

6 Resultados

Com base em todas as técnicas e teorias estudadas nessa dissertação mais a arquitetura e detalhes de implementação utilizados para gerar a solução do módulo de renderização da pele humana desenvolvido, nesse capítulo são apresentados e analisados os resultados obtidos. Primeiramente é feita uma análise de desempenho das técnicas, posteriormente uma análise visual e por fim uma análise geral com base na teoria e experimentação.

6.1. Análise de Desempenho

Para fazer uma análise de desempenho das técnicas estudadas no capítulo 4, são utilizadas duas máquinas com configurações diferentes e oito cenários distintos para os algoritmos de Gosselin et al. (2004a) e d'Eon e Luebke (2007). Os resultados dos testes de desempenho aparecem na Tabela 8.

- Máquina 1: Windows Vista 64 bits, Core 2 Duo 2 GHz, 2GB RAM, ATI Radeon X1950 Pro (máquina de desenvolvimento).
- Máquina 2: Windows 7 64 bits, Intel Core i7 2.8 GHz, 6GB RAM, ATI Radeon HD 5770.
- Cenário 1: uso da técnica de d'Eon e Luebke (2007) gerando texturas intermediárias de convolução de 1024 x 1024 pixels sem as otimizações dos passos 2 e 3 das Tabelas 4 e 6.
- Cenário 2: uso da técnica de d'Eon e Luebke (2007) gerando texturas intermediárias de convolução de 1024 x 1024 pixels.
- Cenário 3: uso da técnica de d'Eon e Luebke (2007) gerando texturas intermediárias de convolução de 2048 x 2048 pixels sem as otimizações dos passos 2 e 3 das Tabelas 4 e 6.

- Cenário 4: uso da técnica de d'Eon e Luebke (2007) gerando texturas intermediárias de convolução de 2048 x 2048 pixels.
- Cenário 5: uso da técnica de Gosselin et al. (2004a) gerando texturas intermediárias de *blur* com 1024 x 1024 pixels sem as otimizações dos passos 2 e 3 das Tabelas 3 e 5.
- Cenário 6: uso da técnica de Gosselin et al. (2004a) gerando texturas intermediárias de *blur* com 1024 x 1024 pixels.
- Cenário 7: uso da técnica de Gosselin et al. (2004a) gerando texturas intermediárias de *blur* com 2048 x 2048 pixels sem as otimizações dos passos 2 e 3 das Tabelas 3 e 5.
- Cenário 8: uso da técnica de Gosselin et al. (2004a) gerando texturas intermediárias de *blur* com 2048 x 2048 pixels.

Tabela 8 – Quadro comparativo com os *frames* por segundo aproximados para cada uma das máquinas e dos cenários analisados.

Cenário/Máquina	Máquina 1	Máquina 2
Cenário 1	28.7 fps	131.9 fps
Cenário 2	57.0 fps	224.9 fps
Cenário 3	7.8 fps	52.1 fps
Cenário 4	20.1 fps	105.7 fps
Cenário 5	60.0 fps	331.7 fps
Cenário 6	79.5 fps	373.2 fps
Cenário 7	18.4 fps	130.5 fps
Cenário 8	26.8 fps	156.0 fps

Todos os resultados para os cenários da Tabela 8 foram obtidos considerando os seguintes dados:

- Mapa de normal de 2048 X 2048 pixels.
- Mapa de cor de 2048 X 2048 pixels.
- *Kernel Size Map* de 2048 X 2048 pixels (para os cenários 5 a 8).
- Mapa de atenuação de energia de 512 X 512 pixels (para os cenários 1 a 4).
- *Stretch Map* de 2048 X 2048 pixels (para os cenários 1 a 4).

- Mapa de sombra e mapa de sombra translúcida modificados gerados com 2048 X 2048 pixels.
- Malha poligonal tridimensional de 1555 vértices e 2986 primitivas.

Todos os resultados de desempenho foram obtidos considerando os mesmos parâmetros e ângulo de visão (conforme a Figura 36) para os cenários da Tabela 8 e tiveram como saída uma imagem de aproximadamente 500 X 540 pixels.

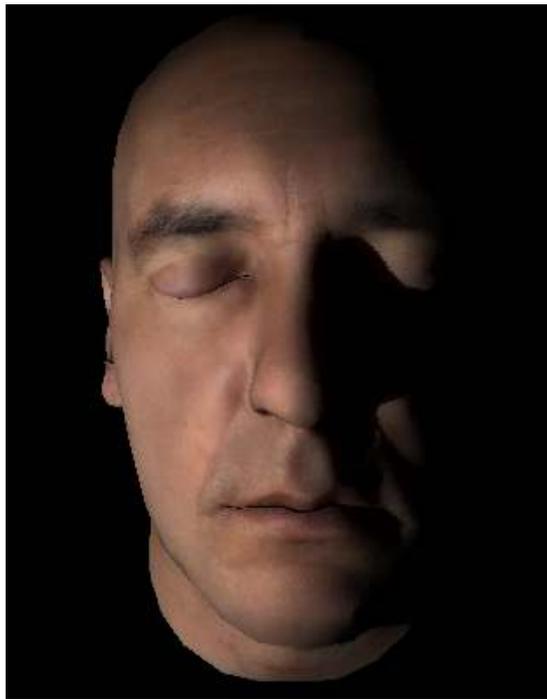


Figura 36 – Ângulo de visão considerado por todos os cenários para capturar o *framerate*.

Com base na Tabela 8 pode-se perceber que a técnica de Gosselin et al. (2004a) é mais rápida do que a técnica de d'Eon e Luebke (2007) quando ambas estão nas mesmas condições. Isso claramente deve-se a quantidade de passos onerosos (convoluções) a mais que a segunda técnica tem em relação à primeira (ver Tabelas 3 e 4).

Os resultados da Tabela 8 também mostram como o uso das técnicas de *frustum culling*, *backface culling* e corte de pixels desnecessários na imagem original (seção 5.3.1) é importante para aumentar o desempenho da aplicação, chegando a alguns casos a aumentar a *performance* em 100%.

Por fim, ambos os algoritmos de Gosselin et al. (2004a) e d'Eon e Luebke (2007) não se portam muito bem para aplicações em tempo real como jogos em máquinas com configurações antigas (máquina 1) para a resolução 2048 X 2048. Porém, para a resolução 1024 X 1024 podem ser utilizados sem grandes problemas. Para configurações mais atuais (máquina 2), ambas as técnicas apresentam bom desempenho, independente do cenário.

6.2. Análise de Qualidade

A análise de qualidade das técnicas é feita através de um conjunto de imagens comparativas entre os dois algoritmos (Figuras 37 a 41). Para a geração e comparação das imagens são considerados os cenários 4 e 8 da seção 6.1 (cenários mais detalhados). Como referência nas imagens citadas, o modelo à esquerda (a) representa a técnica de d'Eon e Luebke (2007) e o à direita (b), Gosselin et al. (2004a).

As Figuras 37 a 39 são obtidas com uma luz pontual na posição (200, 264, 50). Já as Figuras 40 e 41 são geradas com uma luz pontual na posição (150, -314, 0). Todas essas imagens são geradas com a cor de luz (234, 230, 210) no RGB.



(a)

(b)

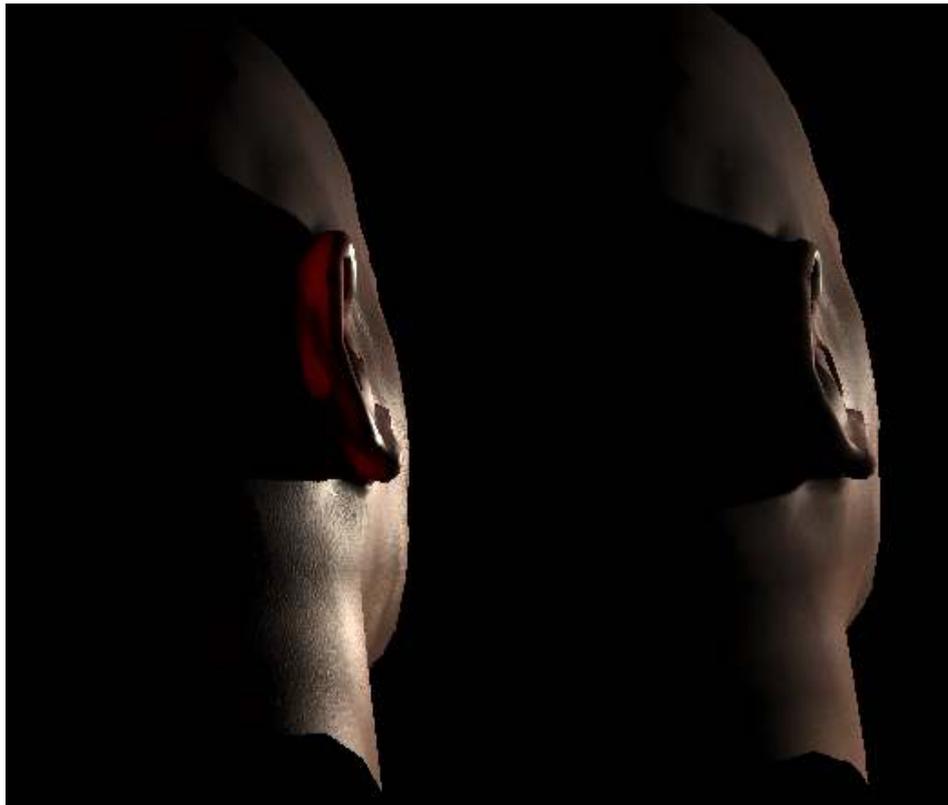
Figura 37 – Visão frontal dos dois algoritmos. A sombra no olho esquerdo do objeto está mais suave na imagem (b), porém detalhes dos poros na testa são perdidos na mesma. Além disso, a barba se encontra mais suave na imagem (a), pois a aplicação da cor do mapa de cor é aplicada antes e depois dos *blurs* (seção 4.2.5).



(a)

(b)

Figura 38 – Visão lateral dos modelos. Assim como na Figura 37, as sombras estão mais suaves com o algoritmo de Gosselin et al. (2004a). Em compensação, pode-se perceber na imagem (b) que vários detalhes na bochecha, na orelha e na junta entre orelha e cabeça são perdidos.



(a)

(b)

Figura 39 – Visão traseira das técnicas. Pode-se perceber que a imagem à esquerda (a) apresenta muito mais detalhes da pele, além de apresentar o espalhamento global da luz na orelha. Também é visualizada a reflexão especular nos modelos (embora ambas as técnicas utilizem o mesmo modelo especular, a diferença nesse aspecto se dá devido à forma final como o termo é calculado, conforme as equações (18) e (35)).



(a)

(b)

Figura 40 – Visão traseira dos modelos. Nessa situação as técnicas não apresentam muitas diferenças entre si. Pode-se perceber que os artefatos indesejados devido ao *unwrap* (seção 5.3) são corrigidos perfeitamente.



(a)

(b)

Figura 41 – Visão frontal dos modelos. Assim como na Figura 39, nessa imagem é apresentado o espalhamento global da técnica de d'Eon e Luebke (2007) e as diferenças em nível de detalhes da pele e reflexão especular de ambas as técnicas.

Com base nas Figuras 37 a 41, pode-se perceber que os resultados da técnica de d'Eon e Luebke (2007) são muito mais realistas do que os resultados obtidos com a técnica de Gosselin et. al (2004a). Isso é esperado, pois a primeira técnica é baseada em um conjunto de modelos físicos sobre a interação da luz com a pele, ao contrário da segunda técnica, que usa uma forma empírica (aplicação de vários filtros de Poisson) para simular o comportamento da luz na pele humana. A aparição de sombras mais “fortes” na técnica de d'Eon e Luebke (2007) ocorre devido à falta de detalhes da malha de entrada utilizada como exemplo.

6.3. Visão Geral

Resumindo o que é mencionado nas seções anteriores, pode-se concluir alguns pontos em relação aos algoritmos estudados anteriormente. A técnica de Gosselin et al. (2004a) apresenta desempenho melhor do que a técnica de d'Eon e Luebke (2007), porém não tem a mesma qualidade rica em detalhes que a última técnica tem. O uso de idéias e algoritmos baseados em modelos físicos permite a simulação de vários aspectos interessantes, como o espalhamento global da luz, que permite visualizar a luz atravessando regiões como a orelha. Ao mesmo tempo, esses conceitos físicos tornam esse algoritmo bem mais complexo (inclusive no entendimento) e com um número grande de passos a mais a serem executados.

Além da complexidade, o algoritmo de d'Eon e Luebke (2007) exige mais pré-computações que o algoritmo de Gosselin et al. (2004a) (textura de atenuação de energia e *Stretch Map*). Em compensação, não é necessário a intervenção de nenhum artista para criar o *Kernel Size Map*, o qual é um processo não automatizado, demorado e suscetível a inúmeros ajustes.

Enfim, ambas as técnicas podem ser utilizadas e incorporadas ao motor de jogos desenvolvido pelo Vision Lab, *Mob3D*, principalmente se utilizadas com placas gráficas e computadores modernos. Cabe ao desenvolvedor e usuário escolher as situações em que cada uma das técnicas deve ou pode ser utilizada.

Abaixo segue um conjunto de cenários propostos para cada uma das técnicas:

- Jogo com muitos personagens (exemplo: jogos de futebol): foco no desempenho, usar técnica de Gosselin et al. (2004a).
- Jogo com poucos personagens (exemplo: lutas de boxe): foco na qualidade, usar técnica de d'Eon e Luebke (2007).
- Cena do jogo onde a face de um personagem está em evidência (exemplo: personagens falando): foco na qualidade, algoritmo de d'Eon e Luebke (2007).
- Também pode ser utilizada uma solução híbrida: caso o personagem esteja há uma distância z maior do que um *delta* pré-definido, utiliza-se a técnica de Gosselin et al (2004a): personagem está longe, não precisa de tantos detalhes. Caso o personagem esteja a uma distância menor do que *delta*, utiliza-se a técnica de d'Eon e Luebke (2007). Esse cenário além de tender a atender os quesitos de desempenho e qualidade, pode ser feito de forma que o usuário não perceba a troca de uma técnica pela outra.