

## 7 Conclusão

Nessa dissertação são apresentadas e estudadas duas técnicas de renderização em pele humana em tempo real. Esses algoritmos são implementados e estruturados de forma a compor um módulo do motor de jogos *Mob3D* do Laboratório de Visualização, TV/Cinema Digital, Jogos e Efeitos Especiais da PUC - Vision Lab. Detalhes de cada uma dessas técnicas são apresentados, assim como um grupo de alterações e mesclas entre elas. Esse trabalho não trata apenas da reprodução dos algoritmos já publicados. Mesclas e alterações são propostas e implementadas, porque nem todos os detalhes e informações necessárias são disponibilizados pelos autores em seus trabalhos. Ademais, determinados parâmetros requerem tratamentos especiais. Encontrar constantes adequadas para o modelo e para algumas situações específicas, por exemplo, demanda vasta experimentação durante o desenvolvimento dos códigos. Neste contexto, a dissertação apresenta contribuições que a tornam mais do que uma mera compreensão e implementação de técnicas já divulgadas pela literatura.

O capítulo 6 apresenta os resultados obtidos com o uso de ambas as técnicas, tanto no aspecto de qualidade quanto no aspecto de desempenho. Nesses dois aspectos, na maioria dos cenários envolvidos, os algoritmos se portam e se apresentam de forma satisfatória. Além disso, é apresentado um conjunto de situações nas quais é mais adequado o uso de uma ou outra técnica. Essas duas opções permitem ao desenvolvedor maior flexibilidade durante a criação de seu jogo, pois o mesmo pode preferir focar mais na qualidade dos resultados exibidos ou reduzir um pouco a qualidade para ter uma aplicação com melhor desempenho. A opção pela redução de qualidade ocorre onde o foco do jogo não se encontra na perfeição da imagem e sim na dinâmica (e.g. jogos de esporte com velocidade e vários agentes, como no futebol). Tais opções podem e devem ser exploradas pelos programadores de jogos que utilizam o módulo.

As técnicas estudadas no capítulo 4 estão desenvolvidas com a maioria das idéias e etapas propostas por seus autores. Algumas dessas etapas não utilizadas

ainda podem ser incorporadas ao módulo desenvolvido de forma a melhorar o processo como um todo. A rugosidade  $m$  da pele é considerada nessa dissertação como uma constante em toda face. Através de experimentos realizados com base em várias pessoas de amostra (Weyrich et al. 2006 citado por d'Eon 2007b), é possível verificar que esse valor varia ao redor do rosto de uma pessoa. Caso queira-se aumentar o realismo de ambas as técnicas, pode-se usar uma textura (*Specular Texture*, d'Eon e Luebke (2007)) para obter esse valor para cada região da face. Embora esta funcionalidade seja mais fisicamente plausível, o custo computacional para aplicações em tempo real (como jogos) pode não compensar. De qualquer maneira, esta funcionalidade pode ser incorporada ao módulo, deixando a opção ao desenvolvedor (dependendo da placa gráfica, esse custo computacional pode ser irrisório).

Outro fator ainda referente à reflexão especular que pode ser incorporado ao módulo é o pré-cálculo da Distribuição de Beckmann. Essa função é complexa (equação (15)), pois envolve cálculos com exponenciais, tangentes, potências, etc. d'Eon e Luebke (2007) propõem usar a mesma idéia de criar uma textura para armazenar o resultado dessa função, como a textura de atenuação de energia, fazendo com que no momento do cálculo de reflexão especular o resultado seja apenas lido na imagem.

Conforme explicado na seção 4.1.1, várias matrizes têm que ser criadas de forma a gerar e comparar o mapa de sombra com o ponto  $p$  processado. Essas matrizes atualmente estão sendo desnecessariamente criadas vértice a vértice pelo código do módulo de renderização de pele humana. Ao chamar o módulo, o programa principal pode passar, como parâmetro, a matriz pronta para projetar um ponto no espaço da luz, aumentando assim o desempenho da aplicação. Essa necessidade de criar as matrizes vértice a vértice ocorre devido à limitação do software utilizado para o desenvolvimento, que não permite a criação de variáveis não constantes nas partes externas aos *shaders*.

Na fase atual do módulo, apesar da preocupação com o desempenho, o foco se mantém na qualidade dos resultados. Apesar do módulo ter sido bem sucedido em sua grande parte, alguns detalhes podem e devem ser ajustados nas próximas fases.

Um primeiro ponto a ser aprimorado é a quantidade de polígonos do objeto utilizado para testes. Apesar das texturas usadas serem de alta definição, a malha

possui poucos pontos, o que provoca resultados com sombras fortes e quadradas, atrapalhando também o resultado do espalhamento global para determinados conjunto de ângulos e posições da luz.

Outro ponto sugerido por d'Eon e Luebke (2007) que pode ser usado para aumentar a qualidade, é a passagem de um *bloom filter* na imagem final, gerando menos artefatos nos resultados e corrigindo os *aliasings* gerados. Além disso, outro ponto de melhoria sugerido por eles é a modelagem de fios de cabelo na face para aumentar o realismo, conforme descrito por Koenderink e Pont (2003 citado por d'Eon e Luebke (2007)).

Hable et al (2009) e Jimenez et al (2009) apresentam técnicas com base no algoritmo de d'Eon e Luebke (2007) para aumentar o desempenho final da aplicação. A primeira utiliza basicamente dois filtros de tamanhos diferentes para representar o perfil de difusão da luz na pele através de algumas amostras representativas. Consequentemente, essa técnica diminui o número de passos significativamente, porém não apresenta alguns detalhes, como o espalhamento global da luz. Já a segunda técnica utiliza um paradigma diferente, onde os *blurs* são realizados no espaço da tela. Essa mudança faz com que informações tridimensionais dos vetores não possam ser usadas posteriormente. Apesar desta limitação, a segunda técnica é útil para cenários com vários personagens, pois não é necessário realizar convoluções para cada personagem.

Outro ponto importante que pode ser inserido no módulo de forma a aumentar o desempenho é a utilização de tamanhos diferentes de textura de acordo com a distância do personagem ao ponto de visualização. Assim, por exemplo, personagens podem utilizar texturas de 1024 X 1024 durante o processamento quando próximos da câmera e personagens distantes podem utilizar texturas de 512 X 512 ou 256 X 256. Essa idéia é apresentada por Jimenez e Gutierrez (2008) e pode ser usada de forma combinada com as outras apresentadas na seção 6.3. Nesse caso, por exemplo:

- se um personagem estiver até uma distância  $d1$ , usa-se a técnica de d'Eon e Luebke (2007) sem alteração das dimensões dos mapas de irradiância;
- se ele estiver entre  $d1$  e  $d2$ , mantém-se a técnica, porém altera-se o tamanho dos mapas gerados; e

- se estiver além de  $d2$  pode-se usar a técnica de Gosselin et al. (2004a)).

Desenvolver os algoritmos citados ou mesclá-los, adaptá-los e incorporá-los ao módulo existente são possíveis melhorias ao motor de jogos, o que aumentará o desempenho e qualidade dos jogos além de permitir maior variedade de opções ao desenvolvedor.

Por fim, vale ressaltar que, assim como d'Eon e Luebke (2007) mencionam, apesar da técnica de perfil de difusão ser explicada de forma a atender os requisitos da pele humana, o conceito pode ser generalizado. Por exemplo, a técnica pode ser usada e adaptada para outros materiais com características translúcidas, utilizando, para esses, os perfis de difusão adequados.