

Uma Nova Metodologia para Renderização Híbrida a Taxas Interativas Combinando Reflexões em Espaço de Tela com Ray Tracing em GPU

Daniel Valente de Macedo, Maria Andréia Formico Rodrigues (Orientadora)
Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Av. Washington Soares 1321, Bloco J, sala 30 – Fortaleza – CE – Brazil
{danielvalentemacedo, andreia.formico}@gmail.com

Resumo—A síntese realista de raios de luz, fundamental em aplicações gráficas (desde a indústria cinematográfica até a de jogos por computador), ainda é um processo extremamente custoso e, na maioria das vezes, inviável em tempo real. Tradicionalmente, cenas estáticas realistas têm sido geradas via *ray tracing*. Já para cenas dinâmicas, algoritmos de renderização em nível de GPU têm priorizado a velocidade em detrimento da qualidade, gerando taxas interativas de quadros/s, porém, repletas de artefatos visuais indesejáveis. Esta tese de doutorado apresenta uma metodologia inovadora para a renderização realista de cenas, tanto estáticas quanto dinâmicas, a taxas interativas, de forma híbrida. Para a geração de reflexos, combina *Screen Space Reflection* (SSR) com *ray tracing* em GPU e, para a síntese de sombras, usa *shadow maps* e/ou *ray tracing*. Testes sistemáticos realizados em cenários variados (contendo objetos planos, curvos, rígidos e deformáveis) mostraram que esta nova metodologia é capaz de gerar reflexos de alta qualidade a taxas interativas (em média, 30 quadros/s) em cenas dinâmicas, incluindo múltiplos raios recursivos.

Keywords-Reflexos dinâmicos em tempo real; Renderização realista; Cenas 3D; Raytracing

I. INTRODUÇÃO

Aplicações gráficas 3D têm estado cada vez mais presentes em áreas relevantes e de interesse estratégico na atualidade: entretenimento, anúncios publicitários, visualização e exploração de dados multidimensionais, simulação de processos em engenharia e na área médica, etc. A qualidade destas aplicações tem aumentado consideravelmente, visando resultados cada vez mais realistas, precisos e rápidos. Esta demanda crescente tem estimulado o surgimento de placas gráficas mais rápidas e processadores mais robustos, com alta capacidade de processamento paralelo. Contudo, mesmo com o avanço em termos de desempenho destes *hardwares*, as placas gráficas atuais ainda apresentam limitações importantes, tanto em termos de qualidade, quanto em termos de eficiência das imagens geradas.

Jogos por computador, filmes e visualização de grandes conjuntos de dados 3D, por exemplo, visam aspectos de renderização importantes, tais como: tempo de processamento, realismo visual, taxas interativas de quadros/s, robustez em termos de escalabilidade (número de polígonos processados ao longo do tempo e taxas de atualização

das geometrias dos objetos em movimento), resolução das imagens geradas, etc. Neste contexto, surgem as GPUs, partes essenciais das placas gráficas, as quais influenciam no desempenho da aplicação. Atualmente, devido ao alto grau de paralelismo proporcionado pela GPU e para tirar proveito de sua robustez, os custosos e tradicionais algoritmos de *ray tracing* começaram a ser explorados em GPU, com foco na renderização de cenas 3D.

Inicialmente, efeitos visuais (sombras, reflexos, luz indireta, etc.) em cenas interativas eram criados usando mapas de texturas pré-computados. Com o progresso na área de renderização e computação paralela em nível de GPU, foram produzidos os primeiros exemplos isolados de cenas 3D simples (com um número bastante limitado de objetos rígidos em movimento e de baixa complexidade geométrica), porém, sintetizadas por *ray tracing* [1]. Atualmente, a técnica de *ray tracing* não é ainda um recurso comum aplicado, por exemplo, na área de jogos digitais. De forma geral, isto decorre do seu elevado custo computacional frente ao *hardware* atual, comprometendo a experiência do usuário com taxas de quadros/s não interativas. Em especial, sombras e reflexos são efeitos relevantes, os quais agregam valor às imagens geradas e, conseqüentemente, à experiência do usuário, buscando uma aproximação com o nível de qualidade exigido pela percepção do olho humano, porém, bastante complexos de serem sintetizados em tempo-real.

Muito recentemente, poucas soluções híbridas combinando rasterização e *ray tracing* foram propostas. A que mais se relaciona a este trabalho é a de Ganestam e Doggett em GPU [2], na qual o *ray tracing* é usado para computar reflexos de objetos que estão próximos ao observador e um *cubemap* de *g-buffers* é aplicado para objetos que estão mais distantes. Há duas diferenças fundamentais entre a solução de Ganestam e Doggett e a apresentada neste trabalho: (1) nesta, reflexos são sempre fornecidos, enquanto que na proposta pelos autores acima citados os reflexos podem falhar quando um objeto (ou parte dele) sofre oclusão no *cubemap*; e (2) nos cenários de teste dos autores citados não foram abordados reflexos gerados em cenas com simulações de corpos deformáveis.

Portanto, este trabalho [3] apresenta uma metodologia inovadora para renderização híbrida de cenas a taxas interativas,

combinando reflexões em espaço de tela com *ray tracing* em GPU. Além disso, como prova de conceito, apresenta um motor de renderização híbrido, denominado *VTracer*, capaz de gerar efeitos visuais realistas (especialmente reflexos, bem como sombras), em cenas 3D estáticas e dinâmicas. Híbrido porque as imagens sintetizadas são geradas como resultado da combinação otimizada da técnica de SSR [4] com *ray tracing*.

II. HIPÓTESE E QUESTÕES DE PESQUISA

Considerando o *hardware* atual e a programação em nível de GPU, este trabalho procurou testar a seguinte hipótese:

“A combinação de uma técnica de rasterização e do algoritmo de *ray tracing*, na forma de uma solução híbrida, permite gerar efeitos visuais realistas (sombras e, particularmente, reflexos) a taxas interativas, tanto em cenas 3D estáticas contendo objetos planos e curvos, quanto em cenas 3D dinâmicas, sejam estes objetos do tipo rígido ou deformável.”

A partir desta hipótese, seis questões de pesquisa foram elaboradas e devidamente respondidas (resultando em publicações em veículos qualificados), conforme destacadas a seguir:

- 1) É possível gerar sombras e usar, de forma híbrida, SSR e *ray tracing* para gerar reflexos em cenas 3D, a taxas interativas? [5], [6], [7]
- 2) É possível gerar reflexos realistas de objetos planos e curvos, sejam estes estáticos ou dinâmicos, através de uma solução híbrida de renderização? [6], [7]
- 3) Quais são as bibliotecas em GPU existentes e estruturas de dados aceleradoras que suportariam uma solução híbrida para a síntese de reflexos em objetos dinâmicos em cenas 3D? [5], [8], [7], [9]
- 4) Quais são as principais estruturas de dados aceleradoras e suas respectivas características de atualização que podem ser otimizadas para atingir taxas interativas com o *ray tracing* em cenas 3D dinâmicas? [8], [7], [9]
- 5) É possível gerar reflexos realistas a taxas de quadros/s interativas, em cenas 3D dinâmicas contendo objetos rígidos ou deformáveis? [8], [9]
- 6) É possível avaliar a solução híbrida em diferentes cenários de teste, inclusive em simulações de dinâmica de corpos rígidos e deformáveis, quanto à sua escalabilidade, bem como quanto ao realismo visual, tempo de processamento e taxa de quadros/s das imagens resultantes? [9]

III. NOVA METODOLOGIA DE RENDERIZAÇÃO HÍBRIDA PARA REFLEXOS

A Figura 1 mostra a arquitetura e os componentes do *VTracer*, sendo dois deles principais, responsáveis pela síntese de efeitos especiais: (1) o compositor de sombras, encarregado da geração de sombras por *shadow maps* [10] ou *ray tracing*; e (2) o compositor de reflexos, responsável

pela combinação das imagens geradas pelo SSR e *ray tracer*. Mais especificamente, o compositor de reflexos é dividido em duas etapas: (1) uma que usa o algoritmo de SSR [4]; e (2) outra que usa um *ray tracer*.

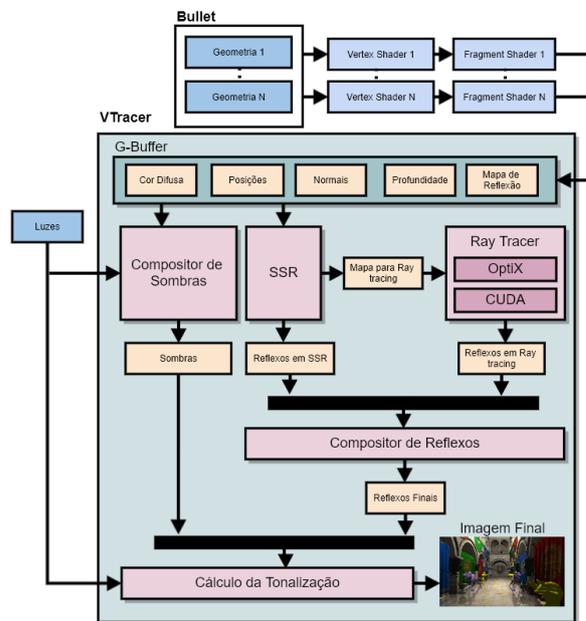


Figura 1: Diagrama da arquitetura e dos componentes do *VTracer*.

Na primeira etapa, o algoritmo recebe um mapa de reflexo do *G-Buffer* (mapa que representa quais *pixels* fazem parte dos objetos reflexivos) e usa esta informação para gerar os reflexos iniciais com SSR. Para tirar proveito dos principais pontos positivos que ambas as técnicas oferecem e contornar suas principais limitações, o algoritmo renderiza, inicialmente, os reflexos com o SSR, gerando também um mapa que representa uma máscara que contém os *pixels* que representam as áreas da cena onde o SSR falhou no cálculo dos reflexos. Este problema é bem conhecido do SSR, pois só é possível processar o que está visível na imagem. A segunda etapa do algoritmo concentra-se na geometria da cena 3D, na máscara de falhas para o *ray tracing* gerada pelo SSR e nas informações do *G-Buffer*, as quais são passadas para o *ray tracer* para calcular os raios secundários que representam os reflexos (as informações que seriam referentes aos raios primários já foram geradas pelo *G-Buffer*), para depois mesclar o reflexo gerado pelo SSR na etapa anterior com os reflexos adicionais gerados pelo *ray tracer*. O *ray tracer* verifica cada *pixel* da máscara de falhas enviada para o *ray tracing*, gerada na primeira etapa, procurando por *pixels* que não foram gerados com o sucesso esperado. Se algum *pixel* com valor 1 for encontrado na máscara (representa uma falha), um novo raio refletido a partir da posição da câmera até esse ponto é calculado e traçado, resultando em

uma nova imagem de reflexo complementar. Caso necessário, mais raios secundários são gerados, até encontrar o objeto refletido. Como resultado, os reflexos de objetos que estão fora do espaço da tela (ou que podem estar sofrendo alguma oclusão por qualquer outro objeto) também são considerados.

É importante ressaltar que, para o *ray tracer* da segunda etapa, dois módulos foram desenvolvidos. O primeiro, com o Optix da NVIDIA (usada na primeira implementação do algoritmo). O segundo, em CUDA, tendo como base uma estrutura de dados em GPU inspirada em uma *Linear Bounding Volume Hierarchy* (LBVH) [11], para acelerar a geração dos reflexos em cenas dinâmicas contendo objetos rígidos e/ou deformáveis. Para acelerar ainda mais o processo, o *ray tracer* implementado distingue entre objetos estáticos e dinâmicos, dividindo-os em duas listas distintas, desta forma, descartando a necessidade de reconstruir e atualizar toda a estrutura de dados da cena a cada quadro da animação. Por fim, as duas imagens dos reflexos são combinadas e um filtro de *Gaussian Blur* [12] é aplicado para minimizar os artefatos gerados pelo SSR, especialmente, em regiões de transição entre os reflexos.

IV. TESTES, MÉTRICAS E RESULTADOS

A nova metodologia foi aplicada em diferentes cenários. Foram realizados testes funcionais para identificação e correção de situações pontuais que pudessem gerar falhas. Testes de desempenho também foram realizados com foco no realismo das cenas sintetizadas, taxas de quadros/s e nível de escalabilidade. Para medir o nível de realismo das imagens geradas nos testes, duas métricas objetivas foram usadas (SSIM e RMSE) [13], [14], comprovando a qualidade das imagens geradas. Alguns resultados realizados com diferentes tipos de objetos refletoras são mostrados na Figura 2. Outros, produzidos através de simulações de dinâmica de corpos rígidos e deformáveis, são mostrados na Figura 3. Animações ilustrativas de vários dos resultados obtidos podem ser visualizadas em <https://goo.gl/NxkEzp>.

Os resultados obtidos confirmam que é possível combinar reflexos em espaço de tela com *ray tracing* em GPU. Adicionalmente, mostram que sombras e reflexos híbridos usando SSR e *ray tracing* podem ser gerados em cenas estáticas, a taxas bem superiores a 30 quadros/s. Com relação às sombras, testes de desempenho foram feitos com *shadow maps*, alcançando taxas de quadros/s bastante superiores e um realismo visual muito similar aos resultados do *ray tracing*.

Casos variados de testes com reflexos híbridos realistas em cenas estáticas e com objetos rígidos, em movimento constante ao longo da animação, também foram produzidos com sucesso. Adicionalmente, foi possível observar que o OptiX e suas estruturas aceleradoras não apresentaram desempenho satisfatório para este tipo de cenário. Esta evidência motivou o desenvolvimento de uma nova solução no *VTracer*, desta vez, totalmente implementada em CUDA, a qual foi usada com sucesso nos testes com cenas dinâmicas.

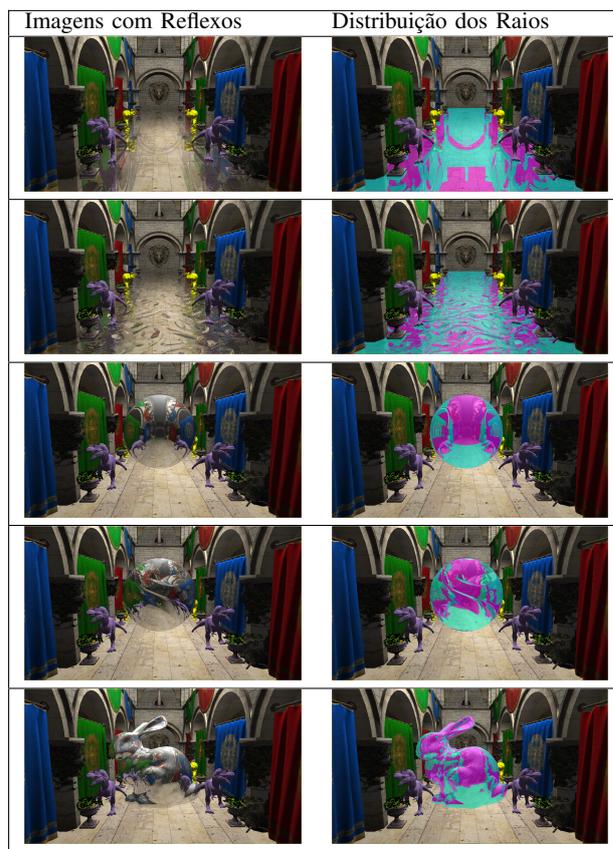


Figura 2: Imagens geradas pelo reflexo híbrido do *VTracer*. Testes com diferentes tipos de objetos refletoras, incluindo a distribuição dos raios (coloridos em rosa e azul, respectivamente, indicando os renderizados por *ray tracing* ou SSR).

Testes de escalabilidade mostraram ainda que, para uma cena com muitos objetos dinâmicos, o problema ainda é desafiador, devido ao alto custo computacional exigido para atualizar as estruturas aceleradoras do módulo *ray tracing*.

V. CONCLUSÃO

O uso de *ray tracing* em aplicações gráficas a taxas interativas está se tornando cada vez mais uma realidade, principalmente, devido aos avanços dos algoritmos, estruturas de dados aceleradoras e *hardware*. Este trabalho apresentou uma nova metodologia híbrida capaz de gerar reflexos dinâmicos realistas a taxas superiores a 30 quadros/s em cenas estáticas e em simulações dinâmicas com corpos rígidos e deformáveis (incluindo múltiplos raios recursivos). Quando comparadas a imagens geradas por uma solução completa em *ray tracing*, as imagens produzidas apresentam uma qualidade visual bastante realista, com mínimos artefatos, os quais são praticamente imperceptíveis. Possibilidades de uso desta solução são as mais diversas, tais como: *plugin* integrado a motores de jogos, ferramenta para simulação realista de reflexos a

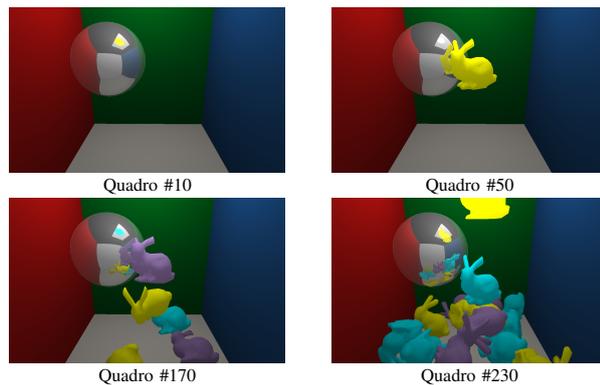


Figura 3: Seis quadros da animação de uma cena dinâmica com uma esfera reflexiva usando o módulo *ray tracer* em CUDA.

taxas interativas, módulo executando em conjunto com outras bibliotecas para compor um kit de desenvolvimento voltado para renderização realista, entre outras.

VI. PUBLICAÇÕES

Seis foram as publicações diretamente decorrentes desta tese de doutorado: [9], [7], [8], [6], [15] e [5]. Outras seis publicações colaboraram para o conhecimento técnico e auxiliaram no estudo e desenvolvimento de aplicações gráficas, contribuindo indiretamente para o trabalho: [16], [17], [18], [19], [20] e [21].

REFERÊNCIAS

- [1] S. Parker, W. Martin, P. Sloan, P. Shirley, B. Smits, and C. Hansen, “Interactive ray tracing,” in *Symposium on Interactive 3D Graphics*, ser. I3D ’99. ACM, 1999, pp. 119–126.
- [2] P. Ganestam and M. Doggett, “Real-time multiply recursive reflections and refractions using hybrid rendering,” *The Visual Computer*, vol. 31, no. 10, pp. 1395–1403, 2015.
- [3] D. V. Macedo, “Uma nova metodologia para renderização híbrida a taxas interativas combinando reflexões em espaço de tela com ray tracing em gpu,” Ph.D. dissertation, Doutorado em Informática Aplicada - Universidade de Fortaleza, 2017.
- [4] T. Sousa, N. Kasyan, and N. Schulz, “Secrets of cryengine 3 graphics technology,” *Advances in Real-Time Rendering in 3D Graphics and Games Course (SIGGRAPH)*, 2011.
- [5] D. V. Macedo and M. A. F. Rodrigues, “A Hybrid Rendering Engine Prototype for Generating Real-Time Dynamic Shadows in Computer Games,” in *XIII SBGames, Computação*, 2014, pp. 938–941. [Online]. Available: <http://tinyurl.com/jambreg>
- [6] —, “Realistic Rendering in 3D Walkthroughs with High Quality Fast Reflections,” in *XIV SBGames, Computação*, 2015, pp. 9–15. [Online]. Available: <http://tinyurl.com/zeqsl93>
- [7] —, “Real-time Dynamic Reflections for Realistic Rendering of 3D Scenes,” *The Visual Computer Journal*, vol. 34, no. 3, p. 337–346, 2016. [Online]. Available: <https://tinyurl.com/y7nh3hwk>
- [8] —, “Comparison of Acceleration Data Structures for High Quality Fast Reflections of Static and Deformable Models in Walkthrough Animations,” *SBC Journal on 3D Interactive Systems*, vol. 7, no. 1, pp. 28–37, 2016. [Online]. Available: <http://tinyurl.com/h6xhmr>
- [9] —, “Fast and Realistic Reflections using Screen Space and GPU Ray Tracing - A Case Study on Rigid and Deformable Body Simulations,” *ACM Computers in Entertainment*, vol. 16, no. 4, 2018.
- [10] R. Fernando and M. Kilgard, *The Cg Tutorial - The Definitive Guide to Programmable Real-time Graphics*. Addison-Wesley, 2003.
- [11] C. Lauterbach, M. Garland, S. Sengupta, D. Luebke, and D. Manocha, “Fast BVH construction on GPUs,” *Computer Graphics Forum*, vol. 28, no. 2, pp. 375–384, 2009.
- [12] L. Shapiro and G. Stockman, *Computer Vision*. Prentice Hall, 2001.
- [13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.
- [14] R. Hyndman and A. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, pp. 679–688, 2006.
- [15] D. V. Macedo, Y. Serpa, and M. A. F. Rodrigues, “Desenvolvimento de Aplicações Gráficas Interativas com a Unreal Engine 4,” *RITA*, vol. 22, no. 2, pp. 181–202, 2015. [Online]. Available: <http://tinyurl.com/gz4zbph>
- [16] D. V. Macedo and M. A. F. Rodrigues, “Experiências com desenvolvimento ágil de um jogo casual para plataformas móveis usando o motor gráfico Unity,” in *XI SBGames, Computação*, 2012, pp. 148–156. [Online]. Available: <http://tinyurl.com/zt4exbb>
- [17] D. V. Macedo, H. Pontes, and M. A. F. Rodrigues, “Criação de arte usando um dispositivo háptico para pintura digital interativa,” in *XV Symposium on Virtual and Augmented Reality*, 2013, pp. 288–291. [Online]. Available: <http://tinyurl.com/jhnlxv8>
- [18] M. A. F. Rodrigues, D. V. Macedo, Y. Serpa, and et al., “Combatendo a halitose: Um Serious Game multiplataforma em saúde bucal,” in *XIII SBGames, Arte e Design*, 2014, pp. 210–219. [Online]. Available: <http://tinyurl.com/h2scjnz>
- [19] —, “Um Serious Game em saúde bucal contra o tártaro,” in *XIV SBGames, Arte e Design*, 2015, pp. 699–702. [Online]. Available: <http://tinyurl.com/j8ao2k4>
- [20] —, “Beyond fun: An interactive and educational 3D traffic rules game controlled by non-traditional devices,” in *30th ACM SAC*, 2015, pp. 239–246. [Online]. Available: <http://tinyurl.com/zr3epfq>
- [21] M. A. F. Rodrigues, D. V. Macedo, H. P. Pontes, Y. R. Serpa, and Y. R. Serpa, “A serious game to improve posture and spinal health while having fun,” in *4th International Conference on Serious Games and Applications for Health (SeGAH 2016)*, 2016, pp. 1–8. [Online]. Available: <http://tinyurl.com/zxsraw4>