

1

Introdução

Segundo Velho & Gomes [35], a Computação Gráfica é a área que estuda os processos computacionais envolvendo modelos geométricos e imagens digitais. As relações entre os dados e processos da Computação Gráfica estão ilustradas no diagrama da Figura 1.1.

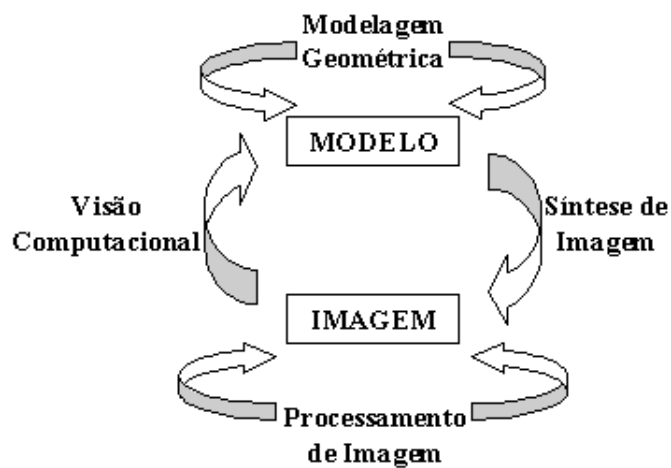


Figura 1.1: Dados e processos da Computação Gráfica. (Imagem adaptada de [35].)

A *Síntese de Imagem* realiza a transformação de modelos geométricos em imagens digitais. A transformação inversa, de imagens digitais em modelos geométricos, compreende a *Visão Computacional*. Têm-se ainda os processos que atuam exclusivamente sobre um dos tipos de dados, seja para criá-los ou para modificá-los; são estes: a *Modelagem Geométrica* e o *Processamento de Imagem*. Dessa forma, do ponto de vista computacional, existem dois tipos distintos de *dados* (modelos e imagens) e vários *processos* que atuam sobre estes dados.

Esta dissertação compreende duas técnicas muito importantes do *Processamento de Imagens*: *amostragem* e *reconstrução*, que são definidas

no Capítulo 2.

Cenas tridimensionais constituem um dos elementos fundamentais da Computação Gráfica. Uma cena 3D compreende os modelos que habitam o mundo virtual. Tradicionalmente, cada modelo é representado por um conjunto de pontos do espaço tridimensional conectados por triângulos.

A modelagem de uma cena 3D pode envolver bilhões de triângulos. Quanto mais realista (complexa) é a cena, maior é o número de triângulos requerido para representá-la. Placas de vídeo modernas, tais como as das séries *GeForce FX* e *ATI Radeon*, podem rasterizar¹ até 2 bilhões de triângulos por segundo [16]. No entanto, a crescente busca por modelos mais realistas, ocasionalmente, tem impedido o seu uso em aplicações que exijam interação em tempo real.

Entretanto, sabe-se que o uso de modelos poligonais não é a única maneira de representar um mundo virtual. Recentemente, a técnica conhecida como *Renderização Baseada em Imagem* (do inglês, *Image-Based Rendering*) vem se tornando uma área cada vez mais importante na comunidade gráfica devido ao seu potencial para produzir imagens muito realistas. Como o próprio nome já diz, *Renderização*² Baseada em Imagem é uma técnica capaz de criar novas visões de uma cena 3D que utiliza como primitivas somente imagens digitais.

Em comparação com a abordagem convencional de renderização, que se baseia em modelos poligonais, a *Renderização Baseada em Imagem* oferece os seguintes benefícios:

Foto-Realismo. Desde que novas visões são geradas a partir de imagens de alta qualidade ou até fotografias reais, é possível alcançar um alto nível de complexidade visual.

Eficiência Computacional. Geralmente, é mais rápido gerar uma nova imagem a partir de uma outra do que renderizar uma cena a partir de sua representação geométrica.

Processo Simples de Aquisição de Primitivas Gráficas. A primitiva gráfica não é mais um modelo poligonal criado por um humano em alguma ferramenta de modelagem, mas um conjunto de imagens,

¹Converter pontos, linhas ou polígonos nos *pixels* que os mesmos representam sobre a tela do computador.

²Compreende a conversão de primitivas especificadas em coordenadas no espaço do objeto para uma imagem no *framebuffer*.

isto é, fotografias ou, por exemplo, imagens geradas via o método de traçamento de raios (*raytracing*) [38].

1.1

Contexto Histórico

Pode-se dizer que as primeiras raízes da evolução dos sistemas de renderização baseada em imagens surgiram com o advento da técnica de mapeamento de textura. Esta técnica foi desbravada por Ed Catmull, desenvolvedor do *RenderMan* e co-fundador da *Pixar Animation Studio*, em sua tese de doutorado, em 1974 [8]. A técnica de mapeamento de textura foi posteriormente aprimorada por Blinn e Newell, que propuseram o uso de imagens para aproximar efeitos de iluminação global, num processo conhecido por mapeamento de ambiente (*environment mapping*) [6].

O mapeamento de textura convencional (Figura 1.2) não é robusto o bastante, pois mesmo que uma textura esteja sendo utilizada, a forma de um objeto virtual ainda depende da geometria do modelo subjacente.

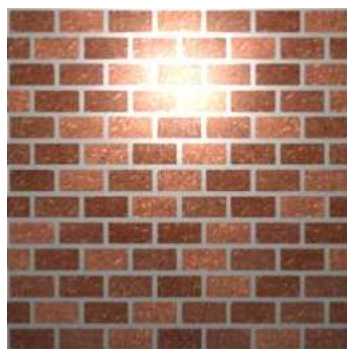


Figura 1.2: Exemplo de uma textura mapeada sobre um polígono.

Sabe-se que o efeito tridimensional na tela do computador é uma combinação de dois outros efeitos: *distorção perspectiva* e *paralaxe*³. Combinadas com informação geométrica, texturas são capazes de produzir *distorção perspectiva*, porém, são incapazes de produzir o efeito de *paralaxe*. Além disso, no mapeamento de textura convencional, onde o atributo mapeado é a cor da superfície, se as condições de iluminação no ambiente virtual onde o mapa de textura é utilizado forem diferentes das condições

³Deslocamento da posição aparente de um corpo devido à mudança de ponto de vista do observador.

de iluminação no ambiente onde a textura foi capturada, a visualização resultante do mapeamento de textura parecerá incorreta e não-realista.

Mapeamento de rugosidades (do inglês, *bump mapping* [5]) é uma solução proposta para os problemas do mapeamento de textura tradicional. Esta técnica simula a aparência de superfícies rugosas por realizar pequenas perturbações na direção dos vetores normais à superfície subjacente. Isto produz efeitos visuais muito realistas, mas a técnica assume que os valores que produzirão o efeito de rugosidade sejam pequenos quando comparados com a extensão da superfície associada. Além disso, *bump mapping* necessita ser empregado em conjunção com iluminação por *pixel*. Uma vez que a superfície, sobre a qual ocorre o mapeamento, não é alterada; arestas do contorno aparecem não modificadas e auto-occlusão, que ocorreria em situações reais, é ignorada. A Figura 1.3 ilustra o mapeamento de rugosidades.



Figura 1.3: Mesma vista da Figura 1.2 renderizada utilizando o mapeamento de rugosidades.

Além do *bump mapping*, uma outra técnica popular para representar detalhes tridimensionais de superfícies é o *mapeamento de deslocamentos* (do inglês, *displacement mapping* [12]). Esta técnica permite modificar a geometria de uma superfície tridimensional através de um mapa de profundidade, podendo desta forma criar superfícies complexas a partir de um conjunto de imagens. No entanto, *displacement mapping* necessita de uma malha de polígonos refinada, uma vez que deformará a geometria no sentido do vetor normal a cada ponto da superfície. Mapas de deslocamento podem ser utilizados para criar representações precisas de superfícies contínuas, porém o custo de processamento associado tem impedido o seu uso em aplicações interativas. A Figura 1.4 ilustra o mapeamento de deslocamentos.

Após os trabalhos sobre mapeamento de textura convencional, os próximos passos na direção da renderização baseada em imagens foram

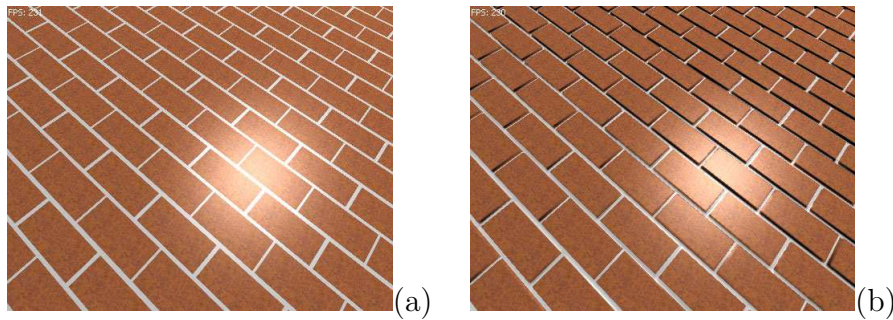


Figura 1.4: Ilustração da técnica de mapeamento de deslocamentos. (a) Textura mapeada do modo convencional. (b) Textura mapeada com deslocamento. (*Cortesia de Fabio Policarpo.*)

estabelecidos por sistemas onde as próprias imagens constituíam os aspectos significantes da descrição da cena, tais como: sistema *Movie-Map* [21], interpolação de vistas (*view interpolation*) [9] e renderização por prioridade (*priority rendering*) [30].

O *Movie-Map* é a forma mais primitiva de uma cena descrita por imagens. Nesta técnica, amostras são armazenadas em disco e acessadas de acordo com o ponto de visão. A técnica de interpolação de vistas, que deu origem ao mais famoso *software* de renderização baseada em imagens (*QuickTime VR*), trabalha com panoramas e baseia-se em *morphing* de imagens. Na renderização por prioridade, o mapeamento de *viewport* é separado do processo de renderização, de maneira que uma imagem pode ser reutilizada em um novo ponto de vista (sofrendo apenas a distorção adequada).

Em 1995, Leonard McMillan e Gary Bishop [25] apresentaram uma formalização de renderização baseada em imagens a partir do conceito de função plenóptica utilizada na área de Visão Computacional [1]. As várias técnicas de renderização baseada em imagens passaram, desde então, a ser descritas com relação à função plenóptica. A partir de 1995, inúmeros trabalhos aprimoraram a técnica de renderização baseada em imagens, tais como: campo de luz (*light field*) [20], *lumigraph* [18], *plenoptic stitching* [3], *sprites* com profundidade [34], mapeamento de textura dependente de visão [14, 15], projeção perspectiva unidimensional [31] e *nailboard* [33].

Em 2000, Manuel Oliveira [26] propôs uma extensão ao mapeamento de textura que ajuda a integrar o ideal foto-realismo da renderização baseada em imagens com as vantagens da renderização poligonal. Esta técnica, resumida na próxima seção, é o ponto de partida para o entendimento da presente dissertação.

1.2

Visão Geral do Mapeamento de Texturas com Relevo

Manuel Oliveira, em sua tese de doutorado [26], propôs uma extensão ao mapeamento de textura que suporta a representação de detalhes de superfícies tridimensionais e paralaxe do movimento da visão. Esta nova técnica, denominada *mapeamento de textura com relevo* (do inglês, *relief texture mapping*), resulta de uma fatoração da equação de *warping* tridimensional de imagens proposta por Leonard McMillan [24] em um passo de pré *warping* seguido pelo mapeamento de textura convencional [27], conforme ilustra a Figura 1.5.

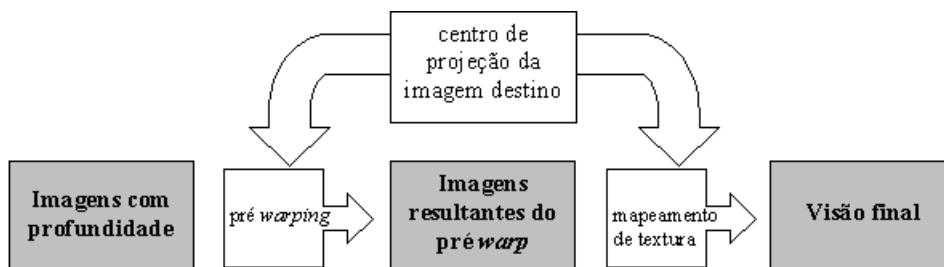


Figura 1.5: Algoritmo de mapeamento de textura com relevo. Imagens com profundidade e o ponto de vista desejado são dados como entrada para a fase de pré *warping* responsável por resolver problemas de visibilidade e preenchimento de regiões vazias. A imagem resultante é utilizada como entrada para a operação convencional de mapeamento de textura que produzirá a imagem final. (Imagem adaptada de [26].)

O passo de pré *warping* é aplicado a imagens com profundidades por *texel*⁴, denotadas *texturas com relevo* (do inglês, *relief textures*). Neste passo, são resolvidos problemas de visibilidade, preenchimento de espaços vazios causados por exposição de superfícies não representadas na imagem fonte, utilizando interpolação linear, e é computada a componente não planar da distorção perspectiva. O pré *warp* é realizado no sistema de coordenadas da imagem fonte e a visibilidade é resolvida em relação ao centro de projeção da imagem destino. A visão final é obtida mapeando-se a imagem resultante do pré *warp* sobre polígonos que possuem mesmas dimensões, posição e tamanho do plano de suporte associado à imagem fonte. O estágio de mapeamento de textura do algoritmo implementa rotação, escala, a componente planar da distorção perspectiva e a filtragem final.

⁴Elemento de textura.

Ao contrário da técnica de *bump mapping*, no mapeamento de textura com relevo, os valores dos deslocamentos podem variar arbitrariamente e os resultados visuais produzidos são corretos para observadores estáticos ou em movimento e distantes ou próximos da superfície representada. Uma vez que texturas com relevo contêm alguma informação geométrica sobre as superfícies que as mesmas representam, é possível utilizá-las como primitivas de modelagem e renderização.

O passo de pré *warping* é descrito por equações e pode ser implementado utilizando-se operações unidimensionais ao longo das linhas e colunas de uma imagem, requerendo interpolação entre somente dois *texels* adjacentes por vez. O estágio final de mapeamento de textura é eficientemente implementado utilizando o mapeamento convencional de textura suportado pelas placas gráficas.

A técnica de mapeamento de textura com relevo aumenta, de forma significativa, o poder expressivo do mapeamento convencional de textura e reduz drasticamente a quantidade de polígonos requerida para modelar uma cena enquanto preserva sua aparência realista. Os resultados obtidos com o uso desta técnica são, na maioria dos casos, virtualmente indistinguíveis da renderização de modelos geométricos mais complexos [26]. A Figura 1.6 ilustra uma comparação entre as técnicas de mapeamento de rugosidades e mapeamento de textura com relevo. As Figuras 1.6(a) e 1.6(c) representam a visualização de uma imagem através da técnica de mapeamento de rugosidades, sendo que em (c) tal imagem é visualizada de um ângulo mais oblíquo. As Figuras 1.6(b) e 1.6(d) representam as mesmas vistas das Figuras 1.6(a) e 1.6(c), porém, visualizadas através da técnica de mapeamento de textura com relevo.

1.3

Objetivos do Trabalho

O objetivo principal desta dissertação é verificar a viabilidade de uso da técnica de mapeamento de textura com relevo em aplicações que exijam interação em tempo real como, por exemplo, jogos eletrônicos.

Desse modo, este trabalho apresenta as seguintes variantes de implementação da técnica de mapeamento de textura com relevo propostas em [26]: amostragem 1D em dois passos, amostragem assimétrica em dois

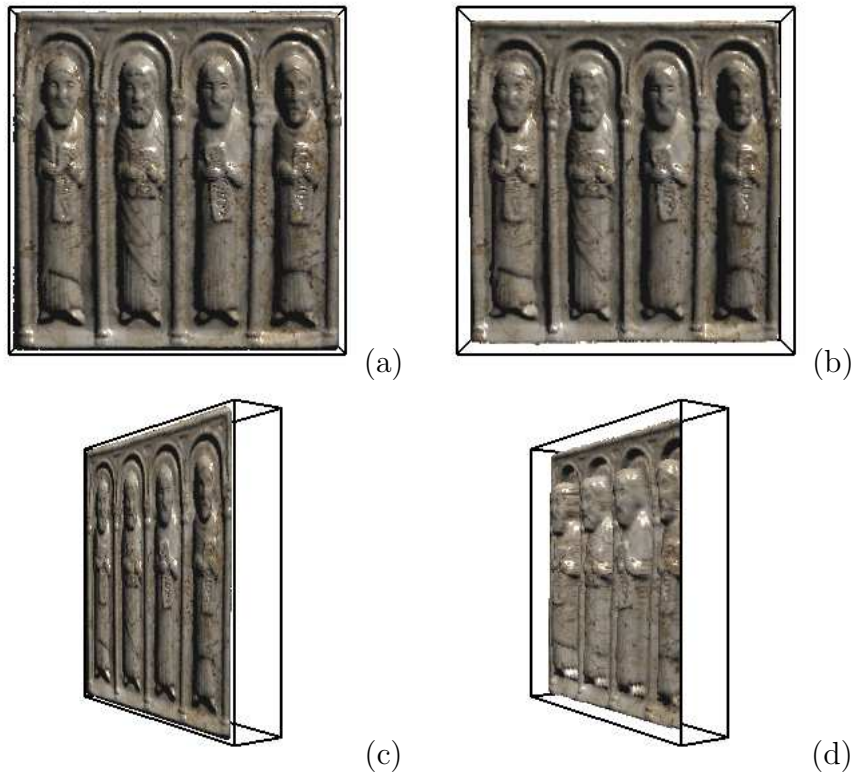


Figura 1.6: Comparação entre as técnicas de mapeamento de rugosidades e mapeamento de textura com relevo. (a) e (c) representam a visualização de uma imagem através da técnica de mapeamento de rugosidades. (b) e (d) representam as mesmas vistas das Figuras (a) e (c), porém, visualizadas através da técnica de mapeamento de textura com relevo.

passos, amostragem com compensação de deslocamento em dois passos e amostragem intercalada em um passo.

No trabalho de Manuel Oliveira [26], texturas com relevo são imagens resultantes de projeção paralela que armazenam informação de profundidade por *texel*. Mais precisamente, estas texturas são representadas como imagens RGBA⁵ onde o canal *alpha* armazena informação de profundidade e os canais restantes armazenam informação de cor. Neste trabalho, com o intuito de aproveitar a crescente evolução do *hardware* gráfico, agora com *pipeline* programável, inclui-se como objetivo acrescentar informação sobre a normal da superfície subjacente à textura com relevo, possibilitando, dessa maneira, que o cálculo de iluminação seja efetuado por *pixel*. Para implementação de tal abordagem, adotou-se a linguagem de *shading*⁶ Cg [16] (*C for Graphics*), da *NVidia*.

⁵Red, Green, Blue e Alpha.

⁶O processo de calcular o valor final da cor no interior de um polígono durante a rasterização.

Vale notar que com o aumento do poder de processamento do *hardware* gráfico, também denominado GPU (*Graphic Processor Unit*), cada vez mais tem crescido a tendência de que este assumira a maior parte do *pipeline* de visualização, deixando a unidade central de processamento (CPU, *Central Processor Unit*) com um tempo ocioso cada vez maior [11]. No sentido de aproveitar este tempo, adotou-se como objetivo estudar uma forma de paralelizar o mapeamento de textura com relevo entre CPU e GPU, utilizando-se para isto da tecnologia de *Hyper-Threading* [23]. Em poucas palavras, *Hyper-Threading* é uma tecnologia que permite a execução simultânea de múltiplas *threads*; isto é possível porque um único processador físico pode ser visto por uma aplicação como dois processadores lógicos.

1.4

Estrutura da Dissertação

Esta dissertação pode ser dividida em três partes: na primeira (Capítulo 2), é apresentada a formalização dos conceitos envolvidos no processo de mapeamento de textura com relevo, formando uma base teórica sobre a qual é desenvolvido o restante do discurso; na segunda (Capítulos 3 e 4), são discutidas as variantes de implementação da técnica de mapeamento de textura com relevo em suas versões seqüencial e paralela; e finalmente, na terceira (Capítulos 5 e 6), são apresentados os resultados da execução de todos os algoritmos.

Mais detalhadamente, o Capítulo 2 apresenta uma definição formal para imagens com profundidade e texturas com relevo, discorre sobre o processo de *warping* tridimensional de imagens, discute a intuição acerca da fatoração da equação de *warping* tridimensional de imagens, apresenta as equações de pré *warping* e apresenta o algoritmo que resolve a ordem compatível de oclusão para imagens com profundidade.

No que diz respeito à implementação da técnica abordada neste trabalho, são descritos, no Capítulo 3, os algoritmos de amostragem a partir de texturas com relevo propostos em [26], bem como suas respectivas vantagens e desvantagens. No Capítulo 4, uma discussão similar é feita para os algoritmos de amostragem a partir de texturas com relevo em sua versão paralela.

O Capítulo 5 discorre sobre a qualidade visual dos resultados obtidos

por cada algoritmo implementado, bem como sobre as estatísticas de tempo consumido durante a execução dos mesmos. Além disto, este capítulo estuda a viabilidade do uso da técnica de mapeamento de textura com relevo em aplicações que exijam tempo real.

Finalmente, no Capítulo 6, é apresentada uma conclusão final deste trabalho e suas possíveis extensões.

1.5

Contribuições Alcançadas

Uma das contribuições alcançadas por este trabalho consiste no uso de mapas de normais em conjunção com mapeamento de textura com relevo. Tal conjunção permite a captura de efeitos dependentes do ponto de vista e da direção de iluminação que antes não eram possíveis de serem representados. Tais efeitos contribuem na aparência de superfícies não-difusas. Vale notar que o uso de mapas de normais produz somente soluções para iluminação local, isto é, tal método não considera contribuições de raios de luzes resultantes de múltiplas inter-reflexões.

Além disto, duas novas abordagens paralelas são propostas com o intuito de otimizar o processamento empregado no algoritmo de mapeamento de textura com relevo, sendo que em alguns casos são obtidos ganhos de até 37% em relação ao tempo de processamento da abordagem convencional.