

Reyes - Renders Everything You Ever Saw

Hugo Henrique da Silva Lima¹, Eduardo Lima Nascimento¹

¹Faculdade de Engenharia de Computação – Universidade Federal do Pará (UFPA)
Rua Itaipú, 38 – Vila Permanente – Tucuruí – PA – Brazil

hugo.hslima@gmail.com, eduardolima384@gmail.com

Resumo. Introdução: Descreve detalhes sobre Reyes e sua aplicação na computação gráfica 3D. **Referencial Teórico:** Nessa seção é detalhada a base teórica utilizada para as análises, fazendo uma breve síntese sobre os temas abordados. **Aplicações:** Foi discutida a ferramenta Renderman um programa de renderização que foi baseado no algoritmo de reyes e como sua utilização evoluiu até o dias de hoje e como ele sintetiza o objetivo deste artigo. **Conclusões:** O estudo mostrou como Reyes revolucionou o mercado cinematográfico através da computação gráfica 3D e como a evolução dos renderizadores contribui para o melhor processamentos de imagens fotorealísticas, com qualidade e velocidade.

Abstract. Introduction: Describes your details about Reyes and its applications in 3D graphics. **Theoretical Reference:** In this section details the theoretical basis used for the analyzes, giving a brief summary on the topics covered. **Applications:** It was discussed the Renderman tool a rendering program that was based on the algorithm of kings and how its use has evolved to the present day and how it synthesizes the purpose of this article. **Conclusions:** The study showed how Reyes revolutionized the film market through 3D graphics computing and how the evolution of renderers contributes to better processing of photorealistic images with quality and speed.

1. Introdução

A computação gráfica vem crescendo desde de meados dos anos 70, muitas melhorias foram descobertas trazendo um grande avanço para área em questão. Podendo ser utilizada na arquitetura, medicina, traçando um caminho até as engenharias, vem ganhando seu brilho especialmente na indústria do entretenimento, presente no cinema, tv, games e na publicidade.

Com a possibilidade de gerar dinamicamente imagens complexas, graças principalmente aos avanços da renderização que desempenha um papel crucial na computação gráfica e é um dos aspectos mais complexos da produção 3D. Uma área que utiliza bastante os métodos de renderização e o de criação de filmes.

Aqui podemos destacar a criação do Reyes um algoritmo desenvolvido na Lucasfilm no início dos anos 80. Esse algoritmo, de alguma forma ou de outra, é usado hoje pela maioria dos renderizadores. O Algoritmo foi projetado para superar as limitações de velocidade e memória de algoritmos fotorrealistas, como traçado de raios, em uso no momento. O seu desenvolvimento foi um momento marcante com relação a imagens digitais, e trouxe diversas possibilidades ao futuro do cinema.

2. Referencial Teórico

2.1. Computação Gráfica

A Computação Gráfica é a área da computação que permite realizar a geração de imagens através de técnicas computacionais. Talvez essa definição seja muito simples para representar essa imensa área de pesquisa já que hoje essa área se dividiu em vários campos de estudos como a Realidade Virtual, Processamento de Imagens e etc [SANTOS et al. 2001]. A computação gráfica 3D é uma área que vem chamando muita atenção nos últimos anos devido o crescente avanço das tecnologias e dos softwares de manipulação de imagens, com eles é possível criar diversas imagens que pode não refletir a própria realidade. Através dessas aplicações as Indústrias cinematográficas veem aperfeiçoando várias técnicas na área de computação gráfica 3D.

De acordo com [BROOKHEAR 2013],

É importante notar que a criação de uma imagem usando computação gráfica 3D envolve dois passos distintos. Um é a criação, a codificação, o armazenamento e a manipulação da cena a ser fotografada. O outro é o processo de produzir a imagem. O primeiro é um processo artístico, criativo; o último é um processo computacionalmente intenso.

O fato de a computação gráfica 3D produzir "fotografias" de cenas virtuais a torna ideal para o uso em video games e produções de filmes de animação, nos quais as restrições da realidade de outra forma limitariam a ação. Um video game interativo consiste em um ambiente virtual tridimensional codificado com o qual o jogador interage. As imagens que o jogador vê são produzidas por meio de tecnologia de computação gráfica 3D. Filmes de animação são criados de uma maneira similar, exceto que o animador humano quem interage com o ambiente virtual no lugar do visualizador final. O produto final distribuído ao público é uma sequência de imagens bidimensionais conforme determinado pelo diretor de produção ou produtor.

Com a expansão de novas interfaces homem-máquina tridimensionais, temos a promessa de muito crescimento para o futuro da computação gráfica 3D.

2.2. Renderização

Renderização é um processo bastante utilizado na computação gráfica 2D e 3D, onde um software de renderização realiza cálculos que tenta traduzir uma determinada cena 3D, através de uma aproximação matemática a fim de que ela resulte em uma imagem 2D finalizada" [RODRIGUES 2017]. Esse processo exige muito recursos do hardware, principalmente do processador, ao transformar um ou mais arquivos em um único resultado final, e necessário definir os elementos de textura, iluminação da cena e reflexão, com isso o programa calcula a perspectiva do plano, as sombras e a luz na imagem achatada [TECNOLOGIA 2017].

Ainda podemos classificar a renderização de duas formas a de tempo real e a offline. A diferença entre as duas é dada pela velocidade que o processo é realizado no computador. A renderização em tempo real geralmente é utilizada na criação de jogos, já que o usuário interage de diversas formas no ambiente do jogo sem que tenhamos controle das ações, é necessário que as imagens sejam apresentadas de forma quase que instantânea, renderizando as imagens em tempo real [TECNOLOGIA 2017]. Com a utilização de hardwares gráficos dedicados (GPUs) e pré-compilando o máximo de informações possível temos uma melhoria na renderização. Muita informação de iluminação de um ambiente num jogo é pré-computada assim facilita e melhora a velocidade de renderização [RODRIGUES 2017]. Na renderização off-line é usada quando a velocidade não é um problema, onde os cálculos normalmente são executados em CPUs multi-core em vez de hardwares gráficos dedicados. A renderização off-line é vista com maior frequência em trabalhos de animação, efeitos, plantas de arquitetura onde a complexidade visual é mais alta e exigem mais do conjunto hardware e software [RODRIGUES 2017].

A renderização ainda possui técnicas computacionais que possui vantagens e desvantagens de acordo com o trabalho que deseja ser realizado. O processo de renderizar sempre exigiu uma maior demanda da capacidade computacional.

2.2.1. Rasterização

A Rasterização ou Scanline é uma técnica muito utilizada na renderização em tempo real, pelo fato dessa técnica prioriza a velocidade. Segundo [SANTOS 2012] podemos afirmar.

O termo vem do latim *rastrum*, relacionado ao ato de realizar um rastro ou esboço de uma representação 3D em um plano 2D utilizando apenas linhas e pontos. O objeto 3D é “escaneado” ou rastreado, de forma a encontrar quais são os pontos ou pixels da imagem que representam o objeto. Como o algoritmo é capaz de gerar apenas linhas e pontos, todo objeto 3D é representado de forma poligonal, sendo este um requisito básico para qualquer pipeline gráfico 3D baseado em rasterização.

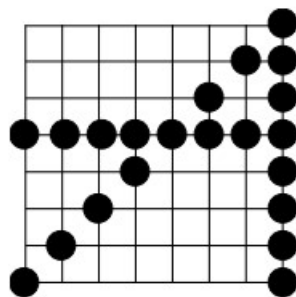


Figure 1. Rasterização de linhas horizontais, verticais e diagonais, de densidade visual variável [LOPES 2013].

Sendo ainda mais claro a respeito dessa técnica a rasterização realiza uma conversão de dados geométricos e de pixel em fragmentos. Onde cada quadrado de fragmento corresponde a um pixel no framebuffer. São levados em consideração diversos pontos como largura da linha, tamanho do ponto, modelo de sombreado e cálculos de cobertura para suportar o antialiasing assim os vértices são conectados em linhas ou os pixels interiores são calculados para um polígono preenchido. Valores de cor e profundidade são gerados para cada fragmento quadrado [SHEREINER 2009].

2.3. Reyes

Reyes é um método de renderização muito utilizado na computação gráfica 3D, desenvolvido por volta de 1980 pela Lucasfilm que atualmente é conhecida como Pixar. Reyes significa Renders Everything You Ever Saw (em português quer dizer Processa tudo o que você viu.) a arquitetura do Reyes é otimizada para renderizar de forma rápida e com alta qualidade de imagens complexas. "O objetivo pretendia ser ambicioso o suficiente para nos forçar a repensar completamente todo o processo de renderização. Buscamos ativamente novas abordagens para a síntese de imagens e conscientemente tentamos evitar nos limitar a pensar em termos de soluções tradicionais ou ambientes computacionais específicos [COOK et al. 1987]."

A arquitetura do Reyes foi desenvolvida com alguns objetivos com relação as cenas complexas que desejavam-se serem renderizadas.

- **Complexidade do modelo:** Para gerar imagens visualmente complexas e ricas, os usuários de um sistema de renderização precisam estar livres para modelar grandes números (100.000) de estruturas geométricas complexas possivelmente geradas usando modelos processuais como fractais e sistemas de partículas.
- **Complexidade de sombreado:** grande parte da complexidade visual de uma cena é gerada pela maneira como os raios de luz interagem com superfícies de objetos sólidos. Geralmente, em computação gráfica, isso é modelado usando texturas. As texturas podem ser matrizes coloridas de pixels, descrever deslocamentos de superfície ou transparência ou refletividade de superfície. Reyes permite que os usuários incorporem shaders procedurais através dos quais a estrutura superficial e a interação óptica são obtidas usando programas de computador que implementam algoritmos procedurais em vez de simples tabelas de consulta. Uma boa parte do algoritmo visa minimizar o tempo gasto pelos processadores buscando texturas de armazenamentos de dados.
- **Rastreamento de raio mínimo:** Na época em que Reyes foi proposto, os sistemas de computador eram significativamente menos capazes em termos de poder de processamento e armazenamento. Isso significa que o traçado de raios em uma cena foto-realista levaria dezenas ou centenas de horas por quadro. Algoritmos como o Reyes, que geralmente não faziam o traçado dos raios, são muito mais rápidos com resultados quase foto-realistas.
- **Velocidade:** Renderizar um filme de duas horas a 24 quadros por segundo em um ano permite 3 minutos de tempo de renderização por quadro, em média.
- **Qualidade de imagem:** qualquer imagem que contenha artefatos indesejados relacionados a algoritmo é considerada inaceitável.

- **Flexibilidade:** A arquitetura deve ser flexível o suficiente para incorporar novas técnicas à medida que se tornam disponíveis, sem a necessidade de uma reimplementação completa do algoritmo [COOK et al. 1987].

O algoritmo Reyes foi desenvolvido para superar algumas dificuldades daquela época, com ênfase particular na localização dos dados, na geometria deslocada, e anti-aliasing de borda livre de ruído, desfoque de movimento e efeitos off-line de profundidade [CHRISTENSEN et al. 2018].

”Reyes lida com a complexidade geométrica ao renderizar um ladrilho de imagem (por núcleo de CPU) de cada vez, e apenas tesselandos os remendos de superfície visível nesse ladrilho. Desta forma, os remendos de superfície são apenas tessellados quando necessário, e as grades de micropolígono são apagadas da memória assim que não forem mais necessários. Superfícies mapeadas por deslocamento não incorrer em sobrecarga de memória adicional [CHRISTENSEN et al. 2018].”

Do mesmo modo, são tratados grandes quantidades de dados de textura que geram mapas MIP lado a lado [WILLIAMS 1983] antes da renderização começa, fazendo a leitura de blocos de textura no nível MIP e armazenando os mosaicos de textura em um cache [PEACHEY 1990]. Assim é possível realizar renderizações 100 vezes melhor que a memória principal.

O conceito básico por trás dos renderizadores de micro polígonos, como Reyes, é dividir cada superfície em pequenos polígonos de tamanho de pixel e projetá-los na tela. Os renderizadores de micropolígono são bastante rápidos e oferecem vários recursos interessantes, como o mapeamento de deslocamento. Eles são capazes de lidar com superfícies paramétricas de forma muito eficiente e não há necessidade de se preocupar com estruturas de subdivisão espacial complicadas. O pipeline de Reyes pode ser dividido nas seguintes etapas:

Limite: Esta propriedade tem por objetivo estimar o tamanho de um objeto geométrico quando projetado em uma tela.

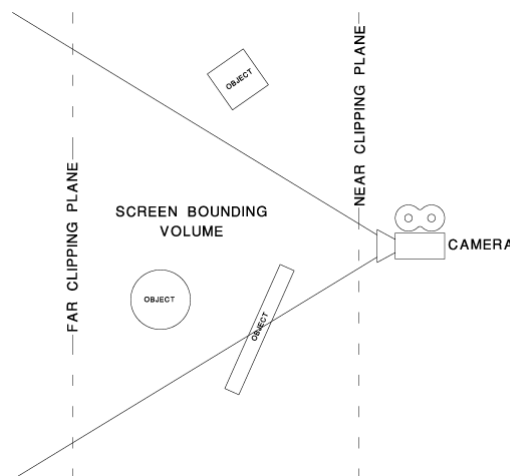


Figure 2. Limite. [BOSWELL 2008]

Existem diversas maneiras para fazer isso, porém a mais simples consiste em dividir cada pedaço da geometria e projetar seus vértices resultantes na tela.

Divisão: Dividir uma forma geométrica serve a dois propósitos. Primeiro, permite que um nível mais granular de limite seja feito. Se um objeto ultrapassar um plano de deformação, a divisão permitirá que as seções externas sejam descartadas. Segundo, a divisão garante que haja um bom equilíbrio entre a memória e o número de micro polígonos necessários para representá-los. Formas geométricas que não foram divididas o suficiente resultarão em micro redes que contêm dezenas ou centenas de milhares de micro polígonos.

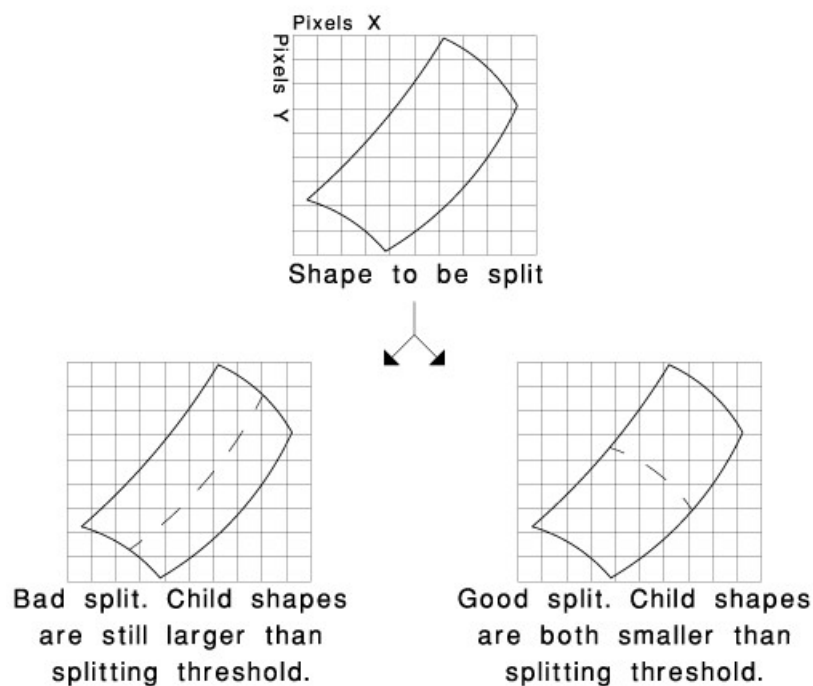


Figure 3. Divisão [BOSWELL 2008].

Micropolígonos: Quando uma forma geométrica é menor que o fronteira de divisão, ela é dividida em micro polígonos. Os micro polígonos são armazenados em uma estrutura de grade chamada micro grid. Devido à fase de delimitação e divisão, cada forma

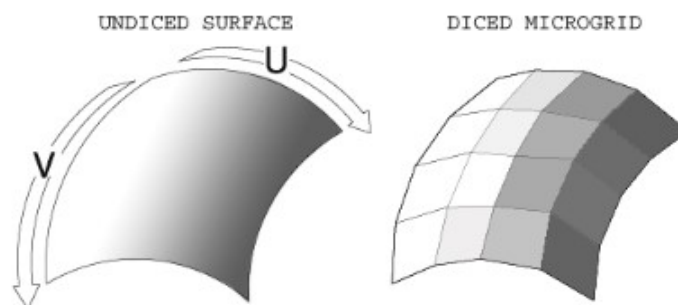


Figure 4. Micropolígonos [BOSWELL 2008].

geométrica na cena tem apenas alguns pixels de largura. A micro rede é simplesmente uma matriz 2d contendo a posição, normal, cor e opacidade associada a cada vértice em cada micro polígono.

Sombreamento: Toda forma geométrica é cortada em uma grade de micro polígonos com uma densidade de aproximadamente um micro polígono por pixel, e isso, simplifica drasticamente muitos aspectos do sombreamento. Por exemplo, não há necessidade de interpolar coordenadas de textura, pois o Reyes fornece uma estrutura extremamente flexível para definir a aparência da superfície, não há praticamente nenhum limite para o que você pode fazer dentro de um sobre a superfície.

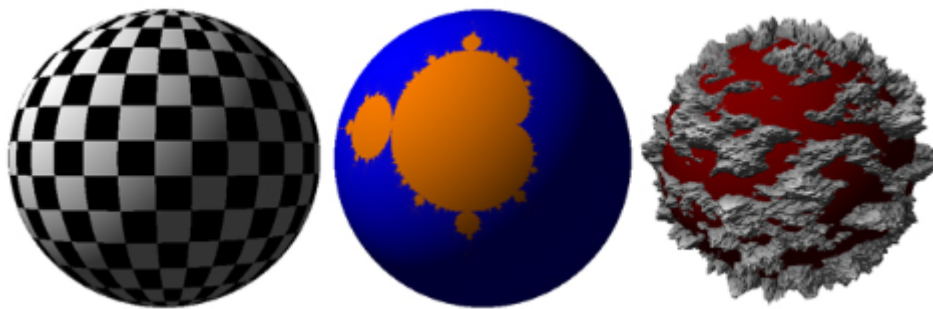


Figure 5. Sombreamento [BOSWELL 2008].

Deslocamento: Toda forma geométrica é cortada em uma grade de micro polígonos com uma densidade de aproximadamente um micro polígono por pixel, e isso, simplifica drasticamente muitos aspectos do sombreamento. Por exemplo, não há necessidade de interpolar coordenadas de textura, pois o Reyes fornece uma estrutura extremamente flexível para definir a aparência da superfície, não há praticamente nenhum limite para o que você pode fazer dentro de um saber de superfície.

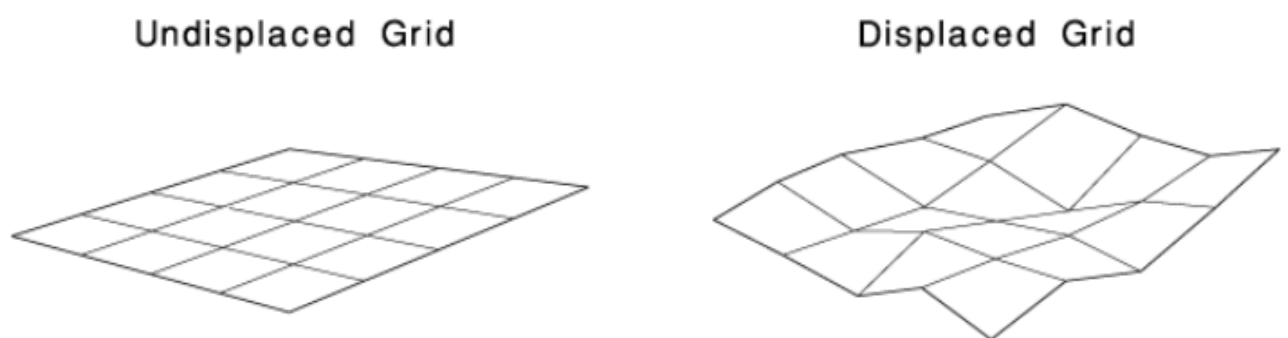


Figure 6. Deslocamento [BOSWELL 2008].

3. Aplicações

3.1. Renderman

O Renderman foi desenvolvido a partir do algoritmo de renderização de Reyes [COOK et al. 1987], ele é uma API desenvolvida pela Pixar para descrever cenas tridimensionais e convertê-las em imagens digitais de alta qualidade. Ele estabelece uma

interface entre os programas de modelagem e os renderizadores gerando imagens de qualidades fotorealísticas [PRICE 2009].

RenderMan é utilizado para criar efeitos visuais digitais até os dias de hoje, um filmes recentes de Hollywood que utilizou está API foi o The Lord of the Rings: The Fellowship of the Ring. A Figura 7 mostra imagens de recurso do filme Toy Story. Que foi renderizado com Reyes RenderMan.



Figure 7. Buzz e Woody de Toy Story (© 1995 Disney•Pixar).

”RenderMan recebeu vários prêmios da Academia, e foi o primeiro pacote de software a receber uma estatueta do Oscar® (para Ed Catmull, Loren Carpenter, e Rob Cook em 2001 “para avanços significativos no campo da renderização cinematográfica como exemplificado em RenderMan da Pixar”) [CHRISTENSEN et al. 2018].”

4. Conclusões

A indústria cinematográfica vem demonstrando uma grande evolução durante os últimos anos, que reflete principalmente com a evolução dos métodos de renderização como Reyes claro que a muito o que ser descoberto, e apesar da computação gráfica 3D está desempenhando um belo papel e de se esperar que aja uma modernização já que os filmes produzidos hoje ainda são distribuídos em forma de imagens bidimensionais. Mas podemos esperar que no futuro possamos distribuir tais filmes em forma de mundo virtuais onde o espectador poderia ter uma experiência muito mais imersa com os cenários. O Reyes foi um grande propulsor para a evolução dos renderizadores e sua importância é reconhecida até hoje.

References

- BOSWELL, A. (2008). *The Reyes Rendering Architecture*.
- BROOKHEAR, J. G. (2013). *Ciência da Computação - Uma Visão Abrangente*. Bookman Companhia Editora, 11th edition.

- CHRISTENSEN, P., FONG, J., JONATHAN SHADE, W. W., SCHUBERT, B., KENSLER, A., FRIEDMAN, S., KILPATRICK, C., RAMSHAW, C., BANNISTER, M., RAYNER, B., BROUILLAT, J., and LIANI, M. (2018). Renderman: An advanced path tracing architecture for movie rendering. *ACM Transactions on Graphics*.
- COOK, R. L., Carpenter, L., and Catmull, E. (1987). A arquitetura de renderização de imagens de reyes. *Computer Graphics SIGGRAPH '87 Proceedings*.
- LOPES, J. M. B. (2013). RasterizaÇÃo. *Instituto Superior Técnico*.
- PEACHEY, D. (1990). Texture on demand. *Technical Report 217*.
- PRICE, D. A. (2009). *A Magia da Pixar*. Editora Campus Elsevier, 1th edition.
- RODRIGUES, G. (2017). *O que é Renderização?*
- SANTOS, A. L. (2012). Elevando o nível gráfico de aplicativos interativos com uso de ray tracing em tempo real de superfícies paramétricas. *Centro de Informática UFPE*.
- SANTOS, E. T., Zuffo, M. K., Netto, M. L., and Lopes, R. D. (2001). Computação gráfica: Estado da arte e a pesquisa na usp. *EPUSP*.
- SHEREINER, D. (2009). *OpenGL Programming Guide*. Pearson Education, 7th edition.
- TECNOLOGIA, C. N. (2017). *O que é renderização ou render?*
- WILLIAMS, L. (1983). Pyramidal parametrics. *Computer Graphics (Proceedings of SIGGRAPH)*.