

1

Introdução

O termo *visualização* corresponde, no contexto desta dissertação, aos métodos que permitem a extração de informações relevantes a partir de conjuntos de dados complexos, com o auxílio de técnicas de computação gráfica (Paiva et al., 1999), constituindo-se como uma ferramenta para a interpretação e análise de dados computacionais. A área de *visualização científica* se preocupa com a extração de informações de caráter científico a partir de conjuntos de dados que representam fenômenos complexos.

Os dados associados a regiões de volumes (informações tridimensionais) são chamados *dados volumétricos*. Esses dados são geralmente amostras de uma função de difícil reconstrução, que associa valores às posições do espaço. Os valores nas posições intermediárias podem ser obtidos pela interpolação entre os valores vizinhos. O conjunto de técnicas voltadas para a visualização de dados volumétricos é conhecido como *visualização volumétrica* (Drebin et al., 1988), e constitui uma subárea da visualização científica.

Muitas vezes, os dados volumétricos estão dispostos topologicamente como uma grade (ou malha) retilínea composta por células hexaédricas (Figura 1), que pode ser representada como uma matriz de índices (i, j, k) . Uma característica dessa representação é que ela apresenta uma estrutura de conectividade implícita entre os elementos (células hexaédricas) de que é composta, permitindo que todos os vizinhos de uma célula sejam obtidos pelo incremento (ou decremento) do índice da célula atual. Essa característica define o que é chamado malha *estruturada* (Speray, 1990; Yagel, 1996). As malhas que não apresentam essa característica são chamadas malhas *não-estruturadas* e as suas células podem ser poliedros arbitrários, que por sua vez podem ser decompostos em tetraedros. Dessa forma, um modelo arbitrário pode sempre ser representado por uma malha não-estruturada de tetraedros. As malhas estruturadas também podem ser incluídas como um caso particular de malhas não-estruturadas. Os dados representados em uma malha são, geralmente, associados às suas células ou aos

vértices dessas células. Exemplos de malhas estruturadas e não-estruturadas são apresentados na Figura 2.

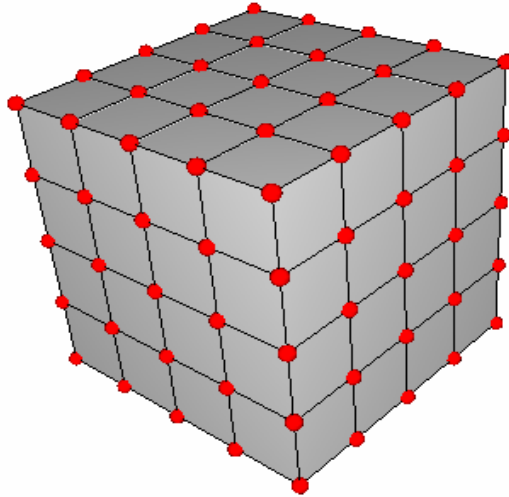


Figura 1 – Dados volumétricos dispostos como uma grade tridimensional.

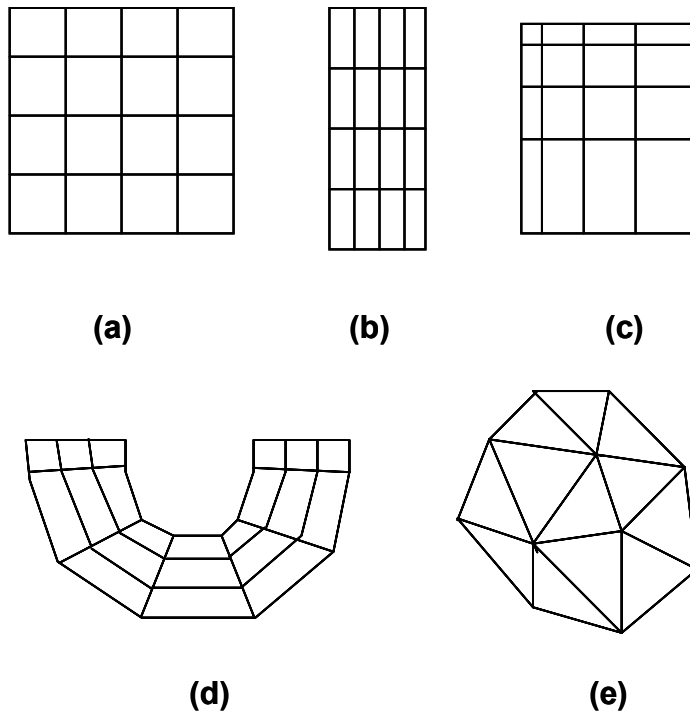


Figura 2 – Exemplos de malhas tridimensionais projetadas. As malhas são apresentadas hierarquicamente, da mais restrita à mais geral: (a) grade cartesiana ou uniforme, (b) grade regular, (c) grade retilínea, (d) malha estruturada, (e) malha não-estruturada de tetraedros.

Há duas abordagens básicas (Paiva et al., 1999) para realizar a visualização volumétrica: extração de superfícies (Figura 3b) e *renderização*¹ direta de volumes (Figura 3a). Na primeira, são construídas representações poligonais de superfícies relacionadas às características desejadas da função de interesse (p. ex., iso-superfícies), que são, em geral, visualizadas utilizando técnicas de renderização de polígonos, suportadas diretamente pelas placas gráficas atuais. Na segunda abordagem, que será a utilizada nesta dissertação, os dados são diretamente visualizados, sem a extração explícita de superfícies.

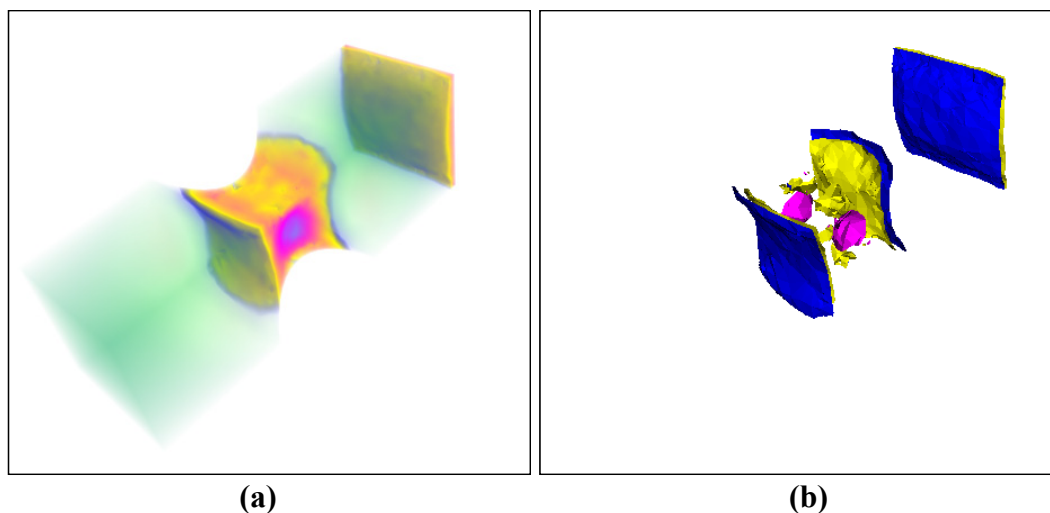


Figura 3 – Visualização volumétrica dos resultados da análise de elementos finitos para uma malha de tetraedros, usando duas técnicas: (a) renderização direta de volumes, (b) extração de iso-superfícies.

As etapas básicas para a visualização das malhas não-estruturadas utilizando renderização direta de volumes são ilustradas na Figura 4. A *classificação*, que é discutida no Capítulo 2, determina os dados que devem ser visualizados e os separa de acordo com as suas características. Isto é normalmente feito por uma *função de transferência*, que associa valores de cor e opacidade aos dados volumétricos. Após a classificação os dados são projetados para gerar uma imagem final, e, assim, poderem ser visualizados.

¹ No contexto desta dissertação, o termo “*renderização*” está relacionado ao processo de geração de imagens a partir de um conjunto de dados que descrevem a geometria e os atributos necessários para sua visualização (p. ex., coordenadas e cores de vértices de uma primitiva geométrica).

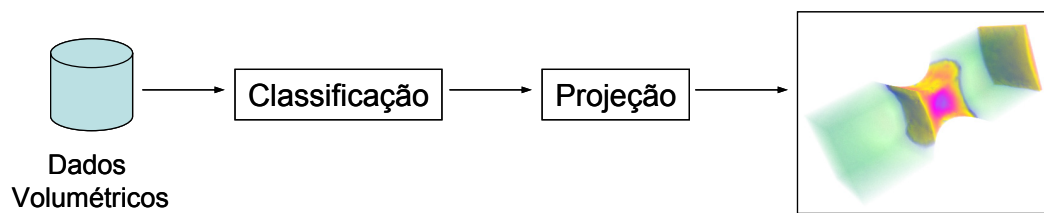


Figura 4 – Etapas da renderização direta de volumes.

Nos últimos anos, a evolução da placas gráficas utilizadas nos computadores pessoais (*PC*) permitiu a visualização interativa de diversos tipos de informações, sem a necessidade de estações gráficas de custo elevado. Uma das inovações mais importantes (e recentes) foi o surgimento das placas com unidades de processamento gráfico (*GPU – Graphics Processor Unit*) programáveis, o que aumentou dramaticamente suas possibilidades de utilização. Apesar de desenvolvidas principalmente para o mercado de jogos eletrônicos, a utilização dessas placas em outras áreas, como a visualização volumétrica, tem sido objeto de diversas pesquisas.

As GPUs modernas (Comba et al., 2003) são otimizadas para o processamento de dados vetoriais, permitindo executar simultaneamente uma operação para cada uma das componentes de um vetor de dimensão 4. Diversas operações aritméticas (p. ex., seno, exponencial, produto interno) comumente utilizadas são eficientemente implementadas com apenas uma instrução em linguagem de máquina. Os dados enviados para a placa gráfica são os vértices de primitivas poligonais (geralmente pontos, linhas ou triângulos) e alguns atributos adicionais, como cor e coordenadas de textura. Os vértices são individualmente processados, na etapa de *geometria*, e enviados para a *rasterização*, que é responsável por preencher cada posição (*pixel*) da tela de projeção pertencente à primitiva. As placas gráficas programáveis permitem intervir nessas duas etapas, por meio de um *programa por vértice* e um *programa por fragmento* (OpenGL ARB, 2005; NVIDIA, 2004a). O primeiro oferece a possibilidade de alterar as características de cada vértice enviado para a placa, enquanto o segundo permite modificar os atributos de um pixel antes que ele seja desenhado. Um pixel nesta condição é chamado de *fragmento*.

1.1. Motivação

A motivação para esta pesquisa foi o desejo de investigar técnicas de visualização volumétrica para serem adicionadas ao Pos3D (Celes, 1990), um programa para pós-processamento de resultados da análise de modelos de elementos finitos tridimensionais, desenvolvido pelo laboratório de Tecnologia em Computação Gráfica (Tecgraf) da PUC-Rio, para a Petrobras.

Uma malha de elementos finitos corresponde à discretização de um domínio geométrico contínuo. A discretização equivale a uma decomposição celular do domínio no qual o interior de cada célula, que é chamada *elemento*, representa uma região disjunta do espaço e é definida por um conjunto de *nós*. Normalmente, um elemento finito é um poliedro (Figura 5) cujos vértices são nós do elemento, e cada nó possui associado um valor de propriedade. O valor da propriedade em qualquer posição no interior do elemento é determinado pela interpolação, linear ou não, dos valores dos nós. Elementos não-lineares possuem nós que não são vértices e podem apresentar faces não-planares (Figura 5c). No caso geral, as malhas de elementos finitos podem ser consideradas como malhas não-estruturadas.

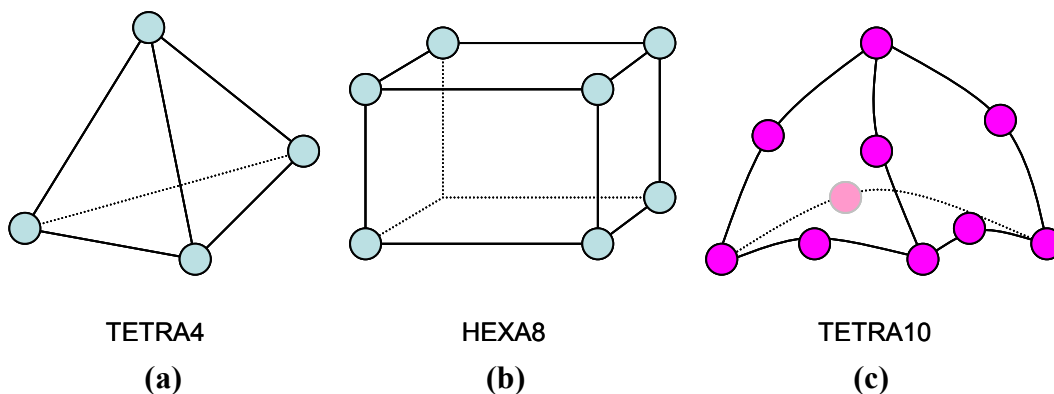


Figura 5 – Exemplos dos principais elementos finitos: (a) tetraedro linear (4 nós), (b) hexaedro linear (8 nós), (c) tetraedro quadrático (10 nós).

A aplicação das técnicas de visualização volumétrica ao Pos3D deve atender a alguns requisitos, como:

- visualizar interativamente malhas em domínios não-convexos, formadas pelos principais tipos de elementos (tetraedros, hexaedros e prismas);
- permitir a modificação interativa da função de transferência;
- permitir cortes volumétricos da malha.

Esta pesquisa se restringe às malhas de tetraedros lineares (faces planas e interpolação linear), nas quais a propriedade visualizada consiste em amostras de um campo escalar.

1.2. Objetivo

O objetivo principal desta pesquisa é investigar a visualização volumétrica interativa de malhas não-estruturadas. Para isso, são estudados e comparados os principais algoritmos que exploram os recursos de programação das placas gráficas modernas. Adicionalmente, são propostas algumas modificações sobre os algoritmos originais. As malhas não-estruturadas enfocadas neste trabalho são as de elementos finitos, com os resultados podendo ser estendidos a outras de características semelhantes, como, por exemplo, malhas de diferenças finitas usadas em simulações de reservatórios naturais de petróleo.

As principais contribuições desta pesquisa são:

- estudo e comparação dos algoritmos de projeção de células e traçado de raios para visualização volumétrica em placas gráficas programáveis;
- proposta de uma nova estruturação dos dados na placa gráfica, para eliminar o gargalo da transferência de dados do algoritmo de projeção de células;
- proposta de aplicar a integração da função de transferência na GPU ao algoritmo de traçado de raios.

O Capítulo 2 discute a classificação de dados volumétricos, fundamental para a renderização direta de volumes. A primeira seção trata do papel da função de transferência neste processo e a segunda apresenta os modelos ópticos mais utilizados, além dos métodos para sua aplicação à visualização de malhas não-estruturadas.

O Capítulo 3 apresenta algoritmos para a visualização interativa dos dados volumétricos. Dois paradigmas são discutidos: *projeção de células* e *traçado de raios*, ambos acelerados pela placa gráfica.

No Capítulo 4, são apresentadas alternativas a alguns aspectos dos algoritmos apresentados no Capítulo 3, além de uma avaliação conceitual desses algoritmos. Adicionalmente, duas modificações são propostas. A primeira (seção 4.1.1) é uma estruturação dos dados para a projeção de células, armazenando os dados na placa gráfica. A outra (seção 4.2) é uma forma de aplicar a integração de segmentos lineares de uma função de transferência (seção 2.3.2) ao paradigma de traçado de raios da seção (3.2).

Os resultados obtidos são discutidos no Capítulo 5, e, finalmente, o Capítulo 6 conclui esta pesquisa e sugere possíveis trabalhos futuros.