al.* [YPG01] propuseram uma técnica baseada na sensibilidade espacio-temporal e na atenção visual para o processamento eficiente de ambientes dinâmicos. Esta técnica consiste na produção de um mapa espacio-temporal de tolerância ao erro, denominado *Aleph Map* que pode ser usado para orientar e acelerar o cálculo de iluminação global em animações que usem pré-processamento, tirando partido das limitações do sistema visual humano. A sensibilidade espacio-temporal indica a tolerância ao erro e a atenção visual indica as zonas sensorialmente importantes em termos visuais.

Esta técnica apresenta melhorias relativamente a outras, anteriormente apresentadas, sendo de destacar as melhorias em relação aos mapas de saliências pelo facto de acrescentar a estes informação temporal de modo a poder capturar frequência espacial e movimento.

O processo começa com a estimação de uma imagem rápida da cena<sup>9</sup> e prossegue em quatro passos principais: estimação do movimento, estimação da frequência espacial, estimação das saliências e cálculo do *Aleph Map*.

A estimação de movimento é feita por estimação de velocidade no espaço da imagem. Duas técnicas diferentes são apresentadas, a primeira aplica-se só a imagens, a segunda usa informação adicional tal como a geometria e o conhecimento das transformações usadas no movimento. Em ambas as técnicas, o objectivo é estimar deslocamentos de pixels de uma *frame* para a seguinte e depois calcular a velocidade da imagem a partir deste deslocamento, usando informação relativa à taxa de amostragem e à densidade do pixel.

A estimação da frequência espacial é feita pela aplicação da abordagem da pirâmide de *Gauss*. Cada nível da pirâmide é redimensionado para a dimensão da imagem original e a diferença absoluta dos níveis é calculada de modo a obter os 7 níveis de banda da pirâmide

---

<sup>9</sup>A imagem estimada pode ser obtida rapidamente usando OpenGL ou *ray tracing* só com iluminação directa.

Laplaciana. Cada nível desta pirâmide é então normalizado somando todos os níveis e dividindo cada nível pela soma para obter a estimação do conteúdo de frequência espacial em cada banda de frequências.

A estimação das saliências é feita por recurso a uma extensão do modelo computacional desenvolvido por Itti *et al.* (ver secção 3.2.1). A extensão incorpora um canal de características adicional, relativo ao movimento.

Finalmente, com os pesos da frequência espacial das respostas por banda, com as velocidades dos pixels da imagem e o mapa de saliências, é estimada a sensibilidade espaciotemporal do sistema visual humano (*Aleph Map*).

### 3.2.3 Métrica perceptual adaptativa

Em 2004, Farrugia *et al.* [FP04] apresentaram uma métrica adaptativa para aplicação no espaço da imagem, baseada em critérios perceptuais. De modo a evitar cálculos desnecessários, esta métrica só é calculada para os pixels que já possuem uma amostra válida (obtida pelo processo de *rendering* utilizado). A métrica utilizada baseia-se numa versão melhorada do modelo de visão de Lubin [Lub95], extendido para 4 canais, proposta por Patanaik *et al.* [PFFG98]. Embora tenha sido desenvolvido para *tone mapping*, a sua saída deste método é perfeitamente utilizável para o cálculo de distâncias pseudo-euclidianas entre duas imagens. Devido ao excessivo tempo de processamento necessário para este modelo, foi desenvolvido um método de cálculo adaptativo para avaliação da métrica de modo a poder ser utilizada em algoritmos de *rendering* progressivo.

O método, para cada uma de duas imagens a avaliar, consiste num primeiro passo que faz uma decomposição espacial (pirâmide de diferenças de Gauss) à totalidade da imagem e depois o plano da imagem é dividido numa árvore de quadrantes (*quad-tree*). Para cada célula da árvore é escolhido um número fixo  $N$  de amostras (dependendo do tamanho da célula). A distância, relativamente à outra imagem, é calculada para cada amostra e o valor médio das distâncias é associado à célula. Com base neste valor, utilizando o teste de homogeneidade descrito em [ARPT02] como critério de subdivisão, decide-se se a célula é subdividida ou não. O algoritmo pára quando não houver mais células para subdivisão.

Este método foi combinado com um algoritmo de *rendering* progressivo sendo a respectiva métrica aplicada a *frames* sucessivas, usando o mapa de distâncias para atribuir um valor de distância a cada triângulo visível. Uma primeira imagem é obtida numa fase de pré-processamento que procura obter uma maior densidade de amostras à volta das arestas

visíveis. Estas amostras são convertidas numa imagem aproximada através de uma triangulação de Delaunay e respectivo *shading* por interpolação de Gouraud. A segunda imagem é obtida por subdivisão da triangulação existente (obtendo uma nova amostra para o centro de cada triângulo) e nova interpolação da malha resultante. A métrica adaptativa é agora usada para calcular o mapa de distâncias entre estas duas imagens. Depois, um valor de distância é atribuído a cada triângulo, sendo este a média das distâncias dos pixels que ficam no interior do triângulo. A partir destes valores, os triângulos são classificados em três categorias usando como referência dois valores de *threshold*,  $T_0$  e  $T_1$ , que definem três domínios: os valores inferiores a  $T_0$ , os valores entre  $T_0$  e  $T_1$ , e os valores superiores a  $T_1$ . Conforme a distância associada a um dado triângulo pertence a um destes domínios, é-lhe aplicado um método de interpolação diferente (respectivamente, nenhuma, interpolação de Gouraud ou um processo de interpolação mais complexo que passa pela obtenção de amostras para todos os pixels localizados no contorno do triângulo em causa). Quando um triângulo tem menos de 5 pixels, são pedidas amostras para todos estes, não sendo aplicado qualquer tipo de interpolação. Um triângulo complexo (com distância superior a  $T_1$ ) é subdividido durante a próxima iteração.

Após todos os triângulos terem sido subdivididos e interpolados, a nova imagem é comparada com a anterior pelo mesmo método e o processo repete-se até que o valor da distância de todos os triângulos seja inferior a  $T_0$ . Se  $T_0 = 0$  então a imagem final será estritamente idêntica à imagem de referência, visto terem sido obtidas amostras para todos os pixels.

### 3.2.4 Snapshot

Em 2005, Longhurst *et al.* [LDC05] apresentaram uma técnica para orientação selectiva no processamento de iluminação global, denominada “Snapshot”. Esta técnica recorre à API do OpenGL como ferramenta para obtenção rápida de uma aproximação à imagem correspondente a cada *frame*. A esta aproximação é aplicado o método de Itti *et al.* (ver secção 3.2.1) com algumas melhorias e acrescentos [LDC06] e é gerado um mapa de saliências que é utilizado para orientar a selecção de novas amostras a requerer ao *renderer*. O modelo utilizado opera em três canais de cor, RG, BY e Luminância (Escuro/Claro). Um mapa de arestas substitui o canal de orientações no modelo de Itti *et al.*. Este foi desenhado de modo

a utilizar *fragment shaders* escritos em Cg<sup>10</sup> da Nvidia.

Para obter uma imagem aproximada (*snapshot*), que contenha um mínimo de informação que assegure que não há perdas potencialmente importantes, são incluídos no sistema cálculos básicos de sombreamento e reflexão. Além disto, com vista a compensar o facto de o OpenGL não considerar a iluminação indirecta ao gerar o *snapshot*, são adicionadas fontes de luz de emissão reduzida à cena. Cada uma destas fontes de luz ilumina directamente uma superfície na cena. Esta contribuição extra é particularmente importante para as superfícies que não recebem luz directa de qualquer fonte de luz primária.

---

<sup>10</sup>A linguagem Cg é indicada especialmente para aplicações de processamento de imagem visto incluir muitas funções matemáticas e rotinas de acesso rápido a texturas.



# Capítulo 4

## Abordagem

Pelo facto de este projecto estar integrado num sistema já existente, houve necessidade de estudar esse sistema de modo a obter uma visão clara das potencialidades e limitações do mesmo. Após este período de estudo concluiu-se que havia necessidade de proceder a algumas alterações no sistema inicial. Por um lado para aumentar a eficiência do mesmo, por outro, para permitir que este suportasse o processo de refinamento a implementar.

### 4.1 Optimização do Sistema de Navegação

O sistema de navegação [Coe05] que serviu de base a este trabalho, cuja estrutura pode ser visualizada na figura 2.1, não tendo sido construído a pensar particularmente no processo de refinamento a implementar, apresentava algumas lacunas que foram resolvidas ou compensadas.

No sistema inicial, o modelo Bracara Augusta era carregado para memória, no SMA (ver secção 2.2), considerando que cada vértice da geometria era partilhado por todos os polígonos vizinhos sem ter em conta a possibilidade de diferentes polígonos poderem pertencer a faces distintas, cujas características e materiais também podiam diferir. Tal situação dava origem a erros de *shading* durante a visualização do modelo. A rotina de carregamento do modelo foi alterada de modo a considerar a criação de uma cópia dos vértices partilhados sempre que estes sejam partilhados por polígonos com características (materiais ou texturas) diferentes.

Relativamente ao cálculo da visibilidade, no sistema inicial, este processava-se para cada nova posição do observador, de modo a calcular quais os triângulos visíveis de entre todos os triângulos da geometria de base. Tendo em conta que, com a implementação do processo de refinamento, os triângulos da geometria de base poderão ser subdivididos; quando um

triângulo da geometria de base está só parcialmente visível, ao ser subdividido, parte dos triângulos de subdivisão poderão não ser visíveis. Sendo assim, é importante que estes não sejam considerados como visíveis pelo facto de isto envolver custos, quer em termos de espaço, quer em termos de tempo de processamento e comunicação. Estes custos aumentam substancialmente conforme o nível de subdivisão aumenta. Para ultrapassar este problema foi implementado um segundo nível de visibilidade, o qual faz uma filtragem dos triângulos de subdivisão de modo a que só os efectivamente visíveis sejam considerados.

No visualizador, a lista de triângulos, pelo facto de estes serem inseridos e removidos da lista, deve ser ordenada pelo menos numa das operações pois, de outro modo, o processo de procura destes torna-se excessivamente demorada, em comparação com o tempo gasto na ordenação. O algoritmo de ordenação inicialmente usado era o *quicksort*. Tendo em conta que a lista estará sempre parcialmente ordenada e este algoritmo é muito pouco eficiente a ordenar listas parcialmente ordenadas, decidiu-se alterar o algoritmo de ordenação inicial e substituí-lo pelo algoritmo *heapsort*, o qual é muito eficiente na ordenação de listas grandes e parcialmente ordenadas. Esta alteração melhorou significativamente o tempo de resposta do sistema de visualização.

Na figura 4.1 pode-se observar um gráfico que apresenta os tempos relativos à resposta do *renderer* (primeira vez que os triângulos são pedidos) e à resposta da *cache* (pedido de triângulos já armazenados).

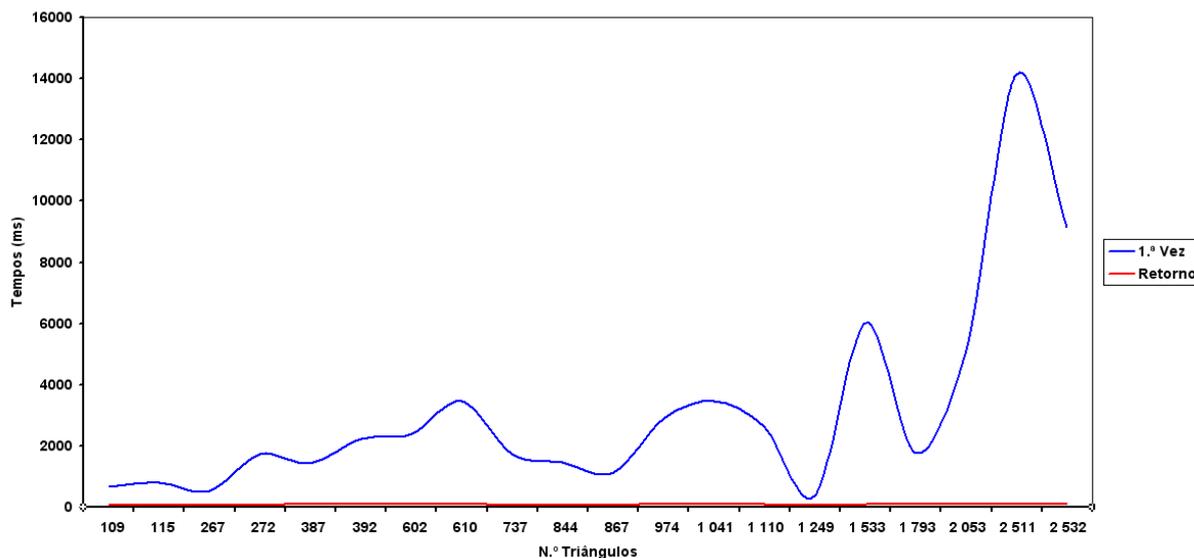


Figura 4.1: *Tempos de resposta renderer versus cache.*

Do gráfico pode-se claramente concluir que, enquanto que os tempos de resposta do *ren-*

*renderer* são proporcionais ao número de triângulos, os tempos de resposta da *cache* mantêm-se praticamente constante. É óbvio que se o número de triângulos aumentar significativamente também o tempo de resposta da *cache* aumenta, mas esse aumento é relativamente pequeno.

Outra alteração feita ao sistema inicial consistiu na substituição do operador global dinâmico de mapeamento de tons<sup>1</sup> de Reinhard [RSSF02] (figura 4.2):

$$L_d(x, y) = \frac{L(x, y) \left(1 + \frac{L(x, y)}{L_{white}^2}\right)}{1 + L(x, y)} \quad (4.1)$$

pelo operador referido por Drago *et al.* em [DMAC03] (figura 4.3) e que usa compressão logarítmica:

$$L_d(x, y) = \frac{\log(1 + L(x, y))}{\log(1 + L_{white})} \quad (4.2)$$

O segundo operador produziu resultados bastante mais satisfatórios como se pode ver nas figuras referidas onde aparecem dois exemplos: um aspecto exterior e um aspecto interior do modelo Bracara Augusta.

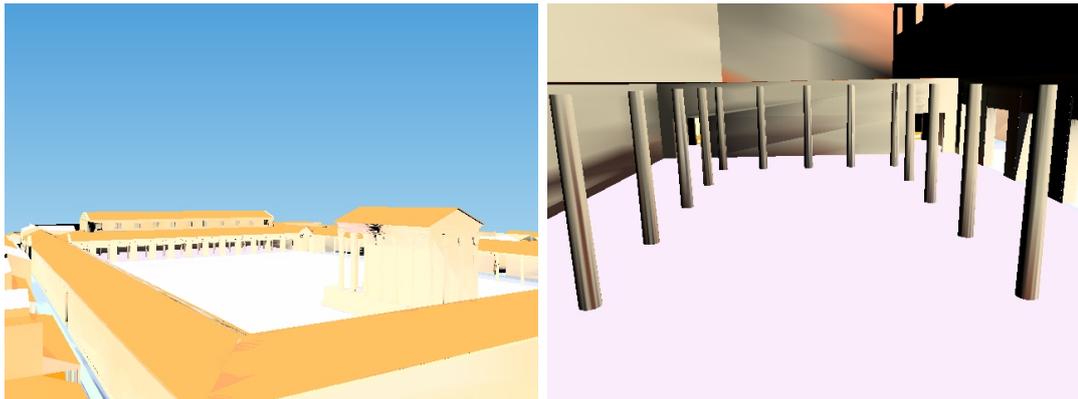


Figura 4.2: *Operador de mapeamento de tons anterior: aspecto exterior e aspecto interior.*

Nas equações 4.1 e 4.2,  $L_d(x, y)$  representa a luminância mapeada para o sistema de visualização correspondente à luminância real  $L(x, y)$  e  $L_{white}$  representa a menor luminância que será mapeada para branco puro.

## 4.2 Métodos de subdivisão

A reconstrução da imagem a partir de um conjunto disperso de amostras torna-se insuficiente em zonas onde ocorram elevadas frequências espaciais tais como limites de sombras,

---

<sup>1</sup>Visto que o *renderer* produz radiâncias no domínio  $[0, +\infty[$ , é necessário mapeá-las para o domínio  $[0, 1]$ , aceite pelos processadores gráficos. Esta operação é designada por *tone mapping*.

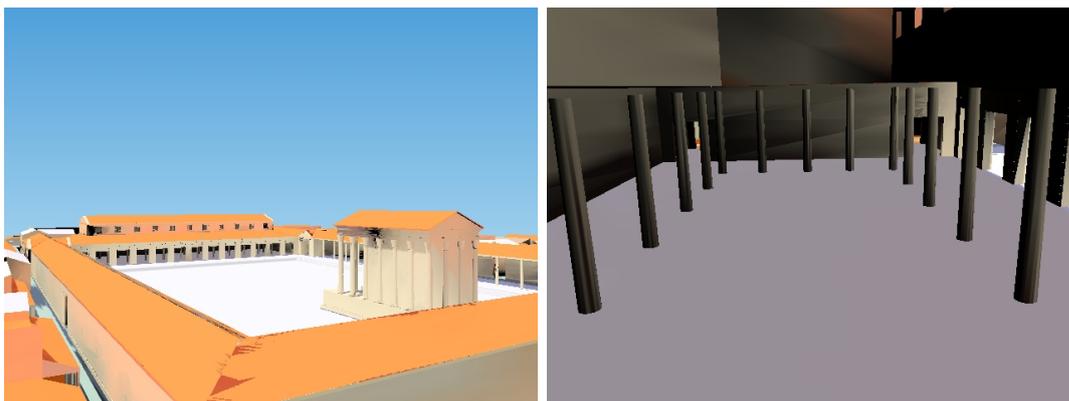


Figura 4.3: *Operador de mapeamento de tons actual: aspecto exterior e aspecto interior.*

materiais especulares (por exemplo, vidro, espelhos) e polígonos com texturas mapeadas. Tal pode ser compensado através do pedido progressivo de mais amostras ao *renderer*.

No SMA, a geometria é representada através de uma malha de subdivisão hierárquica que permite a subdivisão arbitrária dos triângulos originais. Como se verá na próxima secção, sempre que há tempo de processamento disponível (por exemplo, quando o utilizador pára numa determinada posição por algum tempo), o SMA selecciona alguns triângulos para subdivisão com base em algum critério de refinamento.

Relativamente ao processo de subdivisão, foram testados diferentes métodos de subdivisão de acordo com o critério de refinamento utilizado. Numa primeira abordagem, utilizando um critério puramente aleatório de selecção de pontos no espaço da imagem, testou-se a subdivisão no ponto seleccionado. Para tal, era pedido uma amostra de *shading* ao *renderer* para o ponto na imagem (era feito um pedido no espaço da imagem). Quando a amostra era recebida<sup>2</sup> no SMA, identificava-se o triângulo que se projectava nesse ponto e este era subdividido pelo ponto de intersecção (ver figura 4.4). Este método de subdivi-

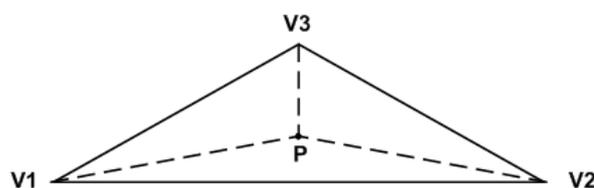


Figura 4.4: *Subdivisão do triângulo no ponto de intersecção.*

visão dá origem a uma malha muito dequilibrada. Por um lado, temos vértices partilhados

---

<sup>2</sup>Além do *shading*, também são recebidas as coordenadas do ponto no espaço dos objectos, sendo estas utilizadas na subdivisão do triângulo.

por um número excessivo de triângulos, e por outro, temos triângulos demasiado alongados cujo efeito na imagem é claramente desagradável (ver figura 4.5) pelos artefactos que gera. Assim, este método de subdivisão foi posto de lado, tendo-se optado por métodos de

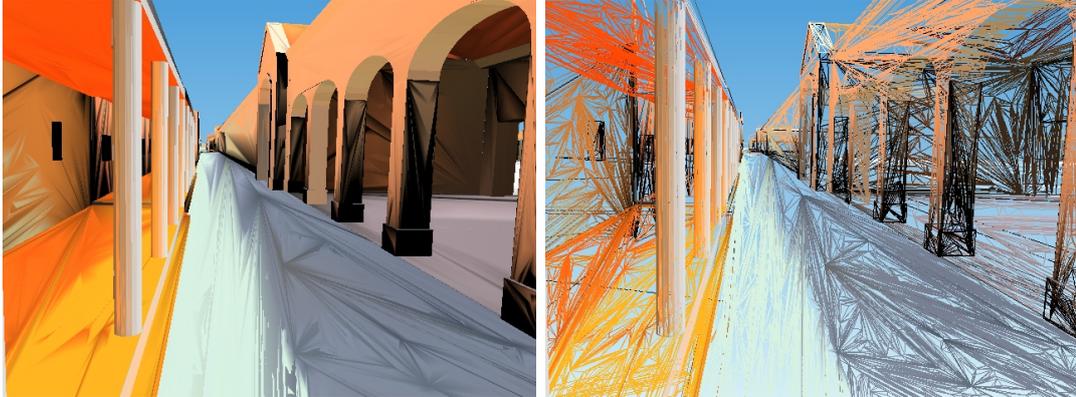


Figura 4.5: Resultados da aplicação da subdivisão de triângulos no ponto de intersecção.

subdivisão orientadas ao triângulo e não ao ponto de intersecção.

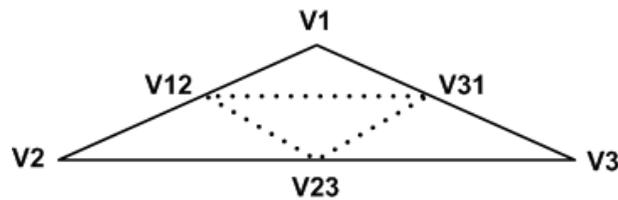


Figura 4.6: Subdivisão binária de triângulos segundo o método de Loop.

De modo a assegurar uma malha bem equilibrada e bem adaptada às necessidades do sistema em termos de refinamento progressivo, optou-se pelo método de subdivisão binária (diádica) de triângulos proposto por Charles Loop [Loo87], o qual divide o triângulo pelos pontos médios das suas arestas como pode ser visto na figura 4.6, dando origem a 4 sub-triângulos. Este método permite refinar uma malha de triângulos de um modo equilibrado, assegurando um bom balanceamento da mesma e permitindo detectar aspectos de interesse que possam intersectar as arestas dos triângulos subdivididos.

Obviamente, a subdivisão de um triângulo isolado dá origem a arestas em T. Para evitar que tal aconteça, é necessário assegurar que sempre que um triângulo é subdividido pelo método de subdivisão binária, se este possuir vizinhos, também estes devem ser subdivididos. Se a estes aplicarmos o mesmo método de subdivisão, teremos uma propagação exagerada do processo de subdivisão. Enquanto houver vizinhos, haverá subdivisão. Para evitar a propagação exagerada, optou-se por subdividir os vizinhos em dois subtriângulos (biseccção) em vez de quatro.

Com esta solução resolveu-se o problema da subdivisão excessiva mas surgiu outro problema. Apareceram artefactos resultantes da subdivisão em dois de triângulos alongados que resultava em triângulos ainda mais alongados. Para evitar esta situação e de modo a assegurar uma malha mais equilibrada passou-se a aplicar os dois métodos de subdivisão: o primeiro, método de Loop, sempre que o tamanho da aresta dividida seja inferior à altura do triângulo (tendo a mesma aresta como base) e o segundo, biseção, em caso contrário. Esta solução mostrou ser eficiente na manutenção de uma malha equilibrada e, por outro lado, assegurou uma eficiente técnica para detectar aspectos de interesse e seguir os respectivos contornos sem produzir demasiadas subdivisões inúteis, como veremos na próxima secção.

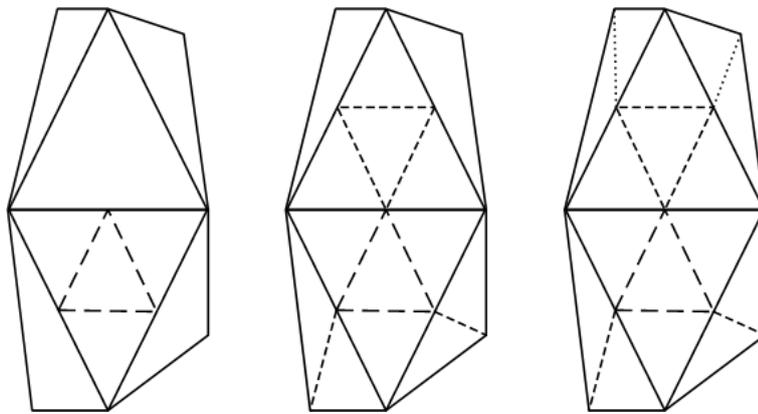


Figura 4.7: *Propagação da subdivisão aos triângulos vizinhos.*

Na figura 4.7, podemos ver da esquerda para a direita, primeiro a subdivisão do triângulo seleccionado para refinamento, depois a subdivisão dos triângulos vizinhos deste e, finalmente, o fim da propagação da subdivisão aos vizinhos.

### 4.3 Critérios de refinamento

Em termos gerais, o que se pretende é seleccionar um conjunto de critérios, viáveis e eficientes, que permitam melhorar progressivamente as imagens produzidas pelo sistema através de uma optimização da utilização do *renderer*. Os critérios de refinamento permitirão decidir a ordem de prioridades nos pedidos a fazer ao *renderer*.

A maioria dos critérios de refinamento usados em trabalhos anteriores operam no espaço da imagem e precisam, por um lado, de uma previsualização da imagem ao nível do SMA, baseada nos dados disponíveis, por outro, dependem da aplicação de operações de processamento de imagem demasiado exigentes para um sistema que se pretende interactivo.

Assim sendo, propõe-se a apresentação e avaliação de novos critérios de refinamento que sejam mais intuitivos e que não consumam demasiada percentagem dos recursos disponíveis. O primeiro critério é de carácter perfeitamente aleatório. Os outros dois baseiam-se numa métrica denominada “Diferenças de Luminância Normalizadas”<sup>3</sup>, um a operar no espaço da imagem e o outro no espaço dos objectos.

### 4.3.1 Critério aleatório

O primeiro critério de refinamento proposto é um critério aleatório (RND) utilizado como referência na avaliação dos outros dois critérios. Este opera no espaço da imagem e consiste na escolha aleatória de pixels da imagem, identificação do triângulo da geometria que se projecta neste pixel e, então, subdividir este triângulo segundo o método de subdivisão apresentado na secção anterior. Quando o número de pontos seleccionados atinge um valor definido como uma percentagem da dimensão da imagem (*largura* × *altura*).

### 4.3.2 A métrica NLD

Na *Shading Cache* [TPWG02] é atribuída uma prioridade a cada triângulo com base numa métrica que utiliza a diferença entre os valores máximos e mínimos da luminância nos seus vértices. A métrica aqui proposta, designada por  $S$ , usa as diferenças entre os valores das luminâncias e é aplicável a qualquer número de amostras na sua forma genérica<sup>4</sup>:

$$S = \sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N [L(x_i) - L(x_j)]^2}{(N^2 \operatorname{div} 4) L_{max}^2}} \quad (4.3)$$

onde  $N$  representa o número de amostras consideradas,  $L(x_i)$  representa a luminância da amostra  $x_i$  e  $L_{max}$  representa o valor máximo da luminância na imagem, sendo usado na normalização da métrica.

Enquanto que a métrica utilizada na *Shading Cache* é eficiente para 2 ou 3 amostras, a métrica representada pela equação 4.3 é eficiente para qualquer número de amostras. Tal facto permite que esta possa ser utilizada em diferentes critérios de refinamento. Neste documento apresentam-se dois critérios baseados nesta métrica. O primeiro, denominado NLD\_IS, aplica-se no espaço da imagem; o segundo, denominado NLD\_OS, aplica-se no espaço dos objectos. Com vista a avaliar a métrica, foi definido outro critério com base na métrica usada

<sup>3</sup>Normalized Luminance Differences (NLD)

<sup>4</sup>Para obter mais informações sobre a métrica NLD ver o anexo A.

na *Shading Cache*, denominada  $NLD\_MM$ . Neste critério utilizou-se a métrica  $NLD$  aplicada a dois valores, o máximo e o mínimo valores das  $N$  amostras, o que equivale a substituir na equação 4.3  $N$  por 2:

$$S = \sqrt{\frac{[\max(L(x_i)) - \min(L(x_i))]^2}{L_{max}^2}} \quad (4.4)$$

Esta métrica deixa de ser eficiente ao considerar casos com um número de amostras significativo pelo facto de só avaliar a diferença entre dois únicos valores quando se pode ter o caso de só um deles diferir dos outros. Sendo usada, aqui, como termo de comparação, esta métrica será utilizada para um número de de 3 e 4 amostras como veremos a seguir.

Ao fazer uma análise das curvas da equação 4.4 para diferentes números de amostras verifica-se que como aumento de  $N$  a curva tenderá a aproximar-se da curva  $y = x$ .

### 4.3.3 O critério $NLD\_IS$

O critério na espaço da imagem, denominado  $NLD\_IS$ , divide o plano da imagem em quadrantes e para cada quadrante calcula  $S$  usando os seus quatro cantos como pontos de amostragem. A equação 4.3, com  $N = 4$  fica:

$$S = \sqrt{\frac{\sum_{i=1}^3 \sum_{j=i+1}^4 [L(x_i) - L(x_j)]^2}{4L_{max}^2}} \quad (4.5)$$

Comparando a curva correspondente a esta equação e a correspondente à equação 4.4 (figura 4.8), construídas a partir de conjuntos de 4 valores aleatórios (um para cada canto do quadrante), verifica-se que o critério  $NLD\_IS$  destaca claramente as maiores variações, restringindo os valores mais elevados a um número restrito de casos enquanto que o critério  $NLD\_MM$  se comporta de uma forma mais ou menos linear.

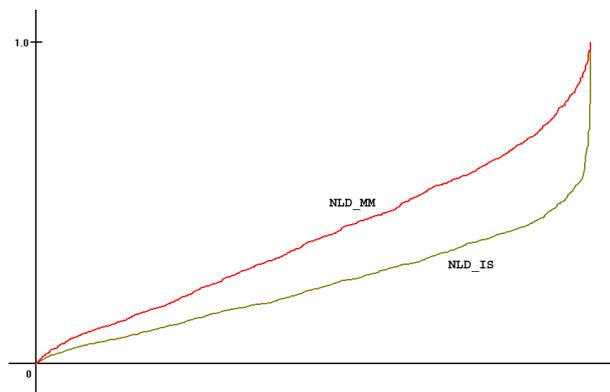


Figura 4.8: Comparação das curvas das métricas  $NLD\_IS/NLD\_MM$ .

O valor de luminância usada para cada canto, normalmente não irá corresponder a uma amostra exacta calculada pelo *renderer*; em vez disso, corresponderá a um valor interpolado pela placa gráfica do SMA a partir dos vértices do triângulo que se projecta no pixel correspondente. O quadrante que apresenta o maior valor de  $S$  é seleccionado para subdivisão, o que significa que o triângulo que se projecta no seu centro é subdividido de acordo com o método de subdivisão apresentado na secção 4.2. A subdivisão só é realizada se for satisfeito o critério de a área de projecção do triângulo no plano da imagem for superior ou igual a 6 pixels. Este processo continua hierarquicamente: cada quadrante no nível  $n$  é subdividido em 4 quadrantes (nível  $n+1$ ) e o mesmo procedimento é aplicado recursivamente. O processo termina quando a dimensão do quadrante é equivalente a um pixel ou uma dada fracção do total número de pixels da imagem tenha sido seleccionado.

Esta métrica assegura que todos os píxeis da imagem são testados, assegurando por isso, uma distribuição das novas amostras por todo o espaço da imagem desde que haja diferenças de luminância nesses espaços.

#### 4.3.4 O critério NLD\_OS

O critério na espaço dos objectos, denominado NLD\_OS, é calculado para cada triângulo visível usando os valores de *shading* dos seus 3 vértices. Um triângulo é seleccionado para subdivisão se o respectivo valor de  $S$  estiver acima de um determinado valor de referência  $S_{ref}$ . A equação 4.3, com  $N = 3$  fica:

$$S = \sqrt{\frac{\sum_{i=1}^2 \sum_{j=i+1}^3 [L(x_i) - L(x_j)]^2}{2L_{max}^2}} \quad (4.6)$$

Comparando a curva correspondente a esta equação e a correspondente à equação 4.4 (figura 4.9), construídas a partir de conjuntos de 3 valores aleatórios (um para cada vértice do triângulo), verifica-se que o critério NLD\_OS destaca ligeiramente as maiores variações, embora as duas curvas se aproximem pelo facto de o comportamento da curva NLD\_MM ser menos linear e mais próxima da curva NLD quanto menor fôr o número de amostras  $N$ . Logo que um triângulo é subdividido e os pedidos de *shading* para os respectivos vértices estão satisfeitos, estes novos triângulos tornam-se por seu lado também candidatos a subdivisão, passando pelo mesmo processo de classificação. O processo termina quando os  $S$  de todos os triângulos estão abaixo do valor de referência  $S_{ref}$ .

Este critério aplica a métrica NLD de forma progressiva de modo a assegurar que se começa pelos triângulos mais prioritários. Tal é conseguido pela definição de um valor de

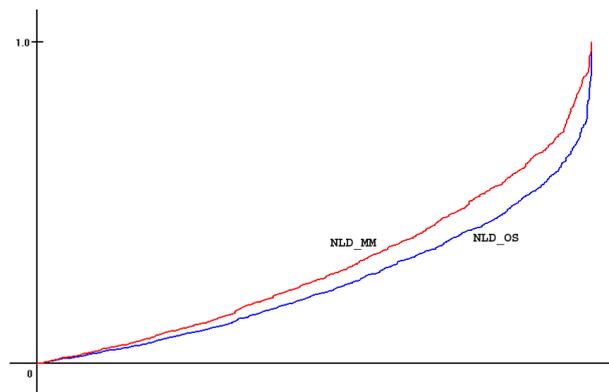


Figura 4.9: *Comparação das curvas das métricas NLD\_OS/NLD\_MM.*

referência inicial para a métrica e depois fazer com que este valor de referência vá sendo reduzido até um valor mínimo (quando este valor de referência mínimo é 0, a imagem será refinada até 1 amostra por pixel). Deste modo assegura-se que os triângulos são seleccionados por alguma ordem de prioridade e, por outro lado, controla-se a qualidade máxima atingida pelo refinamento por definição do valor de referência mínimo.

### 4.3.5 Optimizações

Com a progressão do processo de refinamento, nova informação fica disponível, como, por exemplo, uma maior densidade de shading no plano da imagem e uma malha de triângulos mais fina. O processo de refinamento só se aplica a triângulos marcados como visíveis e tal é determinado através da sua projecção no plano da imagem por recurso ao equipamento gráfico. A manutenção deste conjunto de triângulos, marcados como visíveis, o mais próximo possível do conjunto real de triângulos visíveis é um requisito de máxima importância para que a carga no processo de refinamento e o volume de dados enviados para o visualizador sejam minimizados. Todavia, quando um triângulo visível é subdividido em vários “filhos”, alguns destes podem não estar visíveis. Assim, é necessário que regularmente se reprojecte a nova malha, mais fina, de modo a determinar os triângulos efectivamente visíveis. Sendo assim, sempre que um dado número de novas amostras são recebidas, é feita uma nova projecção da malha.

Os critérios atrás apresentados requerem algumas optimizações relacionadas com o desempenho do sistema e com uma melhor selecção de triângulos a subdividir. Durante a aplicação do critério de refinamento, o SMA está focado nesta tarefa e fecha o acesso à estrutura de *cache* de modo que não pode realizar outras tarefas tais como a recepção de novas

posições do utilizador caso este interaja com o sistema de visualização e/ou a recepção e inserção de novas amostras em *cache* (recebidas do *renderer*). Tal facto pode dar origem a uma redução significativa da interactividade, o que não é aceitável tendo em conta o principal objectivo deste sistema que é responder rapidamente às acções do utilizador. Para ultrapassar este problema, o processo de refinamento é interrompido após a selecção de um determinado número de triângulos,  $R$ , para subdivisão e o controlo é passado ao ciclo principal do sistema. Neste ciclo, os novos pedidos de amostras são enviados para o *renderer*, as novas amostras recebidas são enviadas para o visualizador (se corresponderem a triângulos visíveis) ou, se fôr recebida uma mudança de posição do utilizador, procede-se ao processamento desta nova posição (cálculo de visibilidade, etc.). Se não houver mudança da posição do utilizador, o controlo é retornado ao processo de refinamento para seleccionar outro grupo de  $R$  triângulos. O valor de  $R$  tem impacto, quer no desempenho do sistema, quer no grau de convergência do refinamento, por isso deve ser escolhido cuidadosamente.

A aplicação do critério NLD\_IS, baseia-se na utilização da luminância projectada no espaço da imagem para decidir onde subdividir. Esta deve ser recalculada regularmente de modo a utilizar sempre a última informação disponível para a tomada de decisões. Esta operação é realizada sempre que um dado número de novas amostras são recebidas, ou seja, a reprojecção é feita só uma vez, sendo o cálculo da visibilidade e a obtenção da imagem de luminâncias, realizados em sequência directa.

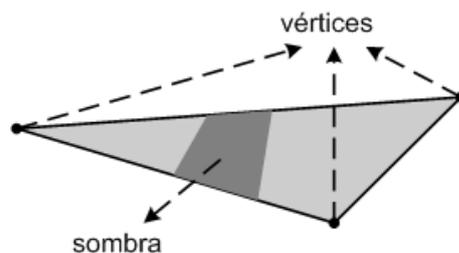


Figura 4.10: *Sombra a cruzar um triângulo sem afectar a iluminação nos seus vértices.*

O critério NLD\_OS apresenta como principal desvantagem o facto de que características importantes de *shading*, tais como sombras, podem aparecer nas faces de triângulos sem que se manifestem nos seus vértices, como pode ser visto na figura 4.10. Em tais casos, a métrica NLD, dada por  $S$ , indicará um valor mínimo e o triângulo não será seleccionado para refinamento, perdendo-se assim informação importante. Para minimizar este problema, uma optimização foi usada. Se um triângulo se projecta num número de pixels maior do que uma determinada fracção da dimensão da imagem, então este será sempre seleccionado

para refinamento, independentemente do valor de  $S$ . Assim, triângulos grandes são sempre subdivididos e, conseqüentemente, se algum ponto de subdivisão cair numa zona de interesse (uma sombra, por exemplo), então  $S$  irá detectar tal situação na próxima fase de refinamento e os triângulos “filhos” serão seleccionados normalmente para subdivisão. Notar que a partir do momento que um triângulo começa a ser subdividido, o método de subdivisão assegura que os seus vizinhos também serão testados (ver secção 4.2) permitindo que a partir do momento que uma característica de interesse é detectada, os triângulos que contém os seus contornos sejam refinados por subdivisão até ao nível do detalhe (ver figura 4.11).

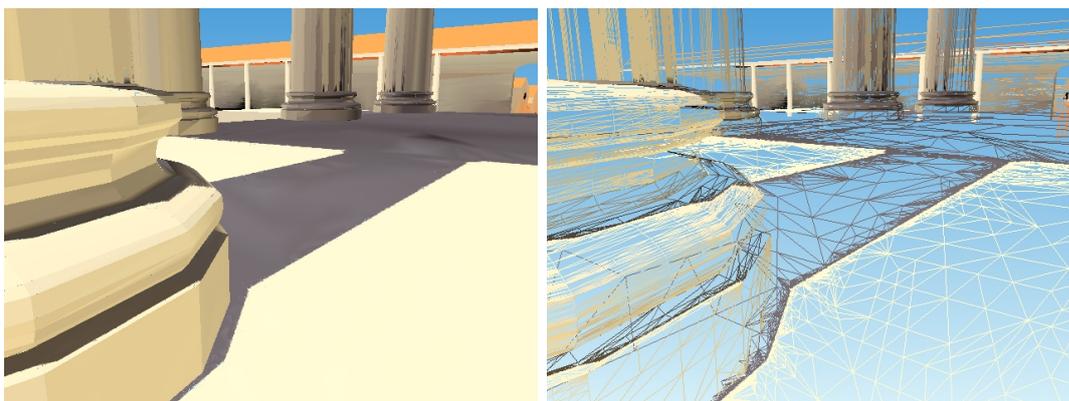


Figura 4.11: *Deteccção de sombras pelo critério  $NLD\_OS$ . À esquerda vê-se a imagem e à direita a malha de triângulos correspondente.*

# Capítulo 5

## Análise de resultados

Em face do objectivo principal deste trabalho ser critérios de refinamento, as experiências de teste realizadas e avaliadas no âmbito deste documento foram em parte realizadas para posições fixas do utilizador e não em navegação pelo modelo.

### 5.1 Recursos utilizados

A plataforma utilizada para implementação e teste do sistema aqui descrito consistiu em duas máquinas interligadas por uma rede Ethernet a 100 Mbits/s:

- o visualizador é um PC com um CPU AMD Athlon e um GPU NVidia GeForce 5200;
- o SMA e o *renderer* correm na mesma máquina, um processador dual-Xeon 3.2 GHz, com 2 GB de RAM e um GPU NVidia 6800GT.

Resultados relativos a navegação interactiva e *rendering* paralelo com Radiance nesta plataforma (sem refinamento) podem ser encontrados em Santos *et al.* [SCBP05] e Coelho [Coe05].

### 5.2 Taxa de convergência

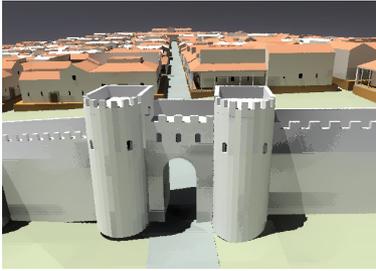
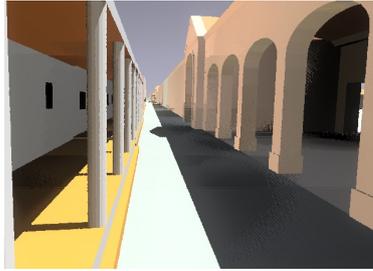
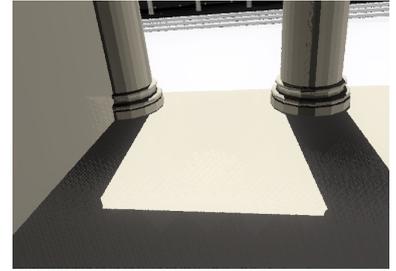
Para avaliar o sistema, relativamente à taxa de convergência e qualidade, foram feitas leituras e obtidas imagens para 6 instantes e para cada critério, a partir do ponto de vista do utilizador do sistema (no visualizador). Estes instantes correspondem a, respectivamente, 0, 1, 2, 3, 4 e 5 segundos após o início do processo de refinamento. Além dos 6 instantes referidos, são ainda apresentados os tempos e número de triângulos para o momento que o

critério de paragem é aplicado (ver secção 4.3). Como termo de comparação foram ainda obtidas as imagens correspondentes às mesmas posições do utilizador directamente com o Radiance (com uso das mesmas configurações) e respectivos tempos. Relativamente à comparação entre imagens obtidas com os critérios e a imagem obtida com o Radiance, é de salientar que, enquanto que com os critérios se obtém uma resposta visual em poucos segundos e se pode navegar em qualquer instante, com o Radiance temos de esperar a totalidade do tempo para ver a imagem com a qualidade final.

É de notar que neste contexto devemos considerar a taxa de convergência como sendo uma medida empírica que relaciona a qualidade subjectiva com o número de triângulos permitindo comparações, necessariamente subjectivas, da eficácia relativa de cada um dos critérios. Para cada critério é feita uma ordenação subjectiva da qualidade relativa, atribuindo as classificações melhor (Q1), médio(Q2) e pior (Q3) a cada um dos 3 critérios. Estas classificações são atribuídas por comparação com a imagem obtida directamente pelo Radiance e cujos tempos são os da tabela 5.1.

Apesar de os testes terem sido realizados para 20 posições diferentes do utilizador relativamente ao modelo, aqui são apresentados resultados para apenas 3 dessas 20 posições (figuras 5.1, 5.2 e 5.3, respectivamente), escolhidas de modo a destacar diferentes aspectos ou particularidades em termos de refinamento. A avaliação das imagens obtidas é realizada de modo empírico relativamente à qualidade das descontinuidades visuais encontradas nas diferentes cenas pelo processo de refinamento, especialmente a qualidade das sombras. A avaliação comparativa dos diferentes processos de refinamento é feita em função dos seis instantes de controlo atrás referidos, considerando o número total de triângulos exibidos no visualizador ( $\Delta_{Vis}$ ) e o número total de triângulos enviados para o visualizador para atingir o presente estado ( $\Delta_{Ins}$ ). Este número inclui triângulos que entretanto foram subdivididos e, portanto, removidos ( $\Delta_{Rem}$ ) do visualizador em algum momento. Assim,  $\Delta_{Ins}$ , reflete o número de triângulos cuja iluminação foi obtida do *renderer* e enviada através da rede.

As três diferentes posições do utilizador para as quais se apresentam resultados são designadas por MC (Muralha da cidade), VR (Vista de Rua) e VT (Vista de Templo). Na vista MC destaca-se o número elevado de triângulos que estão dentro do campo de visão e a sua posição relativamente à posição do observador. Na vista VR destacam-se as sombras que as colunas e os telhados projectam na rua. Na vista VT destaca-se a grande sombra que duas colunas e o tecto do templo projectam e que não é capturada sem o processo de refinamento.

Figura 5.1: *Vista MC.*Figura 5.2: *Vista VR.*Figura 5.3: *Vista VT.*

MC	VR	VT
55	40	39

Tabela 5.1: *Tempos do Radiance, em segundos, para as 3 vistas.*

Como complemento às secções que se seguem pode ser observado a evolução da aplicação dos três critérios em análise aos três pontos de vista escolhidos nas figuras B.1, B.2 e B.3, em anexo. É de referir que ao analisar estas imagens deve ser tido em conta que, sendo os critérios NLD\_OS e NLD\_IS determinísticos, ou seja, produzem sempre os mesmos resultados quando aplicados à mesma cena, o critério RND, sendo aleatório, produz sempre resultados distintos o que justifica o facto de a evolução do refinamento não ser tão óbvia como nos outros dois.

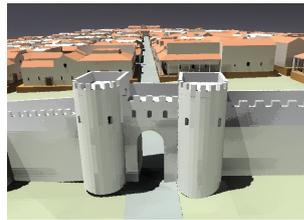
### 5.2.1 Muralha da cidade

Nesta posição do observador, o modelo de base possui 2 511 triângulos visíveis. Analisando a figura 5.4 repara-se que ao fim de 5 segundos a imagem correspondente ao critério NLD\_OS é melhor que a correspondente ao critério NLD\_IS e bastante melhor que a correspondente ao critério RND. Além disso, pode-se verificar pela tabela 5.2 que a uma maior qualidade corresponde também um menor número de triângulos para o critério NLD\_OS. O critério RND apresenta um número bastante elevado de triângulos e apresenta um número elevado de triângulos transferidos entre o SMA e o visualizador. Tal acontece pelo facto de este critério ser bastante mais rápido uma vez que não envolve quaisquer processamento comparável ao que envolve os outros dois critérios. Simplesmente, são gerados dois números aleatórios correspondentes às coordenadas do ponto a seleccionar na imagem. Este é um processo extremamente rápido comparando com, por exemplo, o critério NLD\_IS que necessita de obter uma pré-visualização da imagem no SMA e a partir dessa imagem obter as

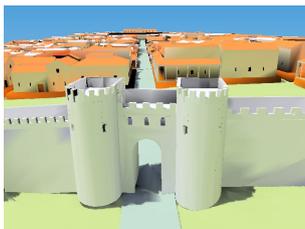
luminâncias a usar no critério.

Tempos (s)		0	1	2	3	4	5
NLD_OS	$\Delta_{Vis}$	2 511	3 975	4 703	5 504	7 791	8 605
	$\Delta_{Ins}$	2 511	4 742	5 996	7 473	11 229	12 863
	$\Delta_{Rem}$	0	767	1 293	1 969	3 438	4 258
NLD_IS	$\Delta_{Vis}$	2 511	5 093	6 938	9 506	12 545	13 087
	$\Delta_{Ins}$	2 511	5 844	8 924	12 653	17 173	17 975
	$\Delta_{Rem}$	0	751	1 986	3 147	4 628	4 888
RND	$\Delta_{Vis}$	2 511	6 672	14 731	18 023	23 632	28 457
	$\Delta_{Ins}$	2 511	8 660	20 942	26 130	35 230	42 970
	$\Delta_{Rem}$	0	1 988	6 211	8 107	11 598	14 513

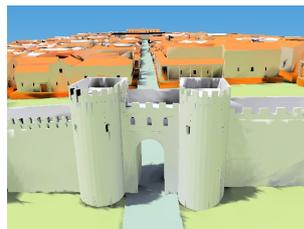
Tabela 5.2: Número de triângulos no visualizador para a posição MC.



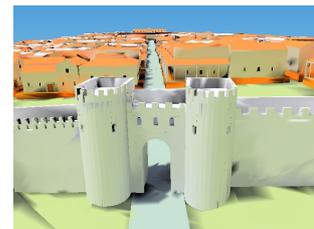
(a) Radiance



(b) Q1: NLD\_OS



(c) Q2: RND



(d) Q3: NLD\_IS

Figura 5.4: Comparação de qualidade para a posição MC aos 5 segundos de refinamento.

Comparando estes resultados com a imagem obtida directamente pelo radiance, verifica-se que o critério que consegue uma melhor aproximação, valorizando as sombras mais salientes, é o critério NLD\_OS. Todavia, é de notar que a penumbra das torres na muralha está melhor delineada na imagem do critério RND. Tal justifica-se pela maior frequência de amostragem obtida por este critério pelo facto de ser mais rápido. Devido às pequenas diferenças de luminância existente nos contornos da penumbra. Os critérios *NLD* demoram

mais tempo a detectarem essas características.

Comparando os tempos, verifica-se que ao fim de 5 segundos a imagem obtida pelo critério NLD\_OS já é bastante aproximada à imagem obtida pelo Radiance, a qual demorou 55 segundos a ser obtida (11 vezes mais rápido).

### 5.2.2 Vista de rua

Nesta posição do observador, o modelo de base possui 602 triângulos visíveis. Analisando a figura 5.5 verifica-se que ao fim de 5 segundos a imagem correspondente ao critério NLD\_OS é melhor que a correspondente ao critério RND no que diz à sombra dos telhados, ao longo da rua, e embora a sombra da primeira coluna à esquerda esteja só em parte detectada, o pouco que tem apresenta melhor qualidade no rigor das arestas. A sombra da segunda coluna do mesmo lado aparece completa e bem delineada. Comparando relativamente a estes dois critérios os dados da tabela 5.3 é clara a diferença em termos de recursos consumidos. O critério NLD\_OS apresenta um número de triângulos, quer efectivos quer transferidos, quase 4 vezes menor que o critério RND.

Tempos (s)		0	1	2	3	4	5
NLD_OS	$\Delta_{Vis}$	602	1 923	2 993	3 982	7 791	8 605
	$\Delta_{Ins}$	602	2 652	5 032	6 221	11 229	12 863
	$\Delta_{Rem}$	0	729	2 039	2 239	3 438	4 258
NLD_IS	$\Delta_{Vis}$	602	3 997	4 217	5 684	12 087	13 255
	$\Delta_{Ins}$	602	5 512	5 512	8 090	17 595	18 656
	$\Delta_{Rem}$	0	1 515	1 295	2 406	5 508	6 200
RND	$\Delta_{Vis}$	602	4 721	13 997	21 941	22 958	31 442
	$\Delta_{Ins}$	602	6 162	21 650	34 671	35 974	46 395
	$\Delta_{Rem}$	0	1 441	7 653	12 730	13 016	14 953

Tabela 5.3: Número de triângulos no visualizador para a posição VR.

O facto de poderem haver algumas dúvidas sobre qual o melhor dos dois resultados ao fim 5 segundos deve-se ao facto de o segundo ser bastante mais rápido que o primeiro atingindo uma maior frequência de amostragem com o mesmo tempo de execução. Na figura 5.6 são apresentadas duas imagens correspondentes à aplicação dos três critérios durante 20



(a) Radiance



(b) Q1: NLD\_OS



(c) Q2: RND



(d) Q3: NLD\_IS

Figura 5.5: *Comparação de qualidade para a posição VR aos 5 segundos de refinamento.*

segundos. Comparando-as, pode-se claramente concluir que a imagem correspondente ao critério NLD\_OS apresenta melhor qualidade.

Quanto ao critério NLD\_IS, é claramente o que dá piores resultados apesar da maior frequência de amostragem relativamente ao critério NLD\_OS. Todavia, ao fim de cerca de 20 segundos, o critério NLD\_IS pode ultrapassar em qualidade o critério RND. Tal facto justifica-se pelo facto de este critério ser bastante mais lento que o RND mas é, todavia, determinístico na selecção de pontos para refinamento o que faz com que, com o passar do tempo, vá convergindo mais rapidamente para a solução de referência pelo facto de os quadrantes irem progressivamente diminuindo de tamanho, enquanto que o critério RND vai tendo cada vez menor impacto na melhoria da imagem devido à sua aleatoriedade. É pelo facto de ser um critério aleatório que a imagem (c) da figura 5.5, sendo obtida ao fim de 5 segundos, pode ser considerada com melhor qualidade, que a imagem (c) da figura 5.6, obtida após 20 segundos.



(a) NLD\_OS



(b) NLD\_IS



(c) RND

Figura 5.6: *Critérios para a vista VR após cerca de 20s.*

Embora o critério NLD\_OS possa ser considerado o que produz melhores resultados, é evidente a existência de zonas escuras (ou mesmo pretas) na imagem correspondente a este critério, por exemplo, na parede do lado esquerdo. Tal facto deve-se a que estando os vértices dos triângulos correspondentes a essas áreas ocultos, a iluminação nos mesmos é nula. Este é um dos problemas que o modelo Bracara Augusta apresenta (ver secção 2.4.2).

Nesta vista, o Radiance demorou 40 segundos a gerar a imagem completa, o que é o dobro dos 20 segundos da melhor imagem obtida com o critério NLD\_OS. Todavia, comparando com a imagem obtida ao fim de 5 segundos, praticamente só se nota a falta da sombra da primeira coluna à esquerda e a mancha negra na parede do mesmo lado que faz com que as duas janelas existentes nessa zona desapareçam completamente.

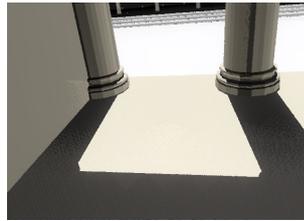
### 5.2.3 Vista de templo

Nesta posição do observador, o modelo de base possui 737 triângulos visíveis. Analisando a figura 5.7 verifica-se que ao fim de 5 segundos uma ordenação clara da qualidade resultante da aplicação de cada um dos critérios. Grosso modo, os três critérios dão resultados parecidos, embora ao nível do detalhe, a delineação das arestas das sombras é muito boa no critério NLD\_OS, sendo essencialmente neste aspecto que as três imagens se distinguem.

Ao fim de 2 segundos, o critério NLD\_OS já tinha definido muito rigorosamente a sombra. Ao fim dos 5 segundos verifica-se que este critério conseguiu melhores resultados com metade dos triângulos do critério NLD\_IS e cinco vezes menos que o critério RND.

Tempos (s)		0	1	2	3	4	5
NLD_OS	$\Delta_{Vis}$	737	1 472	7 075	7 332	7 402	7 830
	$\Delta_{Ins}$	737	1 836	11 098	11 579	11 707	12 347
	$\Delta_{Rem}$	0	364	4 023	4 247	4 305	4 517
NLD_IS	$\Delta_{Vis}$	737	3 142	7 484	8 791	11 383	16 058
	$\Delta_{Ins}$	737	4 286	10 758	12 696	16 417	23 431
	$\Delta_{Rem}$	0	1 144	3 274	3 905	5 034	7 373
RND	$\Delta_{Vis}$	737	13 336	23 911	24 835	33 638	35 433
	$\Delta_{Ins}$	737	19 404	32 449	38 124	48 333	54 046
	$\Delta_{Rem}$	0	6 068	8 538	13 289	14 695	18 613

Tabela 5.4: Número de triângulos no visualizador para a posição VT.



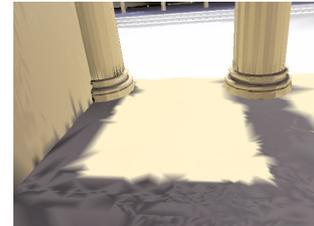
(a) Radiance



(b) Q1: NLD\_OS



(c) Q2: RND



(d) Q3: NLD\_IS

Figura 5.7: Comparação de qualidade para a posição VT aos 5 segundos de refinamento.

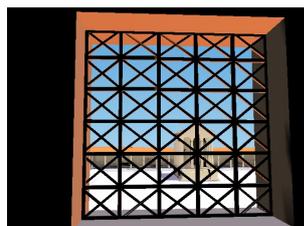
Nesta vista, o Radiance demorou 39 segundos a gerar a imagem completa, o que é quase 20 vezes superior aos 2 segundos necessários ao critério NLD\_OS para obter uma boa aproximação (dos 2 aos 5 segundos não houve grandes alterações na imagem como pode ser observado na figura B.3).

### 5.3 Desempenho do critério NLD\_OS

A partir dos dados apresentados na secção anterior pode concluir-se que o critério NLD\_OS apresenta características interessantes que o tornam o mais eficiente dos três critérios apresentados. Verificou-se que este é bastante selectivo na procura das zonas de interesse para refinamento. Embora apresente algumas lacunas no que diz respeito a encontrar zonas de interesse em triângulos cujos vértices estão iluminados de igual forma, estas foram resolvidas em parte pela aplicação de algumas optimizações ao algoritmo (secção 4.3.5).



(a)



(b)



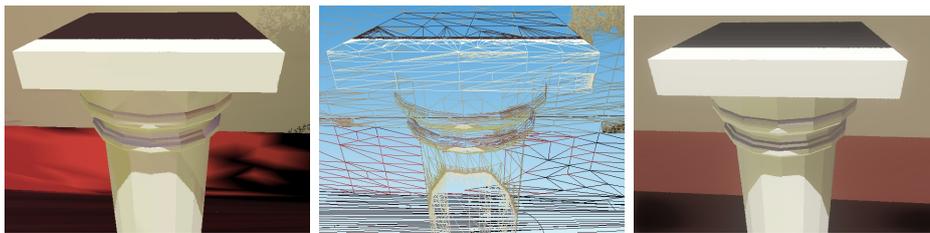
(c)

Figura 5.8: Efeitos das optimizações aplicadas ao critério NLD\_OS.

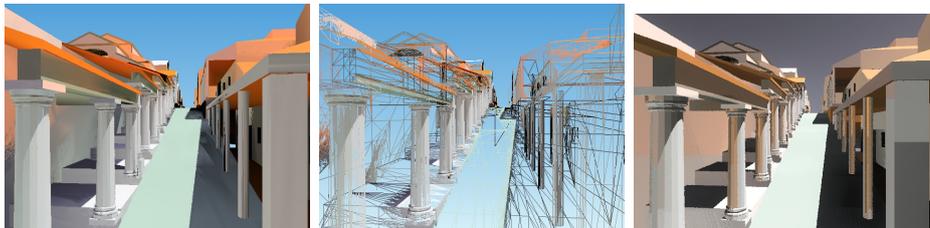
A acção das optimizações referidas pode ser encontrada em alguns exemplos sugestivos como os que se apresentam na figura 5.8. Em (a) verifica-se que a sombra do templo foi identificada com rigor embora este esteja posicionado sobre um espaço rectangular representado na geometria de base por dois triângulos de tamanho significativo. Pelo métrica *NLD* do critério NLD\_OS, nenhum destes dois triângulos seria em principio seleccionado, mas pelo facto de se projectarem num elevado número de pixels, foram seleccionados para refinamento. A partir do momento que estes foram subdivididos começaram a surgir diferenças de *shading* nos vértices dos seus filhos o que levou a que o critério começasse a actuar através da métrica que levou a uma rápida convergência para imagem de referência. No exemplo (b), a situação torna-se interessante pelo facto de o templo estar a ser observado através de uma janela. Mesmo assim, o detalhe da sombra do templo aparece claramente delineado. Tal deve-se à mesma optimização relativa ao tamanho dos triângulos. No exemplo (c) vê-se com rigor as sombras do telhado assim como da coluna. Estas sombras são rapidamente encontradas, começando o processo do mesmo modo dos anteriores; projectando-se estas em triângulos de tamanho razoável, embora estes não apresentem diferenças entre os seus vértices, ao serem seleccionados para subdivisão, os respectivos filhos começam a descobrir as sombras e a delinear os seus contornos.

A versão final do critério NLD\_OS, utilizada nos testes de que se apresentam aqui os resultados, pode com certeza ser ainda melhorada, quer por recurso a outros critérios em combinação com este, quer por melhoria do seu algoritmo após realização de um estudo mais aprofundado das potencialidades e características do mesmo. Entretanto, tendo em conta a versão actual e as potencialidades que poderam ser descobertas até ao momento, analise-se mais alguns exemplos sugestivos.

Na figura 5.9, em (a), podemos observar, à direita, os detalhes dos contornos das sombras que se destacam na coluna, ao centro, as malha de triângulos após refinamento e à direita a imagem de referência obtida pelo Radiance. Verifica-se que o critério é muito eficiente na detecção e seguimento dos limites das sombras, especialmente sombras bem definidas. Todavia, verifica-se que o critério não distingue aquilo que é essencial por estar em primeiro plano, como a coluna, e o que não é tão essencial como a parede de fundo onde este seleccionou para refinamento um conjunto significativo de triângulos (canto superior direito da imagem). O exemplo (b) é bastante completo em conteúdo e oferece uma visão clara das potencialidades que se conseguiram encontrar até ao momento no critério NLD\_OS. Por um lado, há sombras fortes resultantes das sombras dos telhados sobre a rua ascendente. Por outro, temos sombras



(a)



(b)

Figura 5.9: *Potencialidades do critério NLD\_OS.*

suaves que aparecem bem definidas na parte superior de todas as colunas cilíndricas do lado esquerdo da imagem. Todavia, repare-se que a sombra existente na primeira coluna rectangular, à direita, não foi detectada. Tal facto pode justificar-se, por um lado, pelo facto de os triângulos da geometria de base não apresentarem diferenças de iluminação entre os seus vértices, e por outro pelo facto de estes triângulos apresentarem uma dimensão inferior à definida para os triângulos a seleccionar segundo a optimização atrás referida. Este pormenor sugere que outras optimizações deverão ser implementadas de modo a evitar que uma sombra mais ou menos significativa como esta não seja detectada.

## 5.4 Desempenho do sistema

Apesar do objectivo principal deste trabalho ser critérios de refinamento, é importante apresentar alguns dados relativos à forma como o sistema completo se comporta em termos de tempo de resposta.

Como já foi referido anteriormente (secção 4.1), o tempo de resposta do sistema quando as amostras pedidas se encontram na *cache* do SMA é quase instantânea. Sendo assim, é importante ter uma ideia, por um lado, do tempo de resposta do *renderer*, por outro, do tempo necessário aos dois processos mais pesados em termos de processamento: o cálculo de visibilidade e o cálculo dos triângulos a subdividir. Neste segundo considera-se só o caso do critério NLD\_OS por ter sido considerado o mais eficiente dos três estudados.

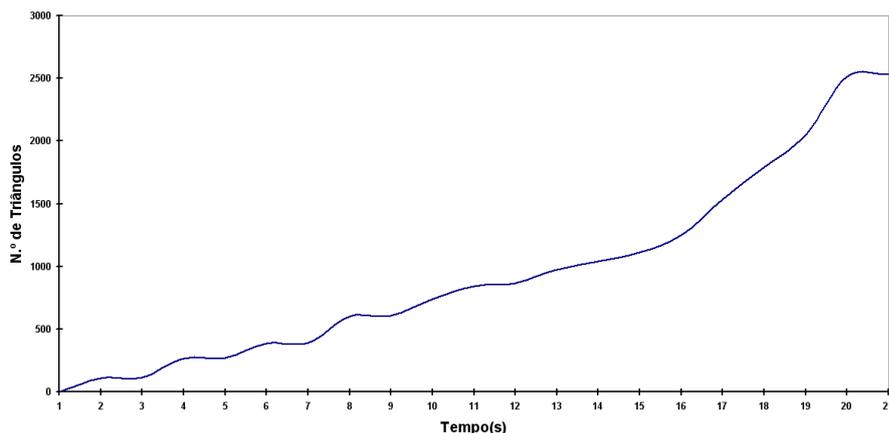


Figura 5.10: *Tempo de resposta do renderer.*

Os tempos de resposta do *renderer* apresentados, foram obtidos no visualizador e não no SMA. Tal deve-se ao princípio de que do ponto de vista do utilizador o que interessa é o que ele sente e observa na sua interacção com o sistema e não o que efectivamente o sistema faz ou como o faz. A informação obtida é apresentada na forma de um gráfico (figura 5.10) que se obteve a partir de dados recolhidos para um conjunto de diferentes cenas do modelo Bracara Augusta. O gráfico dá-nos uma estimativa bastante exacta do comportamento do sistema neste aspecto em particular. Dos dados recolhidos e apresentados no gráfico verifica-se que, em média, o *renderer* (sem paralelismo e a correr na mesma máquina que o SMA) consegue processar cerca de 1000 triângulo em cada 3 segundos, o que equivale a 1 vértice ou ponto por milissegundo. É de referir que o *renderer* está a calcular as inter-reflexões difusas (iluminação global) o que implica traçar centenas de raios.

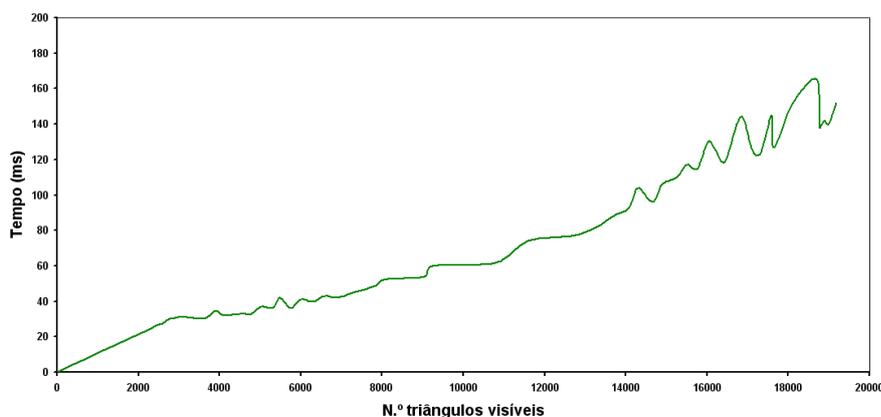


Figura 5.11: *Tempo de cálculo da visibilidade.*

Relativamente ao cálculo da visibilidade, os dados foram obtidos no SMA e são apresen-

tados também na forma de gráfico onde se apresenta o tempo gasto pelo algoritmo em função do número de triângulos visíveis. Os dados apresentados consideram o tempo correspondente aos dois níveis de cálculo de visibilidade. Todavia, o tempo associado ao primeiro nível é aproximadamente sempre igual pelo facto de se aplicar sempre a toda a geometria e é em média de cerca de 30 milissegundos<sup>1</sup>. Só o tempo associado ao segundo nível de visibilidade é variável, ou seja, aumenta em função do número de triângulos. O gráfico da figura 5.11 dá uma ideia clara da relação existente entre o número de triângulos visíveis e o tempo gasto no processamento da visibilidade. Em média é necessário um milissegundo para processar cada 137 triângulos visíveis.

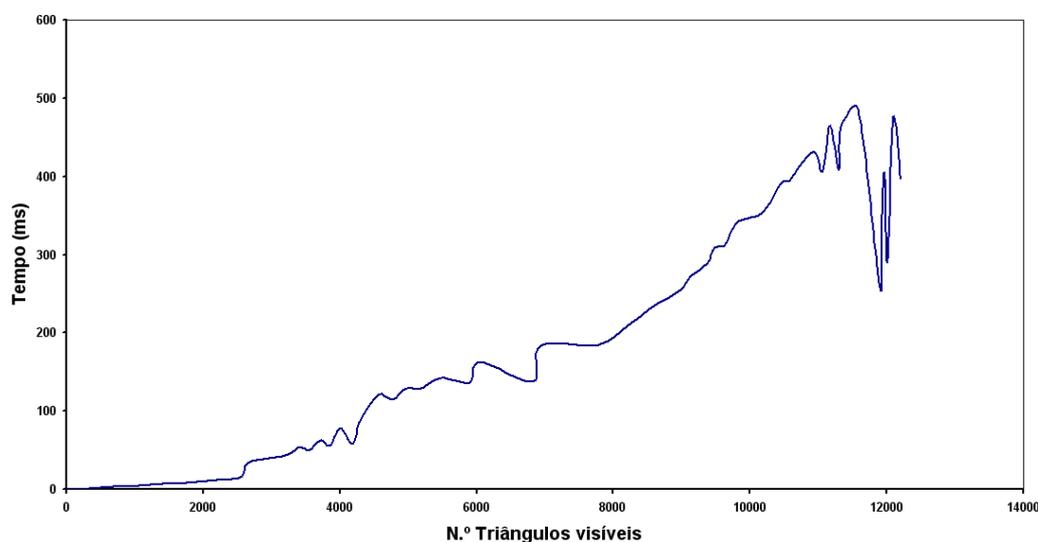


Figura 5.12: *Tempo de aplicação do critério de refinamento.*

No que diz respeito ao processo de aplicação do critério de refinamento, foi feita uma análise semelhante. Neste caso a curva apresenta algumas variações pontuais e que se devem ao facto de o tempo gasto na aplicação do critério depender das propriedades individuais dos triângulos visíveis, os quais são testados por uma determinada ordem. Como foi referido na secção 4.3.5, o processo de refinamento é interrompido após a selecção de um determinado número de triângulos e o controlo é passado ao ciclo principal do sistema. Deste modo, caso os primeiros triângulos testados sejam em grande número, o processo termina mais rapidamente. Obviamente, o algoritmo será mais rápido numa fase inicial do processo de refinamento e mais lento numa fase já adiantada pelo facto de a maioria dos triângulos já não

---

<sup>1</sup>O cálculo da visibilidade no primeiro nível é feito recorrendo ao processador gráfico e acelerando o processo pela utilização de *display lists*.

serem seleccionados para refinamento apesar de serem testados. Por observação do gráfico da figura 5.12, verifica-se que o aumento do tempo de processamento é ligeiramente acentuado, sofrendo algumas oscilações súbitas. Para os dados recolhidos, nos quais se baseia o gráfico apresentado, em média é gasto 1 milisegundo para o processamento de cada conjunto de 48 triângulos.



# Capítulo 6

## Conclusão e trabalho futuro

Três critérios de refinamento diferentes, usados para seleccionar quais os triângulos a subdividir numa abordagem de subamostragem do *rendering*, foram analisados: um critério aleatório (RND), um critério no espaço da imagem (NLD\_IS) e um critério no espaço dos objectos (NLD\_OS). Os dois últimos são baseados em diferenças de luminância normalizadas (NLD). Os resultados foram obtidos utilizando um modelo da antiga cidade romana, Bracara Augusta.

A análise destes resultados mostrou que o critério do espaço dos objectos identifica claramente descontinuidades visuais, tais como sombras, e adapta a malha de triângulos do modelo aos seus contornos. As outras duas abordagens geram resultados visualmente pouco agradáveis, com contornos mal definidos, requerendo, contudo, mais subdivisões. O critério NLD\_OS obtém melhores resultados mais rapidamente que o critério NLD\_IS, mas é mais lento que o critério RND. O tempo de execução extra é, contudo, compensado pela melhor qualidade da imagem resultante. A optimização do critério NLD\_OS é provavelmente uma boa opção tendo em vista a redução dos tempos na selecção de triângulos.

Em face de algumas limitações no modelo (secção 2.4.2), as opções em termos de critérios de refinamento ficaram de alguma maneira limitadas, particularmente, a questão da ordem dos vértices dos triângulos, afectou de algum modo a tomada de decisões. É de referir que a técnica de combinação de arestas e pontos [BWG02, BWG03] adaptar-se-ia particularmente bem a este projecto, não no seu todo, mas na parte que diz respeito à detecção de arestas. Com a informação relativa às arestas, o processo de refinamento poderia ser bastante mais objectivo, acelerando o processo de refinamento implementado pelo critério NLD\_OS, aqui proposto.

Os critérios de refinamento baseados na percepção, pelo facto de serem muito exigentes

em termos de tempo de processamento, não se adequam particularmente a este projecto, embora se possa antever algumas possibilidades de aplicação deste tipo de critérios, tendo em conta as características das propostas apresentadas por Farrugia *et al.* [FP04] e por Longhurst *et al.* [LDC05] que conseguiram melhorar o tempo de resposta de critérios baseados em percepção. A utilização de critérios baseados em percepção, seja de forma isolada, seja em combinação com os critérios aqui propostos, permitiria fazer com que se evitassem algumas das situações referidas durante a análise de resultados (capítulo 5). Tendo em conta os resultados obtidos com o critério MLD\_OS, a possibilidade de orientar a sua aplicação a zonas mais importantes em termos sensoriais poderia melhorar significativamente a resposta do sistema, quer em termos de tempo, quer em termos de qualidade.

O primeiro passo a dar com vista a melhorar o sistema é concerteza resolver os problemas existentes no modelo, incluindo a correcta definição dos materiais e texturas de modo a poder habilitar o sistema a suportar fenómenos especulares, polígonos texturizados e ainda objectos em movimento. Com a introdução de fenómenos especulares a radiância reflectida passa a ser dependente da posição do observador; as amostras em cache correspondentes a materiais especulares devem, portanto, ser actualizadas frequentemente. Para isso, é importante ter em conta a idade que neste momento é incrementada a cada novo vp. De futuro o envelhecimento deve ter em conta as propriedades dos materiais, devendo estas serem carregadas a partir da geometria de base.

Entretanto, visto que as melhorias atrás referidas dependem do modelo, será oportuno implementar o processo de remoção de triângulos de subdivisão. Numa navegação demorada, o número de triângulos poderá crescer imenso, logo é do maior interesse implementar formas de remover triângulos com base em algum tipo de critérios como a idade, a probabilidade de este ser necessário nas próximas *frames*, ou segundo critérios geométricos tais como a dimensão ou forma dos triângulos. No processo de subdivisão pode-se ter casos em que a subdivisão não traz nada de novo. Por exemplo, uma superfície plana, não especular, que não contenha qualquer tipo de descontinuidades, não precisa de ser subdividida, mas tal só é possível saber após fazer a subdivisão, daí a importância de anular tais casos de subdivisão desnecessária. Tal pode ser conseguido pela implementação de um método de optimização da malha que tenha em conta o facto de que tendo triângulos vizinhos cujos vértices possuem o mesmo valor de *shading* (ou muito próximo) não é necessário ter todos esses triângulos. Podemos reduzir a malha, diminuindo conseqüentemente o tamanho da *cache*. Esta redução do tamanho da malha pode ser feita, quer com base na uniformidade dos vizinhos, quer pelos

critérios acima referidos.

Outro aspecto a melhorar em trabalho futuro tem a ver com a possível aplicação de filtros no visualizador de modo a eliminar pequenos artefactos que surgem, quer por existência de valores de *shading* menos exactos em certos pontos, quer pela pequena dimensão de alguns triângulos que possam estar a ser exibidos. A implementação de tais filtros só poderá ser materializada se estes forem implementados directamente na placa gráfica. De outro modo afectariam significativamente o desempenho do sistema.

Em termos de trabalho futuro, é ainda de referir a intenção de melhorar ainda mais o algoritmo de mapeamento de tons utilizado com vista a exibir imagens perceptualmente mais agradáveis.

Um dos principais objectivos, relativamente a este projecto, é adaptar o visualizador para funcionamento em dispositivos móveis e explorar os protocolos de comunicação sem fios.



# Apêndice A

## Construção da métrica NLD

A métrica proposta neste documento foi baseada na fórmula de cálculo do desvio padrão para uma população finita de amostras.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}} \quad (\text{A.1})$$

Como o que interessa é a variação dos valores entre as diferentes amostras e não a variação destas em relação à respectiva média, a equação fica:

$$\sigma' = \sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N (x_i - x_j)^2}{N}} \quad (\text{A.2})$$

Em seguida, considerando que há um limite superior ( $x_{max}$ ) para os valores das amostras, pode-se normalizar o valor obtido recorrendo a este valor máximo. Sabemos, da equação A.2, que o valor máximo de  $\sigma'$  é obtido quando temos, para um total de  $N$  amostras, metade deste número (ou metade menos 1) com o valor mínimo possível e as restantes amostras com o valor máximo. Sendo assim, considerando que o valor mínimo é 0 (zero), temos

$$\sigma'_{max} = \sqrt{\frac{[(N \text{ div } 2) \times (N - (N \text{ div } 2))]x_{max}^2}{N}} = \sqrt{\frac{(N^2 \text{ div } 4)x_{max}^2}{N}} \quad (\text{A.3})$$

O valor normalizado será dado por

$$\sigma'_{norm} = \frac{\sigma'}{\sigma'_{max}} \quad (\text{A.4})$$

donde temos a equação geral

$$\sigma'_{norm} = \frac{\sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N (x_i - x_j)^2}{N}}}{\sqrt{\frac{(N^2 \text{ div } 4)x_{max}^2}{N}}} = \sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N (x_i - x_j)^2}{(N^2 \text{ div } 4)x_{max}^2}} \quad (\text{A.5})$$

Substituindo na equação  $x_i$  por  $L(x_i)$  e  $x_{max}$  por  $L_{white}$  obtemos a equação 4.3.

Notar que o operador  $\text{div}$ , que aparece nas equações A.3 e A.5, representa a divisão inteira. Por exemplo, para  $N = 3$  temos  $N^2 \text{ div } 4 = 2$ .

A métrica NLD (*Normalized Luminance Differences*) é aplicável a qualquer número de amostras  $N$ . Embora no âmbito deste documento só tenha sido utilizada para  $N = 3$  e  $N = 4$ , facilmente se visualizam aplicações para outros valores de  $N$  como por exemplo, para uma população de amostras escolhidas aleatoriamente numa determinada região (por exemplo, um quadrante) duma imagem.

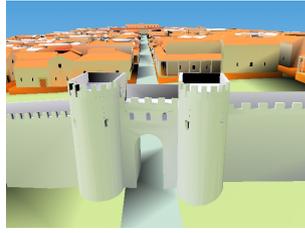
# Apêndice B

## Imagens do processo de refinamento

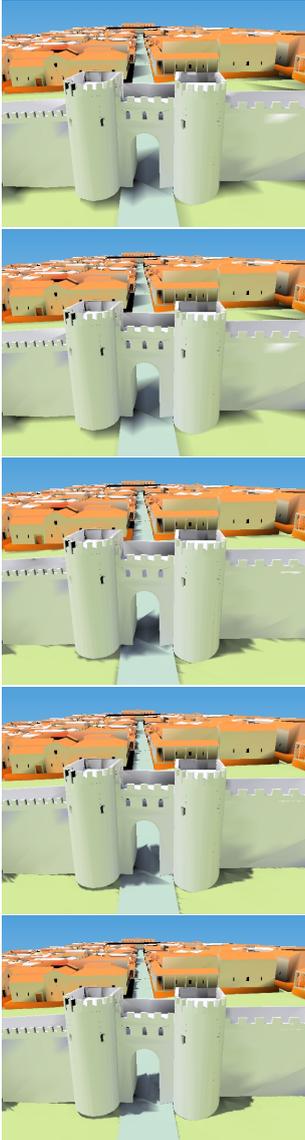
Nesta secção apresenta-se, para cada uma das vistas consideradas na secção 5.2, na análise dos resultados, um conjunto de imagens ilustrativas do processo de refinamento para cada um dos critérios.

Para cada uma das vistas apresenta-se uma figura constituída pelas seguintes imagens:

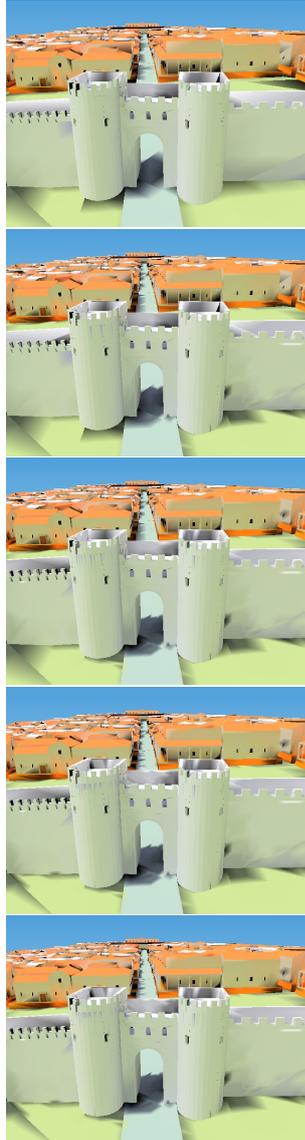
- (a) na primeira linha a imagem sem refinamento;
- (b) nas 5 linhas seguintes, primeira coluna, apresentam-se as imagens correspondentes ao evoluir do refinamento para os instantes 1, 2, 3, 4 e 5 segundos após o início do processo relativo ao critério `NLD_OS`;
- (c) nas 5 linhas seguintes, segunda coluna, apresentam-se as imagens correspondentes ao evoluir do refinamento para os instantes 1, 2, 3, 4 e 5 segundos após o início do processo, relativamente ao critério `NLD_IS`;
- (d) nas 5 linhas seguintes, terceira coluna, apresentam-se as imagens correspondentes ao evoluir do refinamento para os instantes 1, 2, 3, 4 e 5 segundos após o início do processo, relativamente ao critério `RND`;
- (e) por último, na sétima linha da imagem apresenta-se a imagem obtida directamente pelo Radiance, utilizando as mesmas configurações.



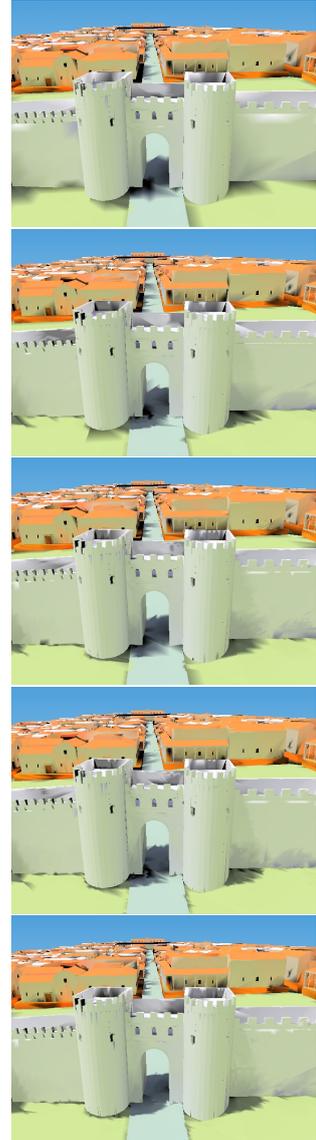
(a) Sem Refinamento



(b) NLD\_OS



(c) NLD\_IS



(d) RND



(e) Radiance

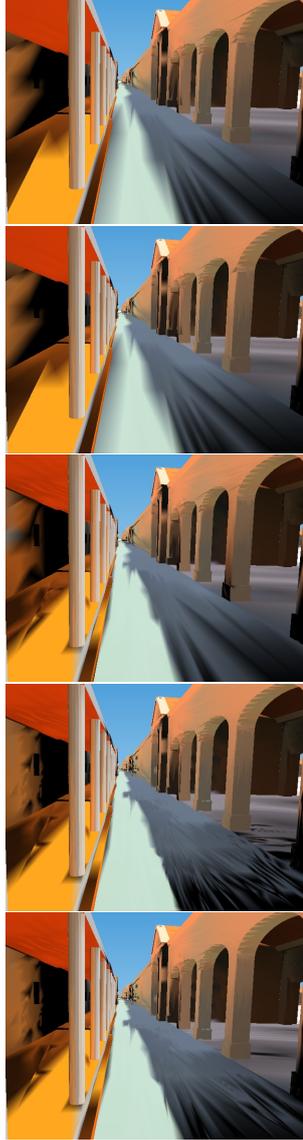
Figura B.1: Refinamento para a posição MC.



(a) Sem Refinamento



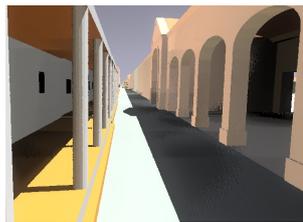
(b) NLD\_OS



(c) NLD\_IS

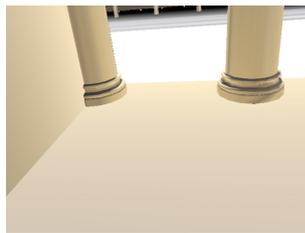


(d) RND

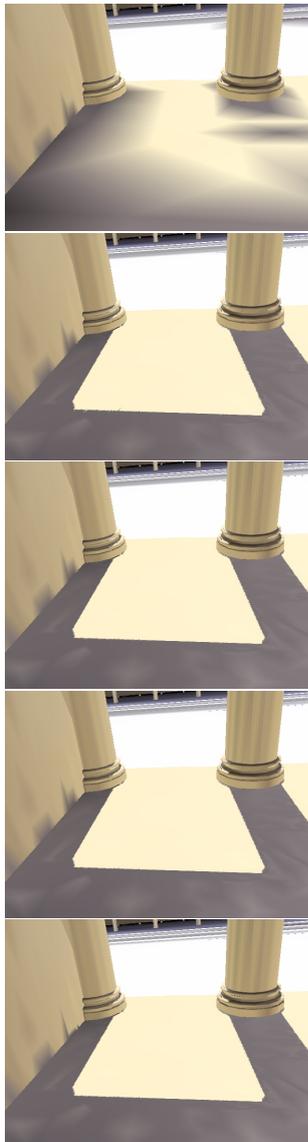


(e) Radiance

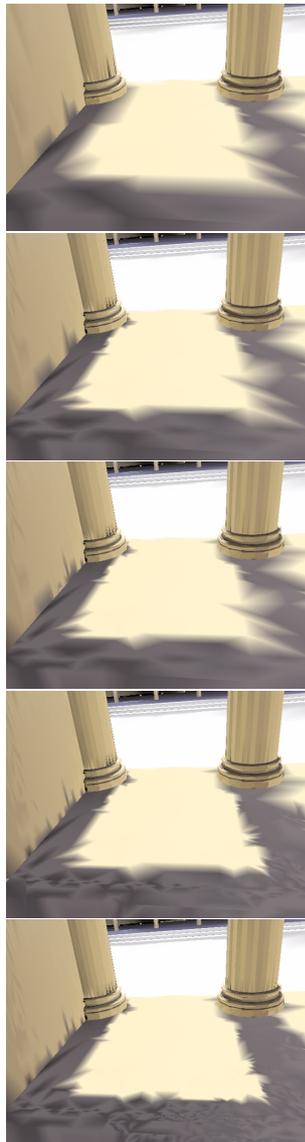
Figura B.2: *Refinamento para a posição VR.*



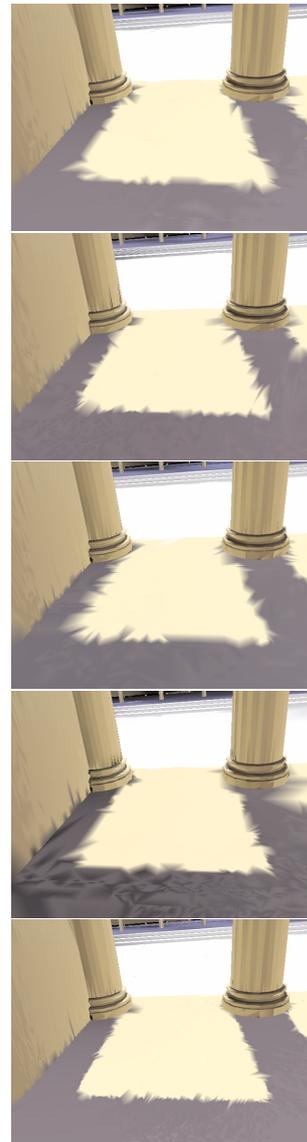
(a) Sem Refinamento



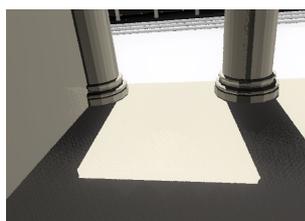
(b) NLD\_OS



(c) NLD\_IS



(d) RND



(e) Radiance

Figura B.3: Refinamento para a posição VT.

# Bibliografia

- [ARPT02] Stéphane Albin, Gilles Rougeron, Bernard Péroche, and Alain Trémeau, *Quality image metrics for synthetic images based on perceptual color differences.*, IEEE Transactions on Image Processing **11** (2002), no. 9, 961–971.
- [BDT99] Kavita Bala, Julie Dorsey, and Seth Teller, *Radiance interpolants for accelerated bounded-error ray tracing*, ACM Trans. Graph. **18** (1999), no. 3, 213–256.
- [Ber02] Paulo Bernardes, *Arqueologia urbana e ambientes virtuais: Um sistema para bracara augusta*, Master’s thesis, Instituto de Ciências Sociais, Universidade do Minho, Janeiro 2002.
- [BWG02] Kavita Bala, Bruce Walter, and Donald P. Greenberg, *Combining edges and points for interactive anti-aliased rendering*, Technical report PCG-02-3, Program of Computer Graphics, Cornell University, Jan 2002.
- [BWG03] ———, *Combining edges and points for interactive high-quality rendering*, ACM Trans. Graph. **22** (2003), no. 3, 631–640.
- [Coe05] Vítor Coelho, *Desenvolvimento de uma infraestrutura robusta e eficiente para navegação interactiva em ambientes virtuais complexos*, Master’s thesis, Departamento de Informática, Universidade do Minho, Março 2005.
- [Del34] Boris Delaunay, *Sur la sphère vide*, Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk **7** (1934), 793–800.
- [DMAC03] Frederic Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba, *Adaptive logarithmic mapping for displaying high contrast scenes*, Proc. of EUROGRAPHICS 2003 (Granada, Spain) (Pere Brunet and Dieter W. Fellner, eds.), vol. 22, Computer Graphics Forum, no. 3, Blackwell, 2003, pp. 419–426.

- [FP04] Jean-Philippe Farrugia and Bernard Péroche, *A progressive rendering algorithm using an adaptive perceptually based image metric*, Computer Graphics Forum, Volume 23, Issue 3 (Aire-la-Ville, Switzerland, Switzerland), Eurographics Association, September 2004, pp. 605–614.
- [IKN98] Laurent Itti, Christof Koch, and Ernst Niebur, *A model of saliency-based visual attention for rapid scene analysis*, IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998), no. 11, 1254–1259.
- [LDC05] Peter Longhurst, Kurt Debattista, and Alan Chalmers, *Snapshot: A rapid technique for driving a selective global illumination renderer*, WSCG 2005, February 2005.
- [LDC06] ———, *A gpu based saliency map for high-fidelity selective rendering*, Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (New York, NY, USA), ACM Press, 2006, pp. 21–29.
- [Loo87] C. T. Loop, *Smooth subdivision surfaces based on triangles*, Master's thesis, Department of Mathematics, University of Utah, August 1987.
- [Lub95] Jeffrey Lubin, *A visual discrimination model for imaging system design and evaluation*, Vision Models for Target Detection and Recognition, World Scientific, 1995, pp. 245–283.
- [PFFG98] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg, *A multiscale model of adaptation and spatial vision for realistic image display*, SIGGRAPH 98 Conference Proceedings (Michael F. Cohen, ed.), Annual Conference Series, ACM SIGGRAPH, Addison Wesley, July 1998, pp. 287–298.
- [RSSF02] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda, *Photographic tone reproduction for digital images*, SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM Press, 2002, pp. 267–276.
- [SCBP05] Luís Paulo Santos, Vítor Coelho, Paulo Bernardes, and Alberto Proença, *High fidelity walkthroughs in archaeology sites*, VAST'2005 - 6th Int. Symposium on

- Virtual Reality, Archaeology and Cultural Heritage (short paper) (Aire-la-Ville, Switzerland, Switzerland), Eurographics Association, 2005, pp. 9–11.
- [SS00] Maryann Simmons and Carlo H. Séquin, *Tapestry: A dynamic mesh-based display representation for interactive rendering*, Proceedings of the Eurographics Workshop on Rendering Techniques 2000 (London, UK), Springer-Verlag, June 2000, pp. 329–340.
- [THG99] Jack Tumblin, Jessica K. Hodgins, and Brian K. Guenter, *Two methods for display of high contrast images*, ACM Trans. Graph. **18** (1999), no. 1, 56–94.
- [TPWG02] Parag Tole, Fabio Pellacini, Bruce Walter, and Donald P. Greenberg, *Interactive global illumination in dynamic scenes*, SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM Press, 2002, pp. 537–546.
- [War94] Gregory Ward, *The radiance lighting simulation and rendering system*, SIGGRAPH '94: 21st International Conference on Computer Graphics and Interactive Techniques (New York, NY, USA), ACM Press, 1994, pp. 459–472.
- [WDG02] Bruce Walter, George Drettakis, and Donald P. Greenberg, *Enhancing and optimizing the render cache*, EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering (Aire-la-Ville, Switzerland, Switzerland), Eurographics Association, 2002, pp. 37–42.
- [WDP99] Bruce Walter, George Drettakis, and Steven Parker, *Interactive rendering using the render cache*, Tenth Eurographics Workshop on Rendering (1999), 19–30.
- [WS98] Gregory Ward and R. Shakespeare, *Rendering with radiance: the art and science of lighting visualization*, Morgan Kaufmann, September 1998.
- [YPG01] Hector Yee, Sumanita Pattanaik, and Donald P. Greenberg, *Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments*, ACM Trans. Graph. **20** (2001), no. 1, 39–65.