

VISUALIZAÇÃO VOLUMÉTRICA DE CAMPOS ESCALARES DEFINIDOS EM TRIANGULAÇÕES RETANGULARES

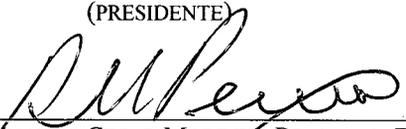
JOSÉ MARIA RIBEIRO NEVES

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por :



PROF. ANTONIO ALBERTO FERNANDES DE OLIVEIRA, D. SC.
(PRESIDENTE)



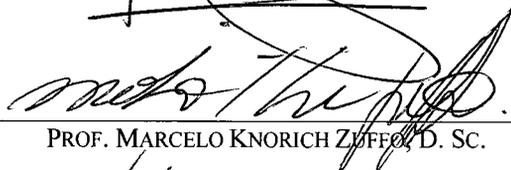
PROF. RONALDO CESAR MARINHO PERSIANO, D. SC.
(ORIENTADOR)



PROF. GEOVAN TAVARES DOS SANTOS, D. SC.



PROF. LUIZ CARLOS PACHECO RODRIGUES VELHO, PH.D.



PROF. MARCELO KNORICH ZUFFO, D. SC.



PROF. CLÁUDIO ESPERANÇA, PH. D..

Rio de Janeiro, RJ - Brasil
Janeiro 1999

NEVES, JOSÉ MARIA RIBEIRO.

Visualização Volumétrica de Campos Escalares
Definidos em Triangulações Retangulares.

[Rio de Janeiro] 1999.

vii, 120p. 29,7 cm

(COPPE/UFRJ. D.Sc., Engenharia de Sistemas
e Computação, 1999)

Tese - Universidade Federal do Rio de Janeiro,
COPPE

1. Visualização Volumétrica
2. Triangulações Retangulares

I. COPPE/UFRJ

II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D. Sc.)

Visualização Volumétrica de Campos Escalares Definidos em Triangulações Retangulares

José Maria Ribeiro Neves
Janeiro de 1999.

Orientador : Prof. Ronaldo César Marinho Persiano
Programa : Engenharia de Sistemas e Computação

Este trabalho apresenta um tipo de estrutura, denominado triangulação retangular adaptativa, como infra-estrutura para representação de dados volumétricos irregulares, no contexto da visualização científica de dados escalares. Demonstra-se a viabilidade e as vantagens de se utilizar esse tipo de estrutura em sistemas que pretendam proporcionar boas condições para exploração interativa desses dados. São definidos uma estrutura de dados e um conjunto de algoritmos para implementação de técnicas de visualização direta, baseados na organização hierárquica da estrutura. Essa característica hierárquica também serve de base para a definição de um modelo de multiresolução, que facilita a extração de modelos simplificados dos dados.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.).

Volumetric Visualization of Scalar Fields Defined on Square
Triangulations

José Maria Ribeiro Neves
January 1999

Thesis Supervisor : Ronaldo César Marinho Persiano
Department : Computation and Systems Engineering

This work presents a structure type, called adaptive rectangular triangulation, as an infrastructure for volumetric representation of irregular data, in the context of scientific visualization of scalar data. The advantages of the defined structure are shown to offer good support to systems aiming the interactive exploration of volumetric data. This work defines a data structure and a set of algorithms that implement direct visualization techniques based on the hierarchical organization of the structure. That hierarchical nature is also used as a basis for the definition of a multi-resolution model that allows the easy extraction of simplified models.

*“E tudo quanto fizerdes, fazei-o de todo o coração,
como ao Senhor, e não aos homens.”*

Paulo (Colossenses)

Agradecimento

A Deus,

pela vida

e pelos corações de boa vontade que Ele me tem permitido encontrar.

Índice

Capítulo 1	Introdução	1
1.1	Métodos de Visualização Volumétrica	3
1.2	Descrição do Trabalho	5
Capítulo 2	Modelos de Dados em Visualização Científica	8
2.1	Discretização do Domínio	9
2.2	Adaptatividade	10
2.3	Multiresolução	11
2.4	Modelagem de Dados Irregulares	13
2.4.1	Interpolação para Técnicas de Visualização	14
2.4.2	Reamostragem em Malhas Auxiliares	15
2.4.3	Interpolação para Reamostragem em Malhas Auxiliares	15
Capítulo 3	Triangulações Retangulares	17
3.1	O Que é uma Triangulação Retangular?	17
3.2	Rotulação de Vértices e Faces	21
3.3	Aspectos Combinatórios de Triangulações Retangulares	22
3.4	Triangulação Retangular Adaptativa	25
3.4.1	Organização Hierárquica	28
3.4.2	Construção de uma Triangulação RA	29
3.5	Geometria das Células	30
3.6	Relações de Vizinhaça	33
3.7	Catálogo de Orientações	35
3.7.1	Organização do Catálogo	35
3.7.2	Orientação dos Filhos da Subdivisão	38
3.7.3	Orientações Indexadas Equivalentes	40
3.8	Estrutura de Dados para Triangulações RA	41
3.8.1	Recuperação das Relações de Adjacência	43
3.8.2	Requisitos de Espaço	44
Capítulo 4	Multiresolução em Triangulações Retangulares Adaptativas	46
4.1	Modelos Simplificados	47
4.1.1	Propagação de Erro	48
4.2	Extração de Modelos Simplificados	49
4.2.1	Modelos Dinâmicos	50
4.2.2	Modelos Dinâmicos em Série	51
4.2.3	Modelos Pré-Computados	52
4.3	Metodologias para Cálculo de Erro	54
4.3.1	Simplificação do Cálculo de Erro	56
4.3.2	Avaliação Comparativa	58
4.4	Requisitos de Espaço	58
Capítulo 5	Visualização Volumétrica Direta	60

5.1	Modelos Óticos.....	60
5.2	O Modelo Adotado	61
5.2.1	Funções de Transferência	62
5.2.2	Simplificações do Modelo.....	63
5.2.3	Reflexão em Superfícies.....	64
5.3	Método de Lançamento de Raios.....	66
5.3.1	Algoritmo Básico.....	67
	Determinação da Face de Entrada	67
	Interseções com Faces Internas	68
	Transição entre células	69
5.3.2	Degenerações.....	70
5.3.3	Coordenadas Baricêntricas X Coordenadas Cartesianas	71
5.4	Método de Projeção.....	71
5.4.1	Algoritmo de Projeção de Tetraedros	73
5.4.2	Mapeamento de Textura.....	76
5.4.3	Considerações Sobre o Modelo Ótico no Método de Projeção	78
5.4.4	Ordenação das Células.....	78
	Ciclos em Malhas de Poliedros	80
Capítulo 6 O Método de Projeção para Triangulações RA.....		83
6.1	Estrutura de Dados	84
6.1.1	Catálogo de Orientações.....	84
6.1.2	Dados do Volume.....	84
6.1.3	Guia para Percurso Ordenado.....	84
6.1.4	Catálogo de Projeções	85
6.2	Algoritmo de Projeção de Tetraedros	86
6.3	Ordenação das Células.....	88
6.3.1	Algoritmo de ordenação	89
	Implementação do Algoritmo de Ordenação.....	90
6.4	Avaliação.....	92
6.4.1	Análise do Procedimento de Ordenação.....	93
6.4.2	Resultados do Procedimento de Projeção.....	93
6.5	Direções Privilegiadas de Projeção.....	95
6.6	Projeção de Modelos Simplificados	98
Capítulo 7 Lançamento de Raios em Triangulações RA.....		99
7.1	Algoritmo Hierárquico	99
7.2	Degenerações.....	103
Capítulo 8 Conclusão.....		104
Apêndice A.....		108
Referências Bibliográficas.....		112

Capítulo 1

Introdução

O termo *visualização científica* é usado como referência a uma ampla gama de processos computacionais, cujo objetivo é oferecer recursos visuais que auxiliem os especialistas na tarefa de análise de dados científicos. Constitui uma área da ciência da computação que trata do estudo e definição de técnicas, algoritmos e estruturas de dados para a produção de imagens representativas dessas informações.

Podem-se imaginar aplicações para visualização científica em todos os campos da ciência, mas seu uso efetivo tem sido predominantemente nas áreas da medicina, em engenharia e ciências físicas em geral, como instrumento auxiliar para análise de dados provenientes de simulações computacionais (dinâmica dos fluidos computacional, análise de elementos finitos, etc.), em meteorologia e em biologia.

Embora existissem iniciativas nesse sentido desde os primórdios da computação digital, uma publicação de 1987, sob os auspícios da National Science Foundation americana, [McCormick87], pode ser considerada a deflagradora de um intenso movimento de investigação nessa área.

Os crescentes volumes de informações gerados pelo grande poder de computação das novas máquinas usadas em simulações, bem como a alta resolução que passava a caracterizar os dispositivos de captura de dados, provocavam grandes mudanças na complexidade desses dados, alterando os requisitos dos métodos de geração de imagens, para viabilizar sua análise.

Por outro lado, o desenvolvimento dos dispositivos gráficos e de arquiteturas especializadas foram fatores que impulsionaram uma mudança do enfoque, na concepção de sistemas, para a incorporação de características tais como interatividade e animação. A interação do usuário com a cena é um recurso fundamental, por exemplo, para desfazer ambigüidades espaciais, que se apresentam quando um conjunto de dados tridimensional é analisado a partir de uma única imagem bidimensional. A possibilidade de girar o modelo ou alterar o mapeamento de cores utilizado na produção da imagem enriquecem sobremaneira a capacidade do usuário de identificar características relevantes dos dados. Tanto a interatividade como a animação passaram, então, a ser consideradas requisitos altamente desejáveis, quando não imprescindíveis, na composição da interface com o usuário de um sistema de visualização.

Esses fatores, em conjunto, geraram uma demanda para o desenvolvimento de técnicas que permitissem implementá-los de forma compatível com as limitações da capacidade computacional dos equipamentos usados para visualização. Essa necessidade ainda se torna mais imperiosa, quando se considera a pretensão crescente de oferecer tais recursos em sistemas de médio e pequeno porte.

Nesse contexto, os esforços nessa área se dividiram entre o aperfeiçoamento de técnicas para oferecerem maior riqueza cognitiva, e a busca de implementá-las de forma eficiente. Dentre os recursos que proporcionam implementações eficientes, deve ser ressaltado, ao lado da paralelização dos procedimentos, a redução dos dados a serem visualizados. Entre os trabalhos que visam diminuir o volume de informações, predominam aqueles voltados para a redução da complexidade espacial de superfícies [Schroeder92] [Silva95], incluindo modelos de terrenos, representação por fronteira de objetos tridimensionais e isosuperfícies produzidas na visualização de grandes conjuntos de dados volumétricos. Mais recentemente, os esforços têm sido voltados para o desenvolvimento de representações multiresolução, que buscam metodologias consistentes e econômicas para a construção de modelos que proporcionem diferentes níveis de detalhamento, e que podem ser usados para implementação de técnicas variadas. Esse tipo de abordagem será discutido mais amplamente ao longo deste texto.

O tratamento dessas questões exige considerar as variadas formas de organização dos dados, motivadas pela diversidade de técnicas com que são gerados ou capturados. De um modo geral, as estruturas podem ser classificadas como regulares e irregulares (capítulo 2). Cada uma dessas classes tem características próprias, o que implica na necessidade de avaliar questões relativas a desempenho, consumo de memória e capacidade de modelagem, visando conciliá-las em busca do melhor aproveitamento dos recursos, para alcançar os fins colimados.

As triangulações retangulares [Persiano93], que serão revistas neste trabalho, aprofundando seu estudo, são propostas como uma opção intermediária, apresentando características de regularidade ao mesmo tempo que oferecem flexibilidade para modelagem. Como solução intermediária, essas estruturas não herdam apenas fatores positivos, mas também carregam desvantagens daquelas tradicionalmente utilizadas, consideradas extremas. Surge, então, a questão natural: o que predominará na estrutura intermediária, as componentes negativas ou as positivas? A contribuição deste trabalho é a demonstração de que o resultado é positivo no que se refere à aplicação em visualização. Mais especificamente, o estudo visa as aplicações em *visualização volumétrica de campos escalares*, ou seja, os dados referem-se a um campo escalar cujo domínio é tridimensional, embora muitas das idéias que serão discutidas são diretamente aplicáveis ou facilmente adaptáveis a outras dimensões.

Do ponto de vista das técnicas utilizadas, embora haja muito em comum entre visualização científica e computação gráfica, podendo mesmo considerar aquela como uma subárea desta, vale ressaltar as peculiaridades que justificam considerá-las em separado. Pode-se dizer que o objetivo da computação gráfica seja a representação de um fenômeno ou cena bem conhecidos, buscando aproximar a imagem da impressão visual da realidade. Os dados são descrições bem definidas, em alguma forma analítica ou geométrica, dos elementos que constituem a cena, incluindo atributos de cor, opacidade, textura, etc. Ainda que sejam sintetizadas, as cenas são definidas, geralmente, a partir da suposição de sua similaridade com objetos reais.

Já a visualização científica visa facilitar, através de imagens, a compreensão de um fenômeno do qual não se conhecem as propriedades, mas apenas um conjunto de informações, geralmente esparsas e incompletas. Esses dados, que podem ser de naturezas diversas, em muitos casos, não têm nenhuma associação intrínseca com

atributos visuais. Essa associação é feita de forma abstrata, em busca unicamente de proporcionar condições que facilitem a exploração visual dos dados, para identificar características relevantes [Globus94]. Utilizam-se técnicas variadas, que podem resultar em imagens de caráter absolutamente distinto, mas que constituem recursos complementares de análise.

1.1 Métodos de Visualização Volumétrica

Seja um conjunto de dados escalares, representativo de um fenômeno ou de uma função cujo domínio (volume) é uma região do espaço tridimensional. Considere-se que esse volume esteja subdividido em células, de acordo com a distribuição dos pontos amostrais disponíveis. As técnicas usualmente empregadas para gerar imagens bidimensionais a partir desses dados enquadram-se em três categorias: *cortes* e extração de *isosuperfícies* (superfícies de nível), as mais tradicionais, e *visualização volumétrica direta*, que vem recebendo maior atenção nos últimos anos, por parte dos grupos de pesquisa das áreas envolvidas. Essas técnicas são encontradas freqüentemente nos sistemas comerciais de visualização como ferramentas conjugadas, que somam subsídios para que o cientista alcance uma boa compreensão do fenômeno de onde seus dados provêm.

A visualização de cortes de um volume é um recurso bastante usado. Determinam-se as células do volume interceptadas pela superfície de corte, para que se possam interpolar os dados sobre essa superfície. A partir daí, aplicam-se técnicas que normalmente incluem o mapeamento de cores em função dos dados escalares e o traçado de contornos. É usual utilizar superfícies de corte definidas por funções implícitas. A aplicação típica é o corte do conjunto de dados por um plano, mas também é freqüente utilizar outras formas de definição dessas superfícies, por exemplo, uma superfície de nível de uma variável, sobre a qual se pode mapear outra variável. Cortes planos também são usados para simular técnicas de visualização volumétrica direta: atribuindo-se propriedade de transparência a um conjunto desses cortes e projetando-os de trás para a frente, obtém-se um resultado visual que proporciona uma noção global do volume, apesar da perda das informações relativas às regiões entre os planos. Essa técnica pode ser implementada com controle adaptativo, de forma a garantir que todas as células contribuirão com a imagem final, ou para limitar o número de células excluídas [Yagel96b].

Extração de isosuperfícies é a técnica para exploração de dados volumétricos equivalente ao tradicional traçado de curvas de nível para funções bivariadas. Dado um valor escalar q , extrair a isosuperfície de q consiste em determinar a superfície $C(q) = \{p \mid f(p) = q\}$. A menos de casos particulares, esses pontos p não são pontos para os quais os valores da variável problema sejam conhecidos. Assim sendo, se existem dois pontos da amostra original com os valores escalares acima e abaixo de q , então a isosuperfície $C(q)$ passa entre eles e, se existe uma aresta da estrutura adotada para decomposição do espaço ligando esses dois pontos, então a interseção dessa aresta com a superfície é dada por algum método de interpolação, sendo utilizada com maior freqüência a interpolação linear.

Embora apresente limitações, a técnica de extração de isosuperfícies tem um papel de grande importância na investigação de dados escalares. A posição de uma isosuperfície e a sua relação com outras isosuperfícies vizinhas, podem facilitar a identificação de detalhes estruturais dos dados. Uma de suas aplicações mais frequentes é em visualização de dados em medicina, onde é particularmente útil na caracterização das superfícies de contato entre diferentes tecidos. No entanto, seu potencial de utilização se estende a praticamente todo tipo de aplicações que envolvam análise de dados escalares.

O processo computacional de extração de uma isosuperfície pode ser dividido em duas fases. A primeira é uma fase de pesquisa, onde são identificadas as células ativas da estrutura, aquelas que são efetivamente interceptadas pela superfície. Numa segunda fase, de geração, aplica-se um algoritmo para determinar a topologia e a geometria dessa interseção. É a fase de pesquisa que consome o maior esforço computacional. Considerando N o número total de células da estrutura e K o número de células ativas, estima-se que o valor típico de K é $O(N^{2/3})$ [Itoh95] citado por [Chiang98]. Algoritmos que percorrem toda a estrutura para identificar as células ativas [Lorensen87] [Chernyaev95] são considerados ineficientes. Assim sendo, a maior parte dos esforços nessa área estão dirigidos no sentido de evitar esse percurso. [Wilhelms92] usa “octrees” para promover uma ordenação espacial que proporciona aceleração desse processo em estruturas regulares. [Livnat96] apresenta um algoritmo eficiente tanto para estruturas regulares quanto para as irregulares. [Shen96] aperfeiçoa esse algoritmo e propõe sua paralelização.

Apesar de seu inegável valor, o uso de técnicas de extração de isosuperfícies para observar dados volumétricos envolve perda do conteúdo de informações presentes naqueles dados. Uma superfície de nível corresponde, na verdade, a apenas uma porção infinitesimal do domínio sob investigação. Diante disso, buscou-se desenvolver técnicas que permitissem obter imagens bidimensionais que representassem de maneira mais completa esses dados. A partir daí começam a surgir experiências de visualização volumétrica direta, cujos métodos consideram o volume como se fosse preenchido por uma massa semitransparente, semelhante a uma gelatina colorida, com cor e densidade variável em função do valor escalar em cada ponto do domínio. As imagens produzidas veiculam informações do volume como um todo, mas a um custo computacional maior. Em consequência, paralelamente com as iniciativas no sentido de explorar as possibilidades qualitativas desses novos métodos, também aqui são mais volumosos os esforços que se têm despendido para desenvolver algoritmos mais eficientes, visando a melhoria das condições de interatividade desse tipo de tarefa.

Uma das maneiras de implementar visualização volumétrica direta é simular o lançamento de um raio para cada pixel da imagem a ser formada, partindo do olho do observador e atravessando o volume, até interceptar um objeto opaco ou até reduzir a transparência a zero, pela influência da porção do volume já atravessada, o que impede a visibilidade da região posterior [Levoy88] [Garrity90]. Esta é chamada técnica de *lançamento de raios* ou “*ray casting*” (baseada no *espaço da imagem*). Para calcular a influência de cada célula atravessada pelo raio sobre a cor do pixel correspondente, é necessário definir uma metodologia de associação dos valores escalares às propriedades óticas relevantes (cor e opacidade ou transparência). Existem diversos modelos óticos criados para esse objetivo. Esse processo implica na necessidade de estabelecer um método de interpolação para determinar o valor escalar em cada ponto do domínio, bem

como na realização de uma integração das contribuições infinitesimais ao longo do raio. Por considerar o efeito de transparência das células, é necessário que elas sejam tratadas em uma ordem consistente, de trás para a frente ou da frente para trás. Essas questões serão tratadas com maior profundidade no capítulo 5.

Uma abordagem alternativa para visualização volumétrica direta, consiste em percorrer todas as células do volume e projetá-las sobre o plano da imagem, determinando os pixels que ela influencia e quantificando essa influência [Westover90] [Max90] [Shirley90] [Lacroute94]. Esta é conhecida como técnica de *projeção*, sendo sua implementação dirigida pela estrutura do volume, o que justifica ser considerada uma técnica baseada no *espaço do objeto*. Sua aplicação depende de uma ordenação das células por visibilidade, de forma a que elas possam ser visitadas em uma ordem consistente. Normalmente são projetadas primeiro as células mais distantes do observador, à guisa do algoritmo do pintor, simplesmente aplicando cada célula sobre a imagem resultante da projeção de suas antecessoras, compondo [Porter84] a contribuição da nova célula com as já acumuladas nos pixels alcançados por ela. Também fica para o capítulo 5 o aprofundamento deste método.

Deve-se ressaltar ainda, que a visualização volumétrica direta permite observar as superfícies de nível, sem a necessidade de construir uma representação separada de sua geometria. Isso é possível promovendo-se, durante a aplicação de um método direto, um tratamento especial para aquelas células interceptadas pela superfície desejada. Este é um esquema bastante atraente, especialmente pelo potencial cognitivo proveniente da combinação de diferentes técnicas em uma mesma imagem. Essa possibilidade, no entanto, não dispensa a extração de isosuperfícies como recurso de visualização, uma vez que, diante das condições atuais da infra-estrutura de hardware e software, ainda é mais eficiente interagir com uma aproximação poligonal da superfície isoladamente. A combinação de técnicas, naturalmente, implica em maior demanda de recursos computacionais para gerar uma imagem, inclusive porque pode ser necessário percorrer toda uma extensa estrutura para realizar a operação.

Este trabalho dará ênfase às técnicas de visualização volumétrica direta, considerando que são as mais exigentes quanto aos recursos computacionais necessários e, portanto, devem ser priorizadas na concepção das estruturas de representação dos dados, adotadas para um sistema de visualização que pretenda oferecer as diversas técnicas.

1.2 Descrição do Trabalho

Este trabalho insere-se no contexto das investigações em visualização científica, mais especificamente, visualização volumétrica, como uma contribuição à busca de estabelecer formas de representação de dados que facilitem a realização das tarefas computacionais inerentes a essa prática.

De um modo geral, os resultados obtidos demonstram a viabilidade e as vantagens de se utilizar triangulações retangulares adaptativas como infraestrutura para representação de dados volumétricos irregulares, em sistemas que pretendam proporcionar boas condições para exploração interativa desses dados. Em termos mais específicos, as seguintes contribuições devem ser ressaltadas:

1. aprofundamento do estudo das triangulações retangulares, enfatizando o caráter de adaptatividade e de hierarquia dessas estruturas. São identificados e codificados aspectos combinatórios relevantes;

2. concepção de uma estrutura de dados muito simplificada, em relação às outras estruturas adaptativas descritas na literatura, baseada na organização hierárquica e na metodologia construtiva das triangulações retangulares;

3. prova da inexistência de ciclos de visibilidade entre as células das triangulações retangulares;

4. concepção de algoritmos, baseados na organização hierárquica dessas estruturas, que minimizam cálculos geométricos para:

- . ordenação direcional das células

- . implementação das técnicas de visualização direta (lançamento de raios e projeção);

5. implementação do algoritmo de projeção em triangulações retangulares, com realização de testes para avaliação comparativa de desempenho;

6. definição de um modelo de multiresolução, também baseado no caráter hierárquico das triangulações retangulares, de onde obtêm-se modelos simplificados dinâmicos ou pré-computados.

A abordagem inicia-se por uma visão geral de modelos de dados em visualização científica, no capítulo 2, dando ênfase aos modelos para dados com distribuição irregular e ressaltando a importância do aspecto de adaptatividade das estruturas associadas. O capítulo introduz também o conceito de multiresolução, que permite a utilização de modelos simplificados, para proporcionar uma rápida exploração dos dados e a melhoria progressiva da qualidade das imagens.

O capítulo 3 revê os conceitos relativos às triangulações retangulares adaptativas, aprofundando sua análise e organizando de forma sistemática seus aspectos combinatórios e as informações geométricas e topológicas. Seu caráter hierárquico e aspectos construtivos são dissecados. O resultado serve de subsídio para a definição de uma estrutura de dados compacta e eficiente, baseada naquela hierarquia, sobre a qual serão definidos o modelo de multiresolução para as triangulações retangulares adaptativas e os algoritmos para implementação dos métodos de visualização direta sobre essas estruturas.

As triangulações retangulares adaptativas multiresolução constituem o tema do capítulo 4. Trata-se de uma estrutura enriquecida com informações sobre o erro inerente a simplificações que ela pode sofrer. Metodologias alternativas para o cálculo de erro são propostas. Apresentam-se métodos dinâmicos para a obtenção de um modelo simplificado dos dados, definido por um limite constante de erro tolerável nessa aproximação. Esses métodos disponibilizam modelos que atendam a tolerâncias especificadas, sem a necessidade de percorrer toda a estrutura. Modelos simplificados com tolerância variável ao longo do domínio também são desejáveis, e podem ser obtidos como modelos pré-computados.

No capítulo 5, faz-se uma revisão de visualização volumétrica direta, descrevendo em detalhes algoritmos eficientes para implementação das técnicas de lançamento de raios e de projeção sobre triangulações genéricas. Define-se o modelo ótico adotado para esses métodos e analisa-se a questão de ordenação direcional das células dessas estruturas.

A implementação daquelas técnicas sobre triangulações retangulares é tratada nos capítulos 6 e 7. Apresenta-se uma estrutura de dados auxiliar para o método de projeção, que permite substituir diversas operações aritméticas envolvidas nos processos de ordenação e projeção das células por consultas a tabelas pré-computadas. Demonstra-se que as células de uma triangulação RA podem ser ordenadas em relação a qualquer direção e descreve-se um algoritmo para realizar essa ordenação, também esse baseado na hierarquia dessas estruturas. Resultados de experimentos computacionais relativos a esse método também estão incluídos, bem como um estudo sobre direções de projeção que se mostram especialmente vantajosas para aplicação desse método sobre triangulações retangulares.

O capítulo 7, dedicado ao método de lançamento de raios, discute a exploração da hierarquia das triangulações retangulares para sua implementação. O resultado é um algoritmo que, embora não tenha sido experimentado na prática, apresenta claras vantagens teóricas, além de utilizar os mecanismos de multiresolução de forma simples e eficiente.

Os resultados alcançados são analisados no capítulo 8, concluindo favoravelmente pela conveniência de utilizar triangulações retangulares como estrutura de representação de dados volumétricos para sistemas de visualização. Essas estruturas são especialmente vantajosas para a exploração de conjuntos de dados distribuídos irregularmente, quando for admissível a reamostragem desses dados para os nós dessas triangulações, situação em que elas se mostram nitidamente superiores às tradicionais malhas regulares.

Capítulo 2

Modelos de Dados em Visualização Científica

As aplicações de computação científica, normalmente usam dados discretos que, frequentemente, precisam ser associados a modelos contínuos para que possam ser tratados, de forma consistente, como representativos de todo o domínio sob investigação. Em particular, as técnicas de visualização desses dados envolvem reamostragem da grandeza em estudo em posições diferentes daquelas onde ela é conhecida originalmente. O objetivo dessa reamostragem pode ser a constituição de uma estrutura intermediária mais conveniente ou o atendimento a requisitos próprios da técnica escolhida, como por exemplo, a avaliação daquela grandeza nos pontos de interseção de um raio com uma célula da decomposição do domínio. Qualquer uma dessas situações pressupõe a existência de uma função que associa um valor a cada ponto do domínio em questão.

Este capítulo trata da modelagem de campo escalar, do qual se conhecem apenas amostras localizadas em pontos distribuídos em um domínio tridimensional. Esses dados podem ser representados por quádruplas (x_i, y_i, z_i, f_i) , $i=1, \dots, N$, onde $p=(x, y, z)$ são as variáveis independentes, representando pontos do domínio D , e f é a variável dependente, sendo x, y, z e f números reais.

A modelagem consiste em estabelecer uma função $F(p)$, definida em todo o domínio D , que interpole esses dados, ou seja, $F(p_i)=f_i$, $i=1, \dots, N$. Essa função é, então, usada para avaliar a grandeza escalar que ela representa em qualquer ponto $p \in D$.

Existem vários métodos para construir essa função interpoladora. Uma das linhas de diferenciação entre eles está relacionada com a distribuição dos pontos amostrais no domínio. A seção 2.1 apresenta uma classificação para a variedade de estruturas relacionadas às diferentes formas de distribuição desses pontos.

Outra linha de diferenciação entre as técnicas de modelagem de dados é a classe de continuidade da função interpoladora. A maioria dos modelos adotados na prática são da classe C^0 , ou seja, apenas a continuidade da função é garantida, mas não a de suas derivadas. Essa opção se justifica pela simplicidade, que se reflete em um baixo custo computacional dos processos que empregam o modelo. No entanto, com a disponibilidade crescente de hardware de alto desempenho, começam a se tornar atraentes modelos mais sofisticados, de classe C^1 , apresentando continuidade do gradiente [Nielson93] [Foley93]. A seção 2.4 apresenta alguns desses modelos.

2.1 Discretização do Domínio

Neste ponto, faz-se necessário estabelecer uma terminologia para servir de referência às diversas formas de discretização do domínio utilizadas em computação científica e, em particular, em visualização volumétrica, facilitando assim a exposição de idéias e conceitos. Isso será feito com base nas abordagens encontradas em [Speray90] e [Elvins92]. Outros autores propõem classificações diferentes, por exemplo [Haber91] (citado em [Williams92a]).

A maioria dos sistemas computacionais existentes, voltados para visualização de dados, pressupõe uma organização regular dos dados, considerando o domínio subdividido em *elementos* cujas bordas são definidas por arestas retilíneas de comprimento constante, fazendo ângulos retos entre arestas adjacentes. Entre eles se incluem desde os tradicionais sistemas para tratamento de funções bivariadas, que traçam mapas de isolinhas como representações gráficas dessas funções, até os sistemas modernos de visualização de dados tridimensionais provenientes de tomografia e ressonância magnética. Esses sistemas consideram o domínio subdividido em quadrados e em cubos, respectivamente. Em ambos os casos, todos os elementos de discretização (quadrados e cubos) são idênticos e têm suas arestas alinhadas com os eixos de referência. Será adotado o termo *grade regular* (ou malha regular) para designar essa forma de distribuição de dados, incluindo a possibilidade de que as células não sejam necessariamente quadrados/cubos, mas retângulos/paralelepípedos retos, com distanciamento entre pontos diferenciado nas três direções básicas.

Nas grades regulares, as posições dos vizinhos de um dado nó da malha estão definidas implicitamente, a partir da orientação dos três eixos coordenados e de um parâmetro de espaçamento definido para cada um desses eixos. Nessa organização, a representação dos dados fica simplificada, sendo reduzida, normalmente, a um arranjo tridimensional (ou n-dimensional, no caso mais geral) de números reais, que são os valores da função associados a esses pontos.

Qualquer outra forma de organização dos dados e subdivisão do domínio será considerada *irregular*. Ficam incluídas aqui as grades retilíneas com distanciamento irregular entre nós, as grades com arestas curvilíneas e quaisquer outras malhas que apresentem irregularidade topológica ou geométrica. Nessa classe, a representação dos dados inclui, além do armazenamento do valor da função em cada ponto amostral, a indicação explícita de sua localização no espaço e um conjunto de informações de conectividade capaz de caracterizar vizinhança, permitindo uma implementação eficiente das técnicas em foco.

A figura 2.1 ilustra alguns tipos de malhas bidimensionais de cada uma dessas classes.

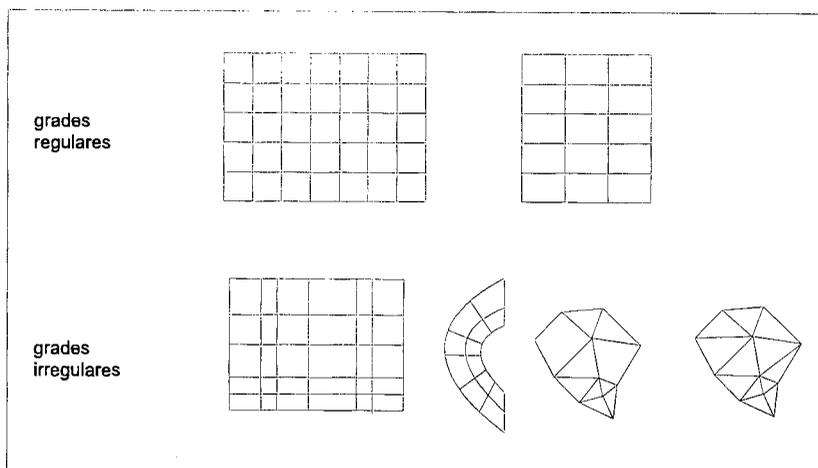


Figura 2.1 – Diferentes formas de organização dos dados

Entre as irregulares, as malhas de tetraedros (malhas simpliciais, também referidas como triangulações ou “tetraedrizações”) receberão atenção especial neste texto, devido à sua simplicidade, conveniências algorítmicas e pelo fato de que as outras formas de organização de dados podem ser, e freqüentemente são, substituídas por malhas simpliciais.

Apenas para explicitar discrepâncias existentes na terminologia empregada nas diversas fontes bibliográficas disponíveis, [Haber91] classifica como malha irregular apenas as estruturas que apresentam tanto irregularidade topológica quanto geométrica. Assim, algumas malhas com arestas curvilíneas, por exemplo, poderiam ser incluídas entre as regulares.

2.2 Adaptatividade

Uma característica de certos tipos de malhas é permitir a existência de regiões do domínio com refinamento diferenciado. Em determinadas aplicações é importante que regiões críticas sejam discretizadas em um grande número de pequenas células, enquanto outras regiões do mesmo domínio podem ser subdivididas em células maiores. Em simulações computacionais, esse requisito está associado à necessidade de conciliar minimização de erros numéricos com a minimização do custo de representação e processamento. Em outros casos essa heterogeneidade de distribuição espacial pode ser proveniente de uma amostragem sujeita a condicionamentos diversos.

Analisados sob essa ótica, os tipos de malhas relacionados acima levam a um impasse. A regularidade facilita a representação e admite algoritmos que proporcionam eficiência de processamento, mas não oferece flexibilidade para adaptatividade. No outro extremo, a irregularidade das malhas simpliciais tem implicações opostas: maior custo de representação e processamento, porém com grande flexibilidade adaptativa.

O tipo particular de estrutura focado neste trabalho, a *triangulação retangular adaptativa*, pode ser classificada como “quase regular”, e constitui uma solução intermediária, capaz de conciliar adaptatividade com eficiência de algoritmos e representação econômica. Trata-se de uma subdivisão simplicial de um domínio

constituído de um arranjo regular de quadrados, no caso bidimensional, ou cubos, em três dimensões, subdivisão essa construída segundo uma particular metodologia. A figura 2.2, retirada de [Persiano93], mostra um exemplo bidimensional, onde uma triangulação retangular adaptativa, que doravante será chamada de *triangulação RA*, é aplicada na discretização de uma região do espaço. O critério de refinamento é definido em função da aplicação. Nesse exemplo, o critério aprova um triângulo se ele não contém nenhuma raiz da equação $x^7 - y^5 + x^2y^3 - xy^2 = 0$, ou se ele é menor do que um limite de tamanho especificado.

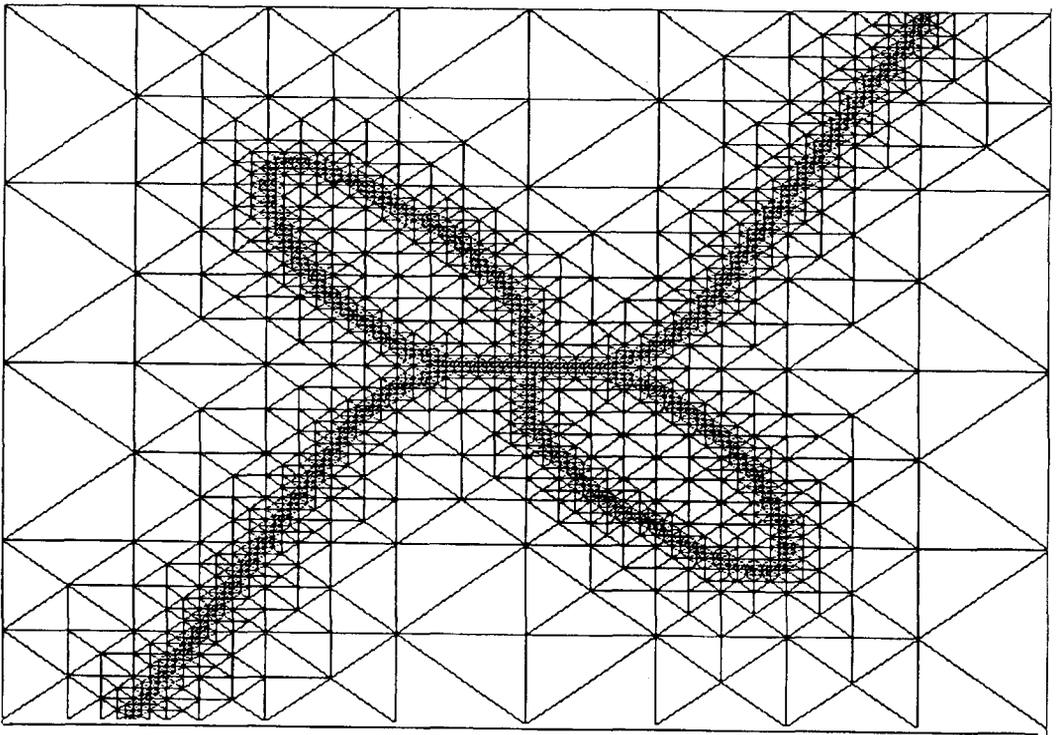


Figura 2.2 – Triangulação RA mostrando o conjunto de raízes de uma equação.

O próximo capítulo apresenta um estudo detalhado dessas estruturas, abordando suas propriedades, metodologia de construção e aspectos que as tornam atraentes do ponto de vista algorítmico. Ao longo deste trabalho as triangulações RA serão exploradas no que se refere à sua aplicabilidade em visualização de dados volumétricos.

2.3 Multiresolução

A possibilidade de escolher o nível de detalhes com que se quer trabalhar em diferentes circunstâncias é extremamente atrativa, sendo uma cogitação cada vez mais presente no projeto de sistemas que devem tratar grandes massas de dados. Particularmente em sistemas de visualização científica, esse é um requisito fundamental quando se pretende construir interfaces interativas, pois, além do grande volume dos

dados, as técnicas utilizadas apresentam, em geral, custo computacional elevado. A abordagem de multiresolução permite uma rápida análise da estrutura geral dos dados, quando a presença de detalhes pode, não apenas ser dispensável, mas até ser inconveniente.

A demanda por modelos multiresolução tem aumentado também devido à proliferação de aplicações a distância, envolvendo transferência de dados via rede de média ou baixa velocidade. Um dos recursos de interface que viabilizam incluir em tais aplicações características visuais sofisticadas é justamente a progressividade da qualidade da imagem. A transferência de modelos simplificados, portanto de menor volume, com detalhamento progressivo, proporciona “feedback” mais rápido para o usuário, bem como uma menor sobrecarga da linha de comunicação, uma vez que a transmissão integral pode ser desnecessária.

Diversas abordagens têm sido usadas para esse problema. A mais simples é prover a base de dados de uma série de modelos com diferentes níveis de detalhamento. Nesse caso, o número de níveis, normalmente pequeno, é pré-estabelecido pelo criador da base de dados, assim como a própria especificação desses níveis.

Esse esquema evoluiu para a definição de um único modelo, do qual se podem extrair modelos mais simples, cada um deles representando o modelo completo com um certo grau de aproximação. Essa forma proporciona uma flexibilidade muito maior para escolha do nível de detalhe conveniente em cada circunstância, de acordo com requisitos específicos, que podem ser determinados em função da precisão desejada e/ou de limitações de recursos computacionais. Um modelo completo, com essas características, é o que se denomina propriamente de modelo multiresolução. É desejável que esses modelos permitam, inclusive, a extração de modelos simplificados especificados por diferentes níveis de precisão em diferentes partes do domínio. Além dessa flexibilidade eles, em geral, também são mais econômicos no que tange ao espaço ocupado.

Diversas propostas têm sido apresentadas, em trabalhos provenientes de diferentes áreas de aplicação, buscando chegar a formas eficientes de implementação de multiresolução. Na área de modelagem de superfícies, [De Floriani95] propõe uma estrutura hierárquica baseada em partições do domínio em células triangulares. Essas partições são aninhadas, de modo que, na geração da estrutura, um triângulo é refinado substituindo-o por uma triangulação do seu domínio. O trabalho mostra que a organização de tais modelos em árvore constitui uma infra-estrutura que facilita o processamento de operações espaciais.

[Hoppe96] propõe um modelo, denominado “malhas progressivas”, baseado na simplificação de uma malha por uma série de transformações em que uma aresta é substituída por um ponto (“edge-collapse”). A reconstrução do modelo completo se dá por operações inversas (“vertex-split”), em um processo em que não há perda de informações. Muitos outros trabalhos em modelagem multiresolução de superfícies já foram publicados. As listas de referências dos dois citados acima constituem boas fontes de indicações.

Passando à modelagem de volumes, onde se situa nosso interesse, as referências são bem mais escassas. Dois trabalhos pioneiros na área de visualização volumétrica

são: [Laur91] e [Wilhelms92]. Ambos utilizam estruturas hierárquicas (“octrees”) [Samet90] como forma de armazenar informações resumidas, relativas a uma amostra regular de um campo escalar, em diferentes níveis. Esses trabalhos estão voltados para métodos específicos de visualização: um método de projeção e o método de extração de isosuperfícies, respectivamente, mas podem ser vistos numa ótica mais geral.

[Bertolotto94] apresenta uma versão tridimensional equivalente à estrutura hierárquica de triângulos citada acima [De Floriani95]. Trata-se, portanto, de uma subdivisão do domínio em uma hierarquia de tetraedros, que define um modelo multiresolução para representação de campos escalares volumétricos, podendo ser usado para representação de dados com distribuição arbitrária.

[Wilhelms96] traz para o âmbito dos dados irregulares a abordagem de seu artigo citado anteriormente. A estratégia é utilizar uma árvore (“k-d tree”) para organizar os dados espacialmente. Cada nó da árvore armazena informações que permitem utilizá-lo como uma aproximação de sua sub-árvore, incluindo uma avaliação do erro inerente a essa simplificação.

[Popovic97] estende o modelo de Hoppe para o denominado “complexo simplicial progressivo”, que inclui a possibilidade de representar modelos com domínios tridimensionais, embora essa potencialidade não seja explorada em seu trabalho.

[Cignoni98] descreve o “modelo simplicial multiresolução”, proposto por seu grupo de pesquisa como uma infra-estrutura unificada para modelos multiresolução. A contribuição de seu trabalho é uma estrutura de dados para representação de tal modelo como um complexo simplicial em dimensão superior. Essa estrutura facilitou a construção de um algoritmo para extrair modelos simplificados com resolução variável, que incluem as relações de vizinhança entre tetraedros.

Finalmente, [Zhou97] apresenta uma infra-estrutura de multiresolução para conjuntos de dados regulares, baseada em subdivisão simplicial. Sua proposta é muito semelhante à triangulação retangular adaptativa (capítulo 3). Por ser voltada para uma aplicação particular, no entanto, ela incorpora algumas simplificações e limitações. Esse artigo voltará a ser citado e analisado em suas particularidades no capítulo 4.

No próximo capítulo será proposto um novo modelo, baseado nas triangulações retangulares adaptativas, mais econômico nos requisitos de memória e mais flexível, no que se refere à gama de aplicações que alcança.

2.4 Modelagem de Dados Irregulares

Como mencionado anteriormente, um requisito importante para análise e geração de imagens a partir de uma amostra discreta de dados escalares é a definição de uma função que represente essa amostra e permita estimar valores para a grandeza em estudo em qualquer ponto do domínio. No contexto da visualização, essa necessidade se apresenta basicamente em duas circunstâncias: quando a amostra conhecida tem uma distribuição irregular e a ferramenta de visualização disponível exige que os dados estejam arranjados regularmente. Nesse caso, aquela função auxiliar será usada para estimar o valor escalar nos nós de uma malha regular. O segundo momento é o da aplicação da técnica de visualização propriamente dita que, seja regular ou irregular a

organização dos dados, sempre envolve algum estágio em que se faz necessária a avaliação da grandeza em foco em pontos no interior do domínio.

Levando em conta a organização de um sistema computacional de visualização de dados, pode-se constatar que a reamostragem em malha regular, quando é feita, é uma tarefa realizada uma única vez em uma seção de visualização, sendo o seu resultado, a estrutura regular auxiliar, usado repetidas vezes em um processo interativo de exploração dos dados através da geração de uma série de imagens. Sendo assim, aquele trabalho de reamostragem tem um peso relativamente pequeno no processo como um todo, do ponto de vista do tempo de execução. Isso permite aplicar métodos de interpolação mais sofisticados nessa fase inicial, sem causar um impacto muito significativo no desempenho do sistema. Consegue-se, assim, melhorar a confiabilidade dos dados atribuídos aos nós da estrutura auxiliar, que serão a base de todas as operações subseqüentes. Por outro lado, as técnicas de visualização propriamente ditas devem empregar métodos de interpolação computacionalmente mais leves, para que possam ser usados intensivamente, sem degradar o desempenho e conseqüentemente o caráter interativo do sistema.

Embora existam vários métodos de interpolação aplicáveis a muitos tipos de conjuntos de dados, não existe nenhum que possa ser indicado como o melhor para todos os casos. A condição ideal seria a de poder experimentar vários métodos e comparar as imagens geradas com cada um deles.

2.4.1 Interpolação para Técnicas de Visualização

Esta seção aborda alguns métodos de interpolação simples e de baixo custo computacional, conforme requisito estabelecido acima para implementação eficiente das técnicas de visualização.

A função interpoladora mais simples é a função “valor mais próximo”, que associa a cada ponto do domínio o valor do ponto amostral mais próximo. Apesar da descontinuidade que a caracteriza, essa função é usada com sucesso em muitas aplicações que apresentam uma amostragem regular suficientemente fina. Em uma grade regular o volume de valor constante em torno de um nó da malha é um hexaedro, denominado *voxel*. A regularidade de tamanho das células, bem como o alinhamento de suas arestas com os eixos do sistema cartesiano de referência são condições muito convenientes, proporcionando facilidade e eficiência na determinação do valor mais próximo de um dado ponto neste tipo de malha.

Em uma distribuição irregular de pontos, os volumes de valores constantes são definidos pelo Diagrama de Voronoi do conjunto [Preparata85] [O’Rourke94]. Essa estrutura não oferece a mesma facilidade para determinação do valor mais próximo, o que dificulta sua aplicabilidade.

A segunda classe de funções interpoladoras, em uma escala crescente de complexidade, abrange a função trilinear para as grades regulares e a função linear por partes para as malhas simpliciais (triangulações), sejam regulares ou irregulares (ou ainda quase-regulares, como uma triangulação RA). Esses casos já se caracterizam por

apresentarem continuidade das funções, embora as derivadas não sejam contínuas nas transições entre células.

2.4.2 Reamostragem em Malhas Auxiliares

A geração de imagens a partir de dados irregularmente distribuídos é uma área de investigação considerada ainda em sua infância por importantes pesquisadores do setor [Kaufman94] [Nielson94]. Apesar do grande número de trabalhos publicados sobre o assunto nos últimos anos, pode-se considerar ainda válida essa afirmativa. Um fato que corrobora essa opinião é a persistência da aplicação do primeiro recurso de que se lançou mão, nos primórdios da utilização de computadores para geração de imagens (mapas de isolinhas) a partir de dados científicos: ainda hoje se usa reamostrar os dados irregulares em uma malha regular para aplicação de técnicas convencionais de visualização.

As vantagens de gerar uma configuração regular para representar dados irregulares são bem conhecidas. Em geral as técnicas que trabalham com discretização do domínio são implementadas de forma mais econômica (espaço e tempo) quando as células dessa discretização têm formato constante. Os ganhos são ainda maiores se essas células são cubos com arestas paralelas aos eixos coordenados (grades cartesianas).

Uma primeira desvantagem dessa abordagem é o fato de que os dados originais são deixados de lado após a reamostragem. A função construída sobre a organização regular para as operações subseqüentes não interpolará aqueles dados. Outra desvantagem surge quando essa organização auxiliar é usada para representar dados cuja distribuição tem características de heterogeneidade espacial acentuadas. Nesses casos a regularidade da estrutura leva a um impasse: o alto grau de refinamento exigido nas regiões críticas induz um refinamento excessivo no restante do domínio e mesmo fora do domínio estrito do problema. As malhas adaptativas surgem como sendo a solução correta para essas situações. Dentre elas, as triangulações se destacam como as mais usadas, pela simplicidade de suas células, sua regularidade topológica e flexibilidade adaptativa.

Em particular, as triangulações retangulares se apresentam com o potencial de conciliar adaptatividade com um significativo caráter de regularidade (quase-regularidade). Este trabalho explora as possibilidades dessa classe de triangulações no que concerne às técnicas de visualização. Pode-se vislumbrar como aplicações, a visualização de resultados de simulações feitas sobre a própria triangulação RA, bem como sua utilização como estrutura auxiliar para visualização de dados irregulares, substituindo as malhas regulares, quando o requisito de adaptatividade for relevante. Esta substituição é uma solução de compromisso entre eficiência e qualidade de representação, conveniente para muitas aplicações.

2.4.3 Interpolação para Reamostragem em Malhas Auxiliares

Para realizar esta tarefa é necessário construir uma função interpoladora definida em todo o domínio do problema, a qual é, então, avaliada nos nós da malha auxiliar adotada. Também aqui pode ser usada uma função linear por partes. No entanto, como

esta reamostragem constitui uma fase de pré-processamento, para os algoritmos de visualização propriamente ditos, e tendo um impacto relativamente pequeno no tempo despendido no processo como um todo, é possível aplicar métodos de interpolação mais refinados e mais exigentes quanto aos recursos computacionais necessários.

A literatura dessa área dispõe de uma grande variedade de métodos, com características e conveniências também muito variadas. [Farin86] [Cendes87] [Alfeld89] [Nielson93] [Foley93] [Neves97a] são trabalhos que descrevem muitos desses métodos e discutem suas propriedades. A escolha entre eles depende de uma análise das condições específicas de cada conjunto de dados, envolvendo aspectos como continuidade da função e de suas derivadas, localidade de funções de base, exigência ou não de subdivisão do domínio (por exemplo, funções polinomiais por partes), etc. Preferencialmente, o método de interpolação mais conveniente deveria ser indicado pelos próprios especialistas responsáveis pela geração dos dados.

Capítulo 3

Triangulações Retangulares

3.1 O Que é uma Triangulação Retangular?

Antes de tudo, uma triangulação retangular é uma triangulação, o que indica a conveniência de iniciar este capítulo definindo essa classe de estruturas.

Definição: Uma triangulação de um domínio D do espaço n -dimensional é uma coleção de simplexos de dimensões menores ou iguais a n , tal que:

- a) as faces de qualquer dos simplexos da coleção fazem parte dela;
- b) os simplexos possuem interiores disjuntos;
- c) a interseção de quaisquer dois simplexos é vazia ou é um simplexo da coleção;
- d) a união dos simplexos da coleção é igual a D .

No presente contexto, é importante ressaltar que as estruturas que atendem a essa definição, possuem a propriedade de *conformidade*, ou seja, todo simplexo de dimensão $k-1$ da coleção constitui uma face (inteira) de um simplexo de dimensão k da coleção. Isso equivale a dizer que não existe um vértice no interior de uma aresta, nem uma aresta no interior de um triângulo, etc. No contexto considerado aqui, o domínio considerado pertence ao espaço tridimensional; embora freqüentes, as referências a domínios bidimensionais são usadas apenas como apelo visual.

O termo *triangulação retangular* será usado para designar uma subdivisão simplicial de um domínio constituído de um arranjo regular de cubos, subdivisão essa construída segundo uma particular metodologia. Essa metodologia será descrita a seguir, paralelamente com a identificação de algumas propriedades da triangulação resultante.

Cada cubo pode ser dividido em seis tetraedros, cortando-o por três planos que se interceptam mutuamente segundo a reta suporte de uma diagonal do cubo (figura 3.1). Cada um desses planos fica definido por essa diagonal em conjunto com uma aresta do cubo ou com uma diagonal de face.

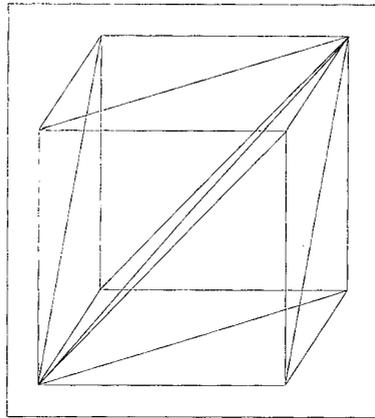


Figura 3.1 – Cubo dividido em 6 tetraedros

A escolha dos planos de corte para cada cubo deve ser feita de forma a que dois cubos vizinhos sejam imagem especular um do outro (não considerando, por enquanto, regiões do domínio com diferentes níveis de refinamento), como mostra a figura 3.2.

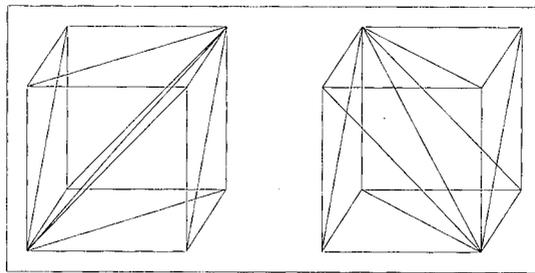


Figura 3.2 – 2 cubos vizinhos divididos em tetraedros

Os tetraedros assim formados são similares, ou seja, quaisquer dois relacionam-se por uma *similitude* (transformação afim resultante da composição de rotações, escalas uniformes, reflexões e translações).

Como a definição dos planos de corte para um cubo, nesse procedimento, está atrelada a alguma de suas diagonais, e como um cubo é sempre imagem especular de cada um de seus vizinhos, observa-se que os quatro cubos que compartilham uma mesma aresta têm (as quatro) diagonais diferentes definindo seus planos de corte.

Uma triangulação formada por um arranjo regular de cubos divididos dessa forma é chamada *triangulação CFK* (Coxeter-Freudenthal-Kuhn) em [Gomes89] e [Persiano93], onde se encontram descrições mais detalhadas. Esses trabalhos distinguem dois tipos de triangulações CFK: o tipo J1 corresponde à descrição acima; um procedimento análogo produz o tipo K1, cuja diferença do primeiro está no fato de que todos os cubos são subdivididos segundo a mesma diagonal. Pode-se considerar que, neste segundo tipo, todas as triangulações dos cubos do arranjo sejam obtidas a partir da de um deles, simplesmente por translação. Devido a suas propriedades combinatórias e geométricas, essa forma de subdivisão do espaço vem sendo utilizada em diferentes contextos e aplicações ([Allgower87], [Castelo90], [Persiano91], [Salim91], [Persiano93]).

Neste trabalho consideram-se apenas as triangulações CFK do tipo J1. Os tetraedros constituintes dessas estruturas são designados como tetraedros *tipo A*. Cada tetraedro desse tipo tem como maior aresta a diagonal do cubo e como menor aresta uma aresta do cubo.

A partir de uma tal triangulação, pode-se derivar uma outra pela aplicação de um procedimento de refinamento a cada um dos simplexos da configuração original. Esse procedimento, denominado *bisseção-ao-meio*, consiste em dividir o simplexo pelo plano definido por três pontos particulares: o ponto médio de sua maior aresta e os pontos extremos da aresta oposta. Os tetraedros resultantes desse processo são designados como tetraedros *tipo B*. Um tetraedro do tipo B tem como maior aresta uma diagonal de face do cubo e como menor aresta a metade da diagonal do cubo. A figura 3.4 mostra a bisseção-ao-meio dos seis tetraedros, agrupados em pares que apresentam o mesmo plano de corte. Vale observar que esses planos são ortogonais aos planos que subdividiram o cubo inicialmente. Note-se ainda que todos os tetraedros resultantes são congruentes entre si.

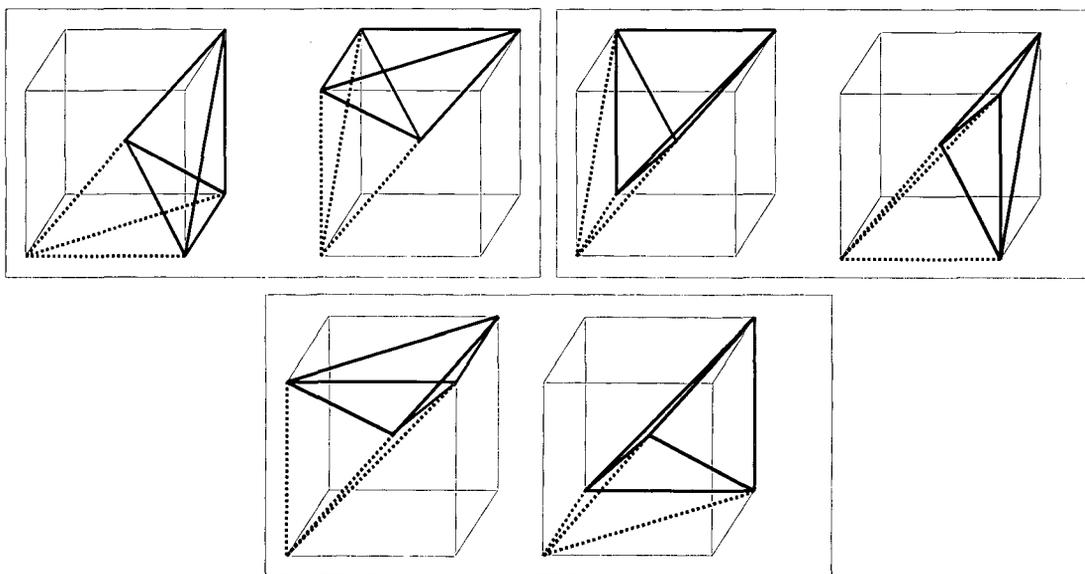


Figura 3.4 – Tetraedros do tipo A divididos em tetraedros do tipo B

Reaplicando o mesmo procedimento de bisseção-ao-meio aos novos simplexos, resulta uma terceira triangulação, cujos tetraedros, diferentes dos anteriores, são também todos congruentes entre si e recebem a designação de tetraedros *tipo C*. A figura 3.5 mostra dois exemplos dessas subdivisões. A observação cuidadosa da figura 3.4 leva a concluir que os planos de corte são os mesmos usados na fase anterior, para a subdivisão dos tetraedros tipo A. Os tetraedros tipo C têm como maior aresta uma aresta do cubo e como menor aresta a metade da diagonal da face do cubo.

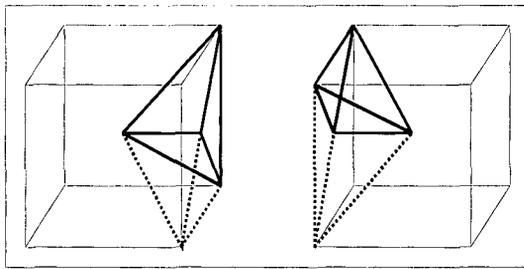


Figura 3.5 – Tetraedros do tipo B divididos em tetraedros do tipo C

Continuando com a aplicação do procedimento de bissecção-ao-meio, o próximo passo levará, pela subdivisão dos tetraedros tipo C, a uma triangulação com as mesmas características da estrutura original, formada por tetraedros tipo A. Os planos de corte são agora paralelos aos planos das faces dos cubos, formando novos cubos cujas dimensões lineares são a metade das dos cubos originais. Dessa forma, os comprimentos das arestas dos novos tetraedros tipo A, são também a metade dos comprimentos das respectivas arestas de seus antecessores do mesmo tipo. A mesma proporção se repete a cada nova geração de tetraedros dos tipos B e C, em relação a seus antecessores imediatos do mesmo tipo. Vale ressaltar ainda que, para esses cubos menores, continua valendo a propriedade de especularidade entre vizinhos, de forma que para os cubos menores, resultantes de um mesmo cubo original, as diagonais que caracterizam suas subdivisões têm diferentes orientações.

A estrutura resultante desse processo de refinamento tem um caráter hierárquico inerente, como ilustra a figura 3.6. Essa propriedade será analisada adiante, sob a ótica de multiresolução.

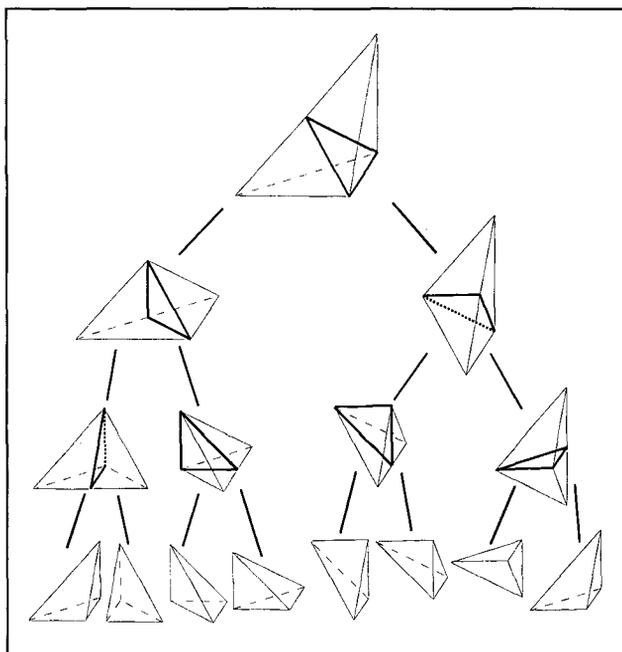


Figura 3.6 – Árvore binária de subdivisão de um simplexo

Essas triangulações são ditas “*retangulares*” pela predominância de ângulos retos entre suas faces.

3.2 Rotulação de Vértices e Faces

O estabelecimento de uma sistemática de identificação para vértices e faces dos tetraedros, favorece amplamente as discussões que se seguem. Embora seja uma decisão arbitrária, as opções feitas para essa finalidade têm reflexos importantes na maior ou menor facilidade com que se faz referência aos elementos da estrutura, tanto do ponto de vista descritivo quanto de implementação. A metodologia adotada, cuja descrição é o objeto desta seção, mostrou-se conveniente em ambos os aspectos. A figura 3.7 mostra exemplos de células com vértices rotulados segundo essa metodologia, regida pelas seguintes normas:

- i) na triangulação correspondente ao primeiro nível de subdivisão dos cubos, as arestas que unem os vértices 0 e 1, 1 e 2, 2 e 3 são arestas do cubos;
- ii) a maior aresta do tetraedro é sempre a que une os vértices 0 e 3;
- iii) os vértices de índice 3 nos novos tetraedros, gerados por uma subdivisão, são os extremos da maior aresta do tetraedro pai;
- iv) o novo vértice criado em uma subdivisão recebe o índice 2.

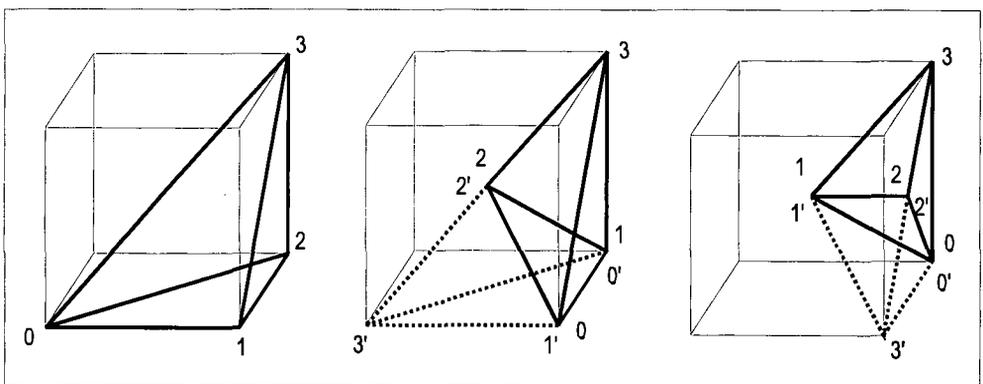


Figura 3.7 – Indexação de vértices de tetraedros

Quanto à indexação de faces, a opção mais conveniente para células simpliciais é associar a cada face o índice do vértice oposto a ela, que é o único vértice do simplexo que não pertence à face. Esta é uma solução econômica que, por identificar implicitamente as faces, evita o consumo de espaço com a referência explícita aos seus vértices.

São relacionadas a seguir algumas propriedades da indexação obtida, as quais facilitam as operações de manipulação da topologia da estrutura:

- i) na subdivisão de um tetraedro, as faces cortadas são as de índices 1 e 2;

- ii) a nova face criada por uma subdivisão tem sempre índice 3;
- iii) as faces de índices 0 e 3 de um tetraedro têm sempre o mesmo formato.

3.3 Aspectos Combinatórios de Triangulações Retangulares

Para iniciar esta exposição são necessárias mais algumas definições. A *estrela* de uma aresta é o conjunto de todos os simplexos da triangulação que compartilham aquela aresta. A estrela de uma aresta é um *núcleo* quando essa aresta é a maior aresta de todos os simplexos de sua estrela. Finalmente, a maior aresta de um núcleo é designada por *eixo* do núcleo.

Até aqui, uma triangulação retangular é uma estrutura regular, uma vez que todos os seus simplexos são do mesmo tipo e têm o mesmo tamanho, além de estarem dispostos segundo um padrão regular. Todos os simplexos de uma tal triangulação pertencem a algum núcleo, ou seja, essas triangulações são formadas apenas por núcleos. O refinamento de toda a estrutura através do procedimento de bisseção-ao-meio, como descrito acima, pode ser visto como uma série de operações de *subdivisão de núcleo* (bisseção-ao-meio dos tetraedros de um núcleo). Essa será considerada a operação básica de refinamento de uma triangulação retangular. A seção 3.4 mostrará que ela pode ser aplicada a cada núcleo isoladamente, o que proporcionará maior flexibilidade a essas estruturas, incorporando a noção de adaptatividade.

Como todos os simplexos de um núcleo são de um mesmo tipo (A, B ou C), é natural a referência ao tipo de um núcleo. Pode-se determinar quantas e quais são as situações em que os tetraedros podem ser encontrados em uma triangulação retangular, analisando os três tipos de núcleos e as diferentes direções que os seus eixos apresentam. O núcleo do tipo A é formado por 6 tetraedros que compartilham uma das diagonais do cubo. Como cada um desses tetraedros tem uma orientação própria e o cubo possui 4 diagonais, conclui-se que existem 24 orientações para tetraedros do tipo A. A figura 3.8 mostra um desses núcleos.

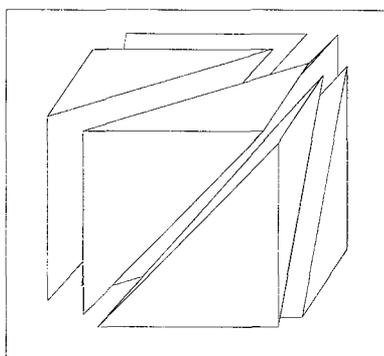


Figura 3.8 – tetraedros de um núcleo do tipo A

Um núcleo do tipo B é constituído por 4 tetraedros pertencentes a 2 cubos vizinhos, formando um octaedro. As células de um núcleo desse tipo compartilham uma das diagonais de face do cubo. Como o cubo tem 6 direções possíveis para suas diagonais

de face, resulta que são também 24 as orientações possíveis para tetraedros do tipo B. A figura 3.9 mostra os componentes de um desses núcleos.

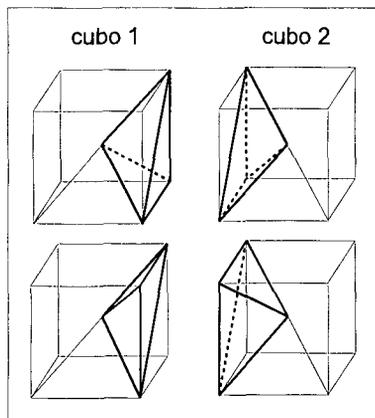


Figura 3.9 – Tetraedros de um núcleo do tipo B, pertencentes a 2 cubos vizinhos

O núcleo do tipo C também forma um octaedro e é constituído por 8 tetraedros, provenientes de quatro cubos vizinhos, que compartilham a aresta comum a esses cubos. Como as arestas de cubos têm apenas três direções possíveis, conclui-se que são também 24 as orientações de tetraedros desse tipo. A figura 3.10 mostra 4 dos 8 tetraedros de um núcleo do tipo C.

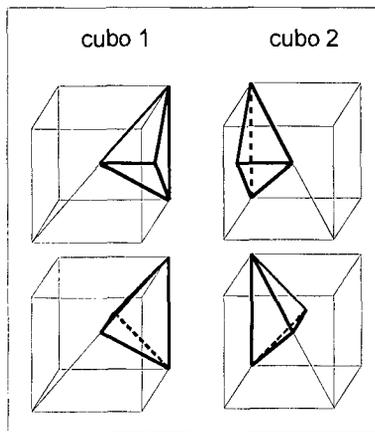


Figura 3.10 – 4 tetraedros de um núcleo do tipo C, pertencentes a 2 cubos vizinhos. Os outros 4 pertencem a 2 outros cubos

O termo “orientação” foi usado até aqui de maneira informal. No entanto, ele irá assumir um papel fundamental na organização das triangulações retangulares, o que justifica associá-lo a um significado preciso. Pode-se definir *orientação de um tetraedro* como sendo a lista dos valores dos ângulos que suas faces fazem com os planos coordenados.

Da análise acima, conclui-se pela existência de 72 orientações de tetraedros distintas em triangulações retangulares, sendo 24 para cada tipo de tetraedro. A

metodologia de indexação adotada permite coexistirem em uma mesma estrutura tetraedros com a mesma orientação mas que apresentam duas distintas indexações de faces e vértices, que correspondem à inversão dos índices associados aos vértices da maior aresta do tetraedro. Isto levou a cunhar o termo *orientação indexada*, para diferenciar as duas instâncias de cada orientação de tetraedro. Assim sendo, existem 48 orientações indexadas para cada tipo de tetraedro, totalizando 144. No entanto, o contexto apenas raramente exige essa diferenciação, o que permite continuar usando o termo orientação tanto para fazer referência às 72 orientações de tetraedros quanto às 144 orientações indexadas, explicitando a diferença apenas diante de alguma necessidade eventual.

Cada uma das diversas orientações é obtida pela aplicação de uma *transformação de re-orientação* a um tetraedro padrão do mesmo tipo. Define-se uma transformação dessa classe por uma ou duas reflexões relativas a um dos 9 planos que contêm 4 vértices de um cubo da subdivisão original. Esses planos são: os 3 planos coordenados

$$x=0, y=0 \text{ ou } z=0$$

e os 6 planos inclinados

$$x=y, -x=y, x=z, -x=z, y=z \text{ ou } -y=z.$$

A orientação de um tetraedro pode ser caracterizada por um vetor. Para isso, basta tomar-se qualquer vetor definido pela geometria do tetraedro e pela indexação de seus vértices e faces, desde que as três coordenadas do vetor sejam distintas em módulo. Foi escolhido como vetor de orientação (V_{ori}), o vetor diferença entre o vértice de índice 3 e o baricentro da face oposta a ele. A figura 3.11 ilustra esse vetor para um tetraedro de cada tipo.

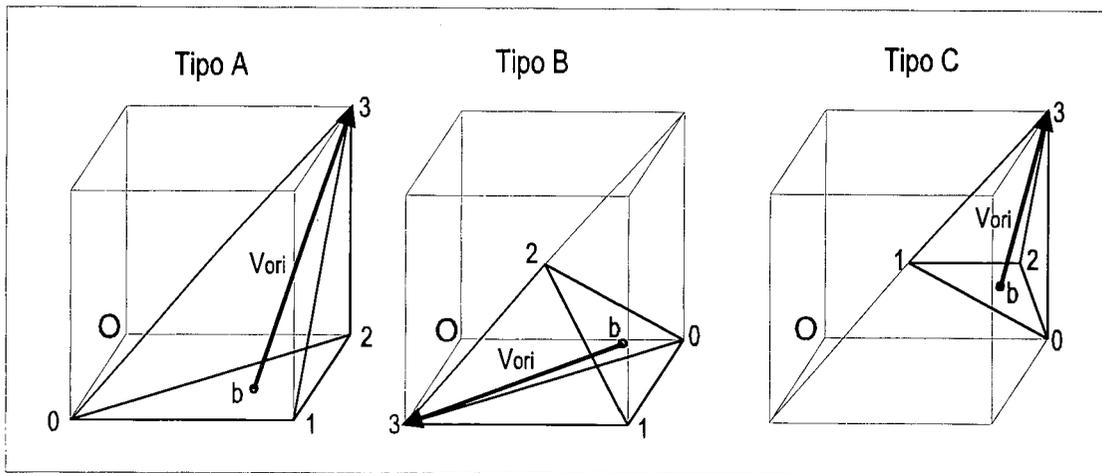


Figura 3.11 – Vetores de orientação de tetraedros.

Para esses casos, tem-se:

tipo A: $v_0 = (1, 0, 0)$, $v_1 = (1, 1, 0)$, $v_2 = (0, 1, 0)$ e $v_3 = (0, 1, 1)$

$$b = (v_0 + v_1 + v_2) / 3 = (2/3, 2/3, 0)$$

$$V_{ori} = v_3 - b = (-2/3, 1/3, 1)$$

tipo B: $v_0 = (0, 1, 0)$, $v_1 = (1, 1, 0)$, $v_2 = (1/2, 1/2, 1/2)$ e $v_3 = (1, 0, 0)$

$$b = (v_0 + v_1 + v_2) / 3 = (3/6, 5/6, 1/6)$$

$$V_{ori} = v_3 - b = (3/6, -5/6, -1/6)$$

tipo C: $v_0 = (0, 1, 0)$, $v_1 = (1/2, 1/2, 1/2)$, $v_2 = (1/2, 1, 1/2)$ e $v_3 = (0, 1, 1)$

$$b = (v_0 + v_1 + v_2) / 3 = (2/6, 5/6, 2/6)$$

$$V_{ori} = v_3 - b = (-2/6, 1/6, 4/6)$$

Apenas a direção do vetor é importante, sendo irrelevante o seu módulo. Portanto, pode-se assumir que, para os tetraedros da figura, os vetores de orientação sejam escalas positivas dos obtidos acima. Por exemplo, pode-se adotar os vetores $(-2, 1, 3)$, $(3, -5, -1)$ e $(-2, 1, 4)$, respectivamente. Conforme mencionado anteriormente, as demais orientações de tetraedros podem ser obtidas através de transformações de reorientação aplicadas sobre tetraedros padrão. Qualquer tetraedro de cada tipo pode ser tomado como padrão, inclusive aqueles utilizados na ilustração. Uma transformação de reorientação é uma composição de reflexões em relação aos planos já identificados. Observa-se, ainda, que a reflexão de um vetor em relação a um plano coordenado equivale à troca do sinal de uma de suas componentes, e que a reflexão em relação a um dos planos inclinados equivale à permutação de duas de suas coordenadas, com a eventual troca de sinal de uma delas. Portanto, nenhuma dessas transformações altera os módulos das coordenadas do vetor. Conclui-se que os vetores de orientação podem ser tomados como sendo vetores que têm como componentes os valores seguintes, em alguma ordem:

$(+/-)1$, $(+/-)2$ e $(+/-)3$, para tetraedros do tipo A;

$(+/-)1$, $(+/-)3$ e $(+/-)5$, para tetraedros do tipo B; e

$(+/-)1$, $(+/-)2$ e $(+/-)4$, para tetraedros do tipo C.

Assim sendo, por ação de combinações de reflexões, o vetor de orientação de um tetraedro poderá alterar-se no máximo em 47 outros vetores, o que caracteriza as 48 orientações de cada tipo de tetraedro. De fato, como as três coordenadas do vetor possuem módulos distintos, sob a ação das reflexões, há 8 possíveis combinações de alteração de sinais e 6 possíveis permutações dos valores distintos dos módulos das coordenadas (ver seção 3.7.1).

3.4 Triangulação Retangular Adaptativa

A seção 3.1 descreveu um processo de subdivisões sucessivas, aplicado de maneira uniforme sobre as células da estrutura. No entanto, em muitas situações essa uniformidade é inconveniente, sendo necessário permitir que essas subdivisões sejam controladas por critérios variáveis para diferentes regiões do domínio, de acordo com requisitos específicos da aplicação. Assim, pode-se chegar à conclusão de que uma determinada célula tem dimensões muito grandes, ou seja, a aplicação pode exigir que a região do espaço onde ela se situa tenha uma subdivisão mais refinada, dando gênese a uma *malha adaptativa*.

Para gerar configurações desse tipo, surge o problema de que o procedimento de bisseção-ao-meio não pode ser aplicado a um tetraedro isoladamente. Essa impossibilidade se deve ao fato de que a maior aresta de um tetraedro, aquela que será subdividida pela inserção de um novo vértice, sempre será compartilhada por outra(s) célula(s). Por estarem sendo utilizadas triangulações bidimensionais como ilustrações, convém observar que nesses casos existem exceções àquela regra: um triângulo que tenha sua maior aresta na fronteira do domínio é considerado um *núcleo degenerado*, e pode ser subdividido isoladamente. A menos desta exceção, a configuração topológica resultante da subdivisão de um único simplexo da estrutura não caracteriza uma triangulação, por não apresentar a propriedade de conformidade, definida no início deste capítulo, uma vez que o novo vértice inserido não é vértice das células vizinhas. Isso causa severas restrições em determinadas aplicações, tais como geração de malhas para elementos finitos e aproximação polinomial de funções, tornando-se necessário resolver essa questão topológica de forma a que a estrutura final seja uma triangulação. A solução proposta em [Barbalho93] é estabelecer, como única operação de refinamento possível, a aplicação do procedimento de bisseção-ao-meio simultaneamente a todos os tetraedros pertencentes a um mesmo núcleo. Essa operação conjunta, designada por *subdivisão de núcleo*, sempre resulta em uma triangulação.

Define-se *triangulação retangular adaptativa*, ou *triangulação RA*, como a estrutura resultante de uma seqüência de subdivisões de núcleos, aplicada a uma triangulação retangular regular, onde a escolha dos núcleos a serem subdivididos tenha sido induzida por algum critério de refinamento não homogêneo. Uma estrutura desse tipo é constituída, a princípio, por uma composição dos três tipos de tetraedros, podendo incluir instâncias das 72 orientações possíveis já relacionadas, com diferentes tamanhos.

A figura 3.12 mostra uma triangulação RA bidimensional. Observa-se que, diferente de uma triangulação retangular regular, a estrutura apresenta simplexos que não pertencem a núcleos. Para subdividir um desses simplexos, por exemplo o simplexo hachurado da ilustração, é necessário formar primeiramente o seu núcleo, o que exige a subdivisão de outras células da vizinhança.

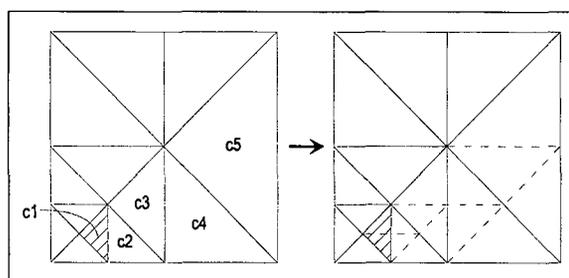


Figura 3.12 – Triangulação RA bidimensional

A figura mostra que esse efeito se estende até que seja alcançado um núcleo preexistente: seja *c1* a célula hachurada, *c2* sua vizinha, com quem *c1* compartilha sua maior aresta, e *c3* a vizinha de *c2* pela maior aresta; *c4* e *c5* completam a seqüência de células alcançadas por esse efeito indutivo, terminando em *c5*, que no exemplo é a maior célula da estrutura. *c5* pertence a um núcleo já formado (neste caso um núcleo degenerado, portanto constituído de uma única célula), podendo, portanto, ser

subdividido, e interrompendo a expansão da região de abrangência do processo. O procedimento para determinar essa vizinhança, bem como a seqüência de refinamentos de núcleos necessários, é designado por *refinamento básico*. Esse procedimento, apresentado como o algoritmo 3.1, foi extraído de [Barbalho93].

Procedimento de refinamento básico de um simplexo S :

Refinamento_básico (S)

A ← aresta mais longa de S

enquanto A não é a maior aresta de algum simplexo S' da estrela de A

 Refinamento_básico (S')

 efetuar a subdivisão do núcleo de S

Algoritmo 3.1 – Refinamento básico

A fase de avanço recursivo desse algoritmo determina a região que será abrangida pelo refinamento; na fase de retorno é que a subdivisão dos núcleos é efetivada, pela bisseção-ao-meio de seus simplexos. As arestas pontilhadas na figura 3.12 são aquelas incluídas pelo processo de refinamento relativo a S. A ordem dessas inclusões é da maior para a menor aresta.

A exposição do procedimento de refinamento básico será encerrada com a análise de algumas de suas propriedades. Antes, porém, faz-se necessário incluir uma nova definição, dentro do contexto das triangulações RA. A *bacia de um núcleo* [Bueno96] é o conjunto de células com a seguinte propriedade comum: a requisição de subdivisão da célula desencadeia, pelo mecanismo indutivo exposto acima, uma série de requisições de subdivisões de outras células, culminando na subdivisão do núcleo em questão. A figura 3.13 mostra uma triangulação RA onde estão marcadas com letras maiúsculas as células componentes de núcleos e com letras minúsculas as integrantes das respectivas bacias.

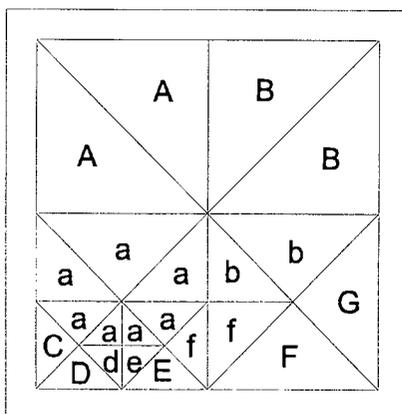


Figura 3.13 – Bacias em uma triangulação RA.

Finalmente, são ressaltadas as seguintes propriedades do procedimento acima.

i) O algoritmo é finito

No aprofundamento do processo recursivo, o refinamento básico é aplicado sobre um novo simplexo, que compartilha a maior aresta de S , mas que possui outra aresta ainda maior. Portanto, cada nova chamada desse procedimento tem como argumento uma célula maior do que a anterior, o que garante a inexistência de ciclos e a finitude do algoritmo, uma vez que a estrutura é finita. Mais precisamente, em triangulações tridimensionais o tamanho da maior aresta dobra a cada três chamadas recursivas, quando é alcançado outro tetraedro do mesmo tipo do primeiro. Isso caracteriza uma progressão geométrica naturalmente finita. No pior caso, quando o processo se estende até o nível mais alto de uma árvore, o nível alcançado pela recursão é $n_{início}-1$, onde $n_{início}$ é o nível de refinamento da célula que deflagrou o processo.

ii) O procedimento é invariante à ordem de sua aplicação

Denotando-se por $B(S, T)$ a estrutura resultante do refinamento básico do simplexo S da triangulação T , então, para quaisquer simplexos $S1$ e $S2$ de T as triangulações $B(S2, B(S1, T))$ e $B(S1, B(S2, T))$ são iguais.

Para confirmar esta assertiva, devem-se considerar duas situações possíveis: se $S1$ e $S2$ pertencem ou não à mesma bacia. Se pertencem, existe interseção entre os conjuntos de núcleos a serem subdivididos nos refinamentos básicos de $S1$ e $S2$. A interseção mínima é o maior núcleo de cada um desses conjuntos. A subdivisão dos núcleos pertencentes a essa interseção, no refinamento básico de $S1$, por exemplo, refletirá no refinamento básico de $S2$ apenas como uma simplificação, truncando a seqüência de núcleos a serem subdivididos. Portanto, a ordem de execução desses refinamentos não altera o resultado final.

Se $S1$ e $S2$ não pertencem à mesma bacia, seus refinamentos básicos são totalmente independentes, do ponto de vista dos núcleos envolvidos, e novamente sua ordem de execução é irrelevante.

Encerrando esta seção, algumas propriedades importantes das triangulações RA são também explicitadas:

i) qualquer simplexo de um determinado tipo pode ser obtido por uma transformação afim de qualquer outro simplexo do mesmo tipo. As transformações necessárias são: translação, escala e reorientação, esta última definida como uma ou duas reflexões relativas a um dos 9 planos que contêm 4 vértices de um cubo da subdivisão original;

ii) os fatores de escala envolvidos nessas transformações são sempre potências de 2, da forma 2^k , onde $k \in \mathbf{Z}$;

iii) para cada tipo de simplexo existem 24 transformações de reorientação distintas.

3.4.1 Organização Hierárquica

O resultado do processo de refinamento de um tetraedro, mostrado na figura 3.6, constitui uma árvore BSP (binary space partition tree), ou seja, uma árvore binária em que cada nó corresponde a uma região do espaço, que pode ser ou não subdividida em

duas sub-regiões. Nesse caso a subdivisão é definida por um plano, conforme estabelecido anteriormente.

Sendo assim, uma triangulação RA pode ser vista como um arranjo de árvores BSP. Será chamada, daqui para a frente, *triangulação raiz*, a triangulação original, formada por tetraedros do tipo A, a partir da qual se obteve uma triangulação RA pela aplicação de procedimentos de refinamento adaptativo. Cada simplexo da triangulação raiz é a raiz de uma árvore BSP que descreve todas as sucessivas bisseções aplicadas ao longo do processo de refinamento. Para facilitar a discussão, é conveniente considerar duas classes de simplexos nessas estruturas: os nós internos das árvores BSP, os quais serão chamadas de *setores* da triangulação; e as *folhas* dessas árvores, que são as células da triangulação propriamente ditas.

Numa árvore BSP de uma triangulação RA qualquer, o plano bissetor de um setor é definido implicitamente pela própria geometria do simplexo. Assim sendo, torna-se desnecessário armazenar esse plano explicitamente associado ao nó, como é feito normalmente com árvores BSP de caráter mais geral.

A representação de triangulações RA como árvores BSP tem utilidade como suporte a diversos algoritmos para resolver problemas de pesquisa (por exemplo, encontrar o simplexo que contém um determinado ponto) e de percurso (por exemplo, o percurso ordenado segundo a visibilidade em uma dada direção). Será dada atenção especial, no capítulo 6, ao problema de ordenação direcional em uma triangulação RA, por ser um requisito fundamental do método de projeção para visualização de campos escalares. As árvores BSP proporcionam uma solução simples, elegante e eficiente para esse problema.

3.4.2 Construção de uma Triangulação RA

Uma triangulação adaptativa é uma forma de decomposição de um domínio, que permite atender a requisitos de refinamento da subdivisão, que variam nas diferentes regiões do domínio. Esses requisitos são dependentes da aplicação e são expressos como critérios de adaptatividade ou critérios de refinamento. A abordagem do problema de visualização de dados através de triangulações RA, pressupõe a existência de uma metodologia de construção dessas estruturas, que permita levar em conta os requisitos específicos do fenômeno em estudo. Por outro lado, é desejável que a geração dessa estrutura possa ser feita por algoritmos de aplicação geral, onde apenas o critério de refinamento esteja vinculado à particular aplicação. [Persiano93] propõe um esquema para construção de triangulações RA para problemas cujos requisitos de refinamento possam ser expressos por critérios *espacialmente decomponíveis e invariantes a inclusões*.

A exigência de que o critério seja espacialmente decomponível significa que toda a triangulação será considerada satisfatória se cada um de seus simplexos for aceito pelo critério. Assim sendo, para qualquer triangulação do domínio do problema, existe um refinamento mínimo que atende aos requisitos específicos.

O critério é invariante a inclusões se a aprovação de um simplexo implica que qualquer outro simplexo no seu interior também satisfaz ao critério.

O esquema de construção proposto consiste em definir uma triangulação retangular inicial T sobre o domínio do problema e, enquanto existir algum simplexo S em T que não atenda ao critério, executar o refinamento básico de S . Como se pode observar, esse esquema não exige nenhuma ordem especial para aplicação do refinamento básico; uma vez que um simplexo é recusado, ele é imediatamente refinado. Diferentes algoritmos realizarão os testes em ordens diferentes.

3.5 Geometria das Células

Uma análise detalhada da geometria das células de uma triangulação retangular, dará subsídios a outras inferências relativas a aspectos combinatórios e topológicos dessas estruturas. Esse estudo é facilitado pela figura 3.14, que mostra uma representação planificada de tetraedros A, B e C obtidos por uma seqüência de subdivisões. Fica evidente a existência de apenas 4 formatos diferentes de faces, designadas por α , β , δ e γ , e que as codificações das faces (formato, índice) são todas distintas, ou seja, a face α_1 , por exemplo, só existe em tetraedro do tipo A.

A tabela 3.1 relaciona esses formatos de faces, com suas respectivas dimensões, relativas à primeira ocorrência de cada um, a partir de um cubo cuja aresta tem comprimento 1:

Faces	comprimentos dos lados
α_1 e α_2	$1, \sqrt{2}, \sqrt{3}$
β_0, β_2 e β_3	$1, 1, \sqrt{2}$
δ_1	$\sqrt{3}/2, \sqrt{3}/2, \sqrt{2}$
γ_0, γ_2 e γ_3	$1, \sqrt{3}/2, \sqrt{3}/2$
α_0 e α_3	$1/2, \sqrt{2}/2, \sqrt{3}/2$
β_1	$1, \sqrt{2}/2, \sqrt{2}/2$

Tabela 3.1 – Dimensões da primeira ocorrência de cada formato de face, a partir de um cubo de lado 1.

Na k -ésima ocorrência de cada uma dessas faces, como produto de sucessivos refinamentos da triangulação, as dimensões de suas arestas serão multiplicadas por $1/2^{k-1}$.

A noção já estabelecida de descendência entre tetraedros, onde os filhos resultam da subdivisão do pai, estende-se, em parte, às faces dos tetraedros: a subdivisão de um tetraedro implica na subdivisão de algumas de suas faces, precisamente as faces de índices 1 e 2. A figura 3.15 mostra as relações de descendência entre as faces dos 3 tipos de tetraedros, admitindo-se que as de índices 1 e 2 sejam descendentes das de

índices 0 e 3, pela simples reindexação, uma vez que essas últimas não são subdivididas. Observa-se, ainda, que as faces de índice 3 não descendem de nenhuma outra, já que são faces novas, pertencentes ao plano de corte usado para subdivisão da célula que lhes deu origem.

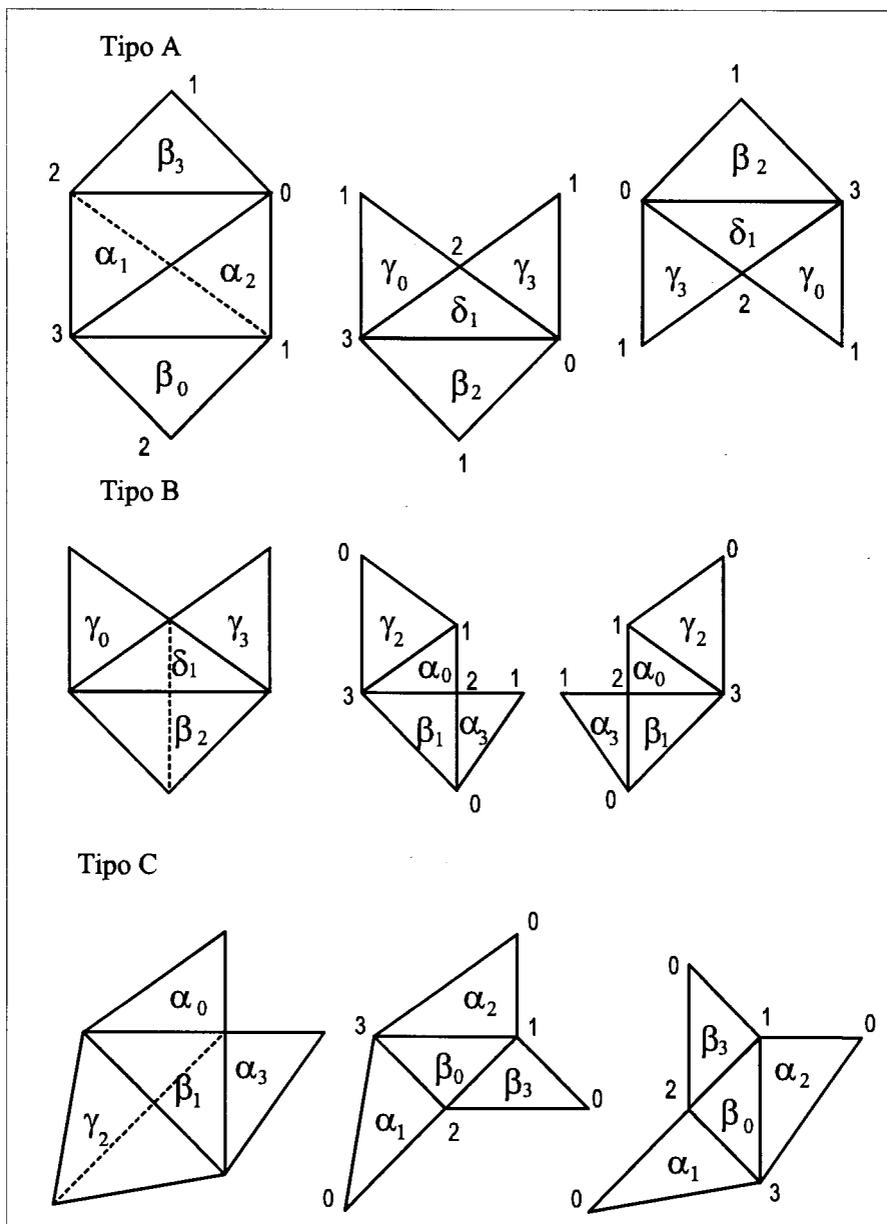


Figura 3.14 – Planificação e identificação das faces dos tetraedros. Os 4 formatos de faces são designados como α , β , δ e γ . As linhas pontilhadas indicam os cortes das operações de subdivisão.

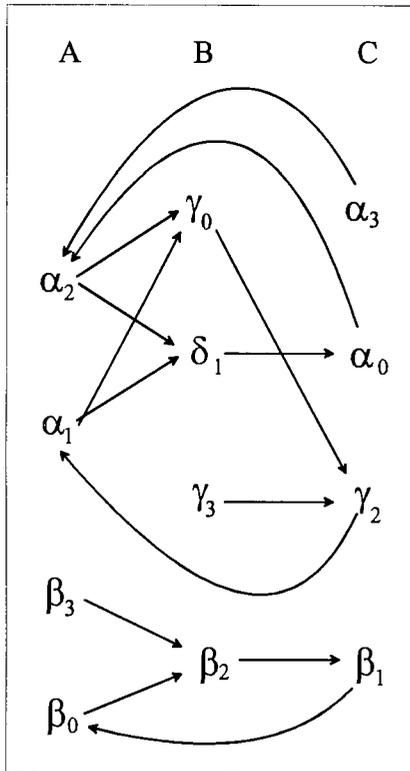


Figura 3.15 – Relações de descendência entre faces de tetraedros. Estão indicados os formatos das faces subdivididas com seus índices, bem como os formatos e índices das faces geradas.

Outro aspecto a ser ressaltado, com relação à geometria das células, é que os três tipos, A, B e C, são tetraedros relativamente “gordos”. Esta é uma característica positiva, que interessa a muitas aplicações, onde a existência de células muito delgadas pode ser uma causa de instabilidade numérica de algoritmos. A quantificação desta propriedade pode ser feita pela medida do menor ângulo de face, que é igual a aproximadamente 35 graus, ocorrendo nas faces α e δ . Outro parâmetro também associado, este de caráter tridimensional, é a medida dos ângulos sólidos internos dos vértices dos tetraedros. Os tetraedros “gordos” são os que têm esses ângulos (mínimo e médio) maiores. O caso ideal é o tetraedro regular, que tem todos os ângulos internos medindo 45 graus. No pior caso, o ângulo mínimo é nulo, caracterizando um tetraedro degenerado. A tabela 3.2 relaciona esses ângulos para os quatro vértices de cada tipo de tetraedro de uma triangulação retangular. Observa-se que o valor mínimo é de 15 graus, e ocorre nos três tipos de tetraedros, sendo a média de 35 graus, valores considerados razoáveis.

Tipo A	Tipo B	Tipo C
15	15	15
45	60	90
45	30	30
15	15	15

Tabela 3.2 – Ângulos sólidos internos dos tetraedros de uma triangulação retangular.

3.6 Relações de Vizinhança

A metodologia proposta, para identificação de vértices e faces das células de uma triangulação retangular, também oferece facilidades no que se refere à caracterização das relações de vizinhança na estrutura. A figura 3.16 é uma representação gráfica de todas as vizinhanças possíveis entre faces dos três tipos de tetraedros, tanto para estruturas regulares (homogêneas) quanto para triangulações adaptativas. Da análise dessas relações ressaltam as seguintes propriedades:

- i) uma face comum a dois tetraedros vizinhos do mesmo tipo, possui o mesmo índice em ambos;
- ii) a face de índice 1 de um tetraedro é sempre compartilhada com um vizinho do mesmo tipo;
- iii) a face de índice 2 também é compartilhada por tetraedros do mesmo tipo, ou corresponde à face 0 ou 3 de um vizinho maior;
- iv) faces de índices 0 e 3 são compartilhadas com tetraedros do mesmo tipo, ou correspondem a faces 2 de vizinhos menores.

Estas propriedades, juntamente com um trabalho de inspeção de cada caso, permitem identificar os índices que um vértice tem em dois tetraedros que compartilham uma face que contém esse vértice. A tabela 3.3 representa essas correspondências, simplificando a resposta à seguinte consulta: qual é o índice do vértice i de um tetraedro t no tetraedro t' , vizinho de t pela face j ? Esta questão surge em cada transição entre tetraedros no percurso das células atravessadas por um raio (ver método de lançamento de raios).

A notação usada na construção da tabela é a seguinte:

- . $i \leftrightarrow i'$ indica que o vértice i de t corresponde ao vértice j de t' .
- . $=$ indica que os índices dos vértices são os mesmos nos dois tetraedros. Consequentemente, os índices da face também coincidem.
- . $[j, j']$ indica os índices da face nos dois tetraedros, quando forem diferentes.

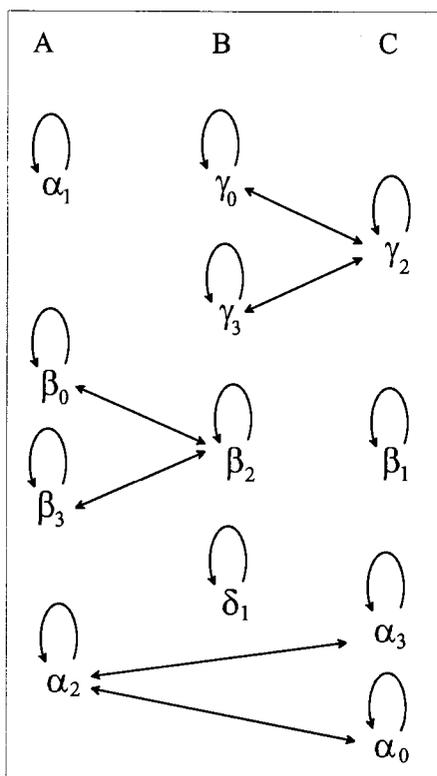


Figura 3.16 – Relações de vizinhança entre tetraedros. Estão indicados os formatos das faces compartilhadas (α , β , δ ou γ) e seus índices nos respectivos tetraedros.

$t' \rightarrow$ $t \downarrow$	A ($v_3=v_3'$)	A ($v_3 \neq v_3'$)	B ($v_3=v_3'$)	B ($v_3 \neq v_3'$)	C ($v_3=v_3'$)	C ($v_3 \neq v_3'$)
A	=	=	$1 \leftrightarrow 0$ $2 \leftrightarrow 1$ $3 \leftrightarrow 3$ [0,2]	$0 \leftrightarrow 3$ $2 \leftrightarrow 0$ $1 \leftrightarrow 1$ [3,2]	$0 \leftrightarrow 1$ $1 \leftrightarrow 2$ $3 \leftrightarrow 3$ [2,0]	$0 \leftrightarrow 1$ $1 \leftrightarrow 2$ $3 \leftrightarrow 0$ [2,3]
B	$0 \leftrightarrow 1$ $1 \leftrightarrow 2$ $3 \leftrightarrow 3$ [2,0]	$0 \leftrightarrow 2$ $1 \leftrightarrow 1$ $3 \leftrightarrow 0$ [2,3]	=	$1 \leftrightarrow 0$ $0 \leftrightarrow 1$ $2 \leftrightarrow 2$ [3,3]	$1 \leftrightarrow 0$ $2 \leftrightarrow 1$ $3 \leftrightarrow 3$ [0,2]	$1 \leftrightarrow 3$ $2 \leftrightarrow 1$ $0 \leftrightarrow 0$ [3,2]
C	$1 \leftrightarrow 0$ $2 \leftrightarrow 1$ $3 \leftrightarrow 3$ [0,2]	$1 \leftrightarrow 0$ $0 \leftrightarrow 3$ $2 \leftrightarrow 1$ [3,2]	$0 \leftrightarrow 1$ $1 \leftrightarrow 2$ $3 \leftrightarrow 3$ [2,0]	$0 \leftrightarrow 0$ $3 \leftrightarrow 1$ $1 \leftrightarrow 2$ [2,3]	($v_2=v_2'$) = =	($v_2 \neq v_2'$) $3 \leftrightarrow 0$ $0 \leftrightarrow 3$ $1 \leftrightarrow 1$ [2,2]

Tabela 3.3 – Correspondência entre os vértices da face de contato de tetraedros vizinhos. Note-se a condição particular a ser verificada quando ambos os tetraedros são do tipo C.

3.7 Catálogo de Orientações

Este catálogo consiste de uma tabela contendo a descrição de todas as características geométricas que dependam apenas da orientação do tetraedro. Essa tabela possui uma entrada para cada orientação de tetraedro, indexadas de 0 a 143. O registro relativo a uma dada orientação define um tetraedro padrão, incluindo referências a seus vértices e às normais aos planos de suas faces. Para estabelecer esses tetraedros padrão são suficientes 15 vértices definidos em relação a um cubo padrão, incluindo os 8 vértices do cubo, os centros das 6 faces e o centro do cubo, e 18 normais relativas aos 3 planos coordenados e aos 6 planos formados pelas diagonais do cubo (aos pares). As listas de coordenadas daqueles vértices e desses vetores normais, complementam o catálogo.

Embora os tetraedros padrão do catálogo possam ser definidos em função dos 15 vértices relacionados acima, por exemplo em um arquivo fixo, a forma mais elegante para se obter o mesmo resultado é definir um único tetraedro padrão de cada tipo e usar reflexões para completar a tabela. A próxima seção fornece instrumentos para isso.

3.7.1 Organização do Catálogo

Esta seção será dedicada a estudar o acesso ao catálogo de orientações. Tanto para a construção da tabela, quanto para realizar consultas a ela, faz-se necessário estabelecer uma sistemática de cálculo dos índices de suas entradas. Deve-se oferecer facilidade para atender a requisições do tipo: i) dado um índice, fornecer o tetraedro padrão correspondente; e, ii) dado um tetraedro, qual é a posição da tabela que lhe corresponde? O primeiro tipo de requisição permite montar o catálogo sem a necessidade de manter um arquivo fixo contendo as descrições dos tetraedros padrão. Será útil também na construção de outras tabelas de uso mais específico. A segunda requisição caracteriza a consulta propriamente dita, permitindo obter do catálogo, ou de tabelas complementares, as informações padronizadas relativas à orientação de cada célula de uma estrutura.

A abordagem que se segue responde primeiro ao segundo tipo de requisição.

Conforme estabelecido acima, a identificação do tetraedro padrão associado a um dado tetraedro, conhecido pelas coordenadas de seus vértices, pode ser deduzida da própria geometria do tetraedro, através de seu vetor de orientação. Como já se observou anteriormente, as coordenadas do vetor de orientação de um tetraedro do tipo A são proporcionais a $(+/-)1$, $(+/-)2$ e $(+/-)3$, em alguma ordem. Há 6 (permutação de 3) desses vetores padrão de orientação com coordenadas positivas:

$$(1,2,3) \quad (1,3,2) \quad (2,1,3) \quad (3,1,2) \quad (2,3,1) \quad (3,2,1)$$

sendo os restantes 42 obtidos trocando-se o sinal de uma ou mais coordenadas. Analogamente, há 6 vetores com coordenadas positivas relativos a tetraedros do tipo B e 6 relativos ao tipo C. A tabela 3.4 apresenta uma possível enumeração para esses vetores:

Índice de permutação (P)	Tipo A (0)	Tipo B (1)	Tipo C (2)
0	1, 2, 3	1, 3, 5	1, 2, 4
1	1, 3, 2	1, 5, 3	1, 4, 2
2	2, 3, 1	3, 5, 1	2, 4, 1
3	3, 2, 1	5, 3, 1	4, 2, 1
4	3, 1, 2	5, 1, 3	4, 1, 2
5	2, 1, 3	3, 1, 5	2, 1, 4

Tabela 3.4

Essa enumeração foi escolhida de modo que os 6 tetraedros do tipo A que subdividem o cubo inicial formem um ciclo com índices de permutações 0, 1, 2, 3, 4 e 5, nessa ordem.

A tabela 3.5 apresenta uma possível enumeração para as configurações de sinais dos vetores de orientação:

Índice de sinais (S)	Sinais
0	+, +, +
1	-, +, +
2	+, -, +
3	-, -, +
4	+, +, -
5	-, +, -
6	+, -, -
7	-, -, -

Tabela 3.5

Dado um tetraedro qualquer, o índice de sua orientação pode ser obtido por:

$$48t + 8P + S$$

onde t é o tipo do tetraedro ($A=0$, $B=1$, $C=2$), e P e S são seus índices nas tabelas 3.4 e 3.5, respectivamente. Tetraedros do tipo A terão orientações com índices entre 0 e 47, os do tipo B, entre 48 e 95, e os do tipo C, entre 96 e 143.

Exemplo:

Tetraedro tipo B:

Vértices : $v_0=(0,0,0)$ $v_1=(2,0,0)$ $v_2=(1,1,1)$ $v_3=(2,2,0)$
 vetor padrão de orientação: $(3,5,-1)$
 vetor padrão positivo: $(3,5,1)$
 vetor de sinais: $(+,+,-)$
 índice de permutação P: 2

índice de sinais S: 4
 índice da orientação: $48 + 8 * 2 + 1 = 68$

Dado um tetraedro qualquer, os seus índices de permutação P e de sinais S podem ser calculados por um procedimento (Algoritmo 3.2) que parte das coordenadas de seus quatro vértices $v0[]$, $v1[]$, $v2[]$ e $v3[]$:

```

s = 0;
j = 1;
for (i=0; i<3; i++) {
  V[i] = 3*v3[i] - v0[i] - v1[i] - v2[i];
  if (V[i] < 0) {
    s = s + j; /* s = s + 2^i */
    V[i] = - V[i];
  }
  j = j * 2;
}

/* a partir daqui, V eh positivo */
if (V[0] < V[1]) {
  if (V[1] < V[2]) p = 0;
  else if (V[0] < V[2]) p = 1;
  else p = 2;
}
else {
  if (V[1] > V[2]) p = 3;
  else if (V[0] > V[2]) p = 4;
  else p = 5;
}

```

Algoritmo 3.2

Esta, no entanto, não é a maneira mais eficiente de acessar o catálogo de orientações, uma vez que exige o cálculo do vetor de orientação a cada consulta referente a uma célula. A idéia imediata para melhorar o processo, evitando recalculá-lo a cada consulta, é armazenar o índice da orientação no registro da célula, quando ele for obtido pela primeira vez. Esta solução torna-se ainda mais atraente quando se constata que esse índice pode ser obtido sem envolver cálculos com aritmética de ponto flutuante. A próxima seção mostra como fazê-lo, recorrendo à regularidade do processo de subdivisão, gerador das triangulações retangulares, que permite deduzir as orientações dos filhos a partir da do pai.

Antes, porém, esta seção deve ser encerrada respondendo à primeira requisição estabelecida na sua abertura: dado um índice de orientação, *ind_orient*, de um tetraedro T, fornecer o tetraedro padrão correspondente.

A tabela 3.6 propõe um tetraedro padrão para cada tipo e seus correspondentes vetores de orientação.

Vetor de orientação	v0	v1	v2	v3
(1,2,3)	(0,0,0)	(2,0,0)	(2,2,0)	(2,2,2)
(1,3,5)	(2,0,0)	(2,2,0)	(1,1,1)	(2,2,2)
(1,2,4)	(2,2,0)	(1,1,1)	(2,1,1)	(2,2,2)

Tabela 3.6

A fórmula que dá o índice de orientação em função de t (tipo), P (índice de permutação) e S (índice de sinais), pode ser revertida de forma a que se obtenha t , P e S a partir do índice de orientação, ind_orient de um tetraedro T . As fórmulas para isso são:

$$t = (ind_orient / 48) \% 3;$$

$$P = (ind_orient \% 48) / 8;$$

$$S = ind_orient \% 8;$$

onde “/” e “%” são operadores de divisão inteira que fornecem o quociente e o resto, respectivamente.

De posse desses elementos, e recorrendo às tabelas 3.4 e 3.5 obtém-se o vetor de orientação de T . Seja, por exemplo, $(-4,-2,1)$ o vetor de orientação de T , tetraedro do tipo C . Comparando esse vetor com o do tetraedro padrão do tipo C da tabela 3.6, conclui-se que o tetraedro padrão associado à orientação de T pode ser obtido, a partir do tetraedro da tabela, trocando-se a primeira com a terceira coordenada e invertendo-se os sinais das duas primeiras coordenadas resultantes. Assim, o tetraedro padrão procurado terá como vértices os pontos:

$$v0 = (0, -2, 2) \quad v1 = (-1, -1, 1) \quad v2 = (-1, -1, 2) \quad v3 = (-2, -2, 2).$$

Assim, a partir de um tetraedro de cada tipo com orientação conhecida, podem-se gerar tetraedros com quaisquer outras orientações.

3.7.2 Orientação dos Filhos da Subdivisão

A determinação do índice de orientação de um tetraedro pode ser feita pelo processo descrito acima. No entanto, a subdivisão de um tetraedro com certa orientação gera dois filhos com orientações bem determinadas, que só dependem da orientação do pai. Lembrando que triangulações retangulares são obtidas por sucessivas subdivisões de tetraedros, é possível traçar regras simples para a determinação da orientação dos filhos de um tetraedro a partir da orientação de seu pai, evitando, assim, os cálculos acima indicados.

Em vista das observações iniciais sobre orientações de tetraedros, pode-se levantar o efeito da subdivisão na orientação estudando apenas um tetraedro padrão de cada tipo. Por exemplo, o tetraedro de vértices

$$v_0 = (0,0,0) \quad v_1 = (1,0,0) \quad v_2 = (1,1,0) \quad v_3 = (1,1,1)$$

é do tipo A e possui $(1,2,3)$ como seu vetor de orientação. Seus dois filhos:

filho 1	$v_0 = (1,0,0)$	$v_1 = (1,1,0)$
	$v_2 = (1/2, 1/2, 1/2)$	$v_3 = (1,1,1)$
filho 2	$v_0 = (1,1,0)$	$v_1 = (1,0,0)$
	$v_2 = (1/2, 1/2, 1/2)$	$v_3 = (0,0,0)$

possuem $(1,3,5)$ e $(-5,-3,-1)$ como seus respectivos vetores de orientação. Ora, então, um tetraedro do tipo A com orientação $(2,-3,1)$ terá seus dois filhos com orientações iguais respectivamente a $(3,-5,1)$ e $(-1,5,-3)$.

Analisando os demais tipos de tetraedros com orientações padrão chega-se à tabela 3.7 indicativa das orientações dos seus filhos:

Tipo	Vetor de orientação		Vetor de orientação dos filhos	
	do pai	Filho 1	Filho 2	
A	$(1,2,3)$	$(1,3,5)$	$(-5,-3,-1)$	
B	$(1,3,5)$	$(1,2,4)$	$(1,-4,-2)$	
C	$(1,2,4)$	$(1,2,3)$	$(1,2,-3)$	

Tabela 3.7

Examinando essa tabela verificamos que o filho 1 de cada tetraedro possui uma orientação cujo índices P de permutação e S de sinal são iguais ao do pai. O filho 2, por seu turno, possui um vetor de orientação cujo índices P e S dependem não só da orientação do pai mas de seu tipo. As tabelas 3.8 e 3.9 fornecem os índices P e S dos filhos a partir dos índices P e S do pai. Por intermédio dela os índices de orientação dos filhos podem ser facilmente calculados.

Permutação do pai	Índices de permutação do filho 2		
	Pai A	Pai B	Pai C
0	3	1	0
1	2	0	1
2	1	5	2
3	0	4	3
4	5	3	4
5	4	2	5

Tabela 3.8

Sinal do pai	Índices de sinal do filho 2		
	Pai A	Pai B	Pai C
0	7	3	4
1	6	2	5
2	5	1	6
3	4	0	7
4	7	7	0
5	2	6	1
6	1	5	2
7	0	4	3

Tabela 3.9

Para a determinação de P e S dos filhos podem-se adotar as tabelas acima ou alternativamente as expressões dadas pela tabela 3.10 que as resumem:

Pai (P,S)	Filho 1 (P1,S1)	Filho 2 (P2,S2)
A	$P1 = P$ $S1 = S$	$P2 = (9-P) \text{ mod } 6$ $S2 = 7-S$
B	$P1 = P$ $S1 = S$	$P2 = (7-P) \text{ mod } 6$ $S2 = (11-S) \text{ mod } 8$
C	$P1 = P$ $S1 = S$	$P2 = P$ $S2 = (12+S) \text{ mod } 8$

Tabela 3.10

O Catálogo de Orientações deve incluir, para cada uma de suas entradas, ou seja, para cada tetraedro padrão, a indicação das orientações de seus dois filhos, evitando as freqüentes avaliações desses índices, que seriam necessárias em determinados processos, como o método de projeção, por exemplo.

3.7.3 Orientações Indexadas Equivalentes

Nesta seção os termos *orientação* e *orientação indexada* são considerados diferentes, conforme mencionado anteriormente. Cada orientação de tetraedro corresponde a duas orientações indexadas distintas. A representação da orientação de tetraedros por meio do vetor de orientação distingue estas duas indexações. Assim, dois tetraedros com mesma orientação mas com indexações diferentes terão vetores de orientação distintos. Há situações em que essa distinção é conveniente. Já em outras circunstâncias pode ser necessário identificar os pares de vetores de orientação que representam uma mesma orientação (ver, por exemplo, no capítulo 6, a seção sobre a construção do catálogo de projeções).

Pela forma como foi definido o vetor de orientação de um tetraedro (V_{ori}), ele pode ser calculado como uma escala positiva do vetor diferença entre o vértice de índice 3 e o baricentro da face oposta a ele. Sendo v_0, v_1, v_2 e v_3 , os vértices do tetraedro, seu baricentro é dado por

$$b = (v_0 + v_1 + v_2) / 3.$$

Usando o valor 3 como fator de escala, resulta a expressão

$$V_{ori} = 3v_3 - v_0 - v_1 - v_2.$$

A outra indexação possível para esse mesmo tetraedro troca v_0 por v_3 e v_1 por v_2 . Assim, essa segunda indexação terá seu vetor de orientação dado pela expressão:

$$V_{ori}' = 3v_0 - v_1 - v_2 - v_3 = V_{ori} + 4(v_0 - v_3)$$

A tabela 3.11 fornece o vetor de orientação da indexação alternativa de cada um dos tetraedros padrão

Vetor de orientação	v_0	v_1	v_2	v_3	Vetor de orientação alternativo
(1,2,3)	(0,0,0)	(2,0,0)	(2,2,0)	(2,2,2)	(-3,-2,-1)
(1,3,5)	(2,0,0)	(2,2,0)	(1,1,1)	(2,2,2)	(1,-5,-3)
(1,2,4)	(2,2,0)	(1,1,1)	(2,1,1)	(2,2,2)	(1,2,-4)

Tabela 3.11 – Tetraedros padrão (v_0, v_1, v_2, v_3) e seus vetores de orientação.

Como o efeito das transformações de reflexão sobre as orientações se dão da mesma forma, tanto no vetor de orientação de um tetraedro quanto no vetor de orientação de sua segunda indexação, pode-se caracterizar, pela tabela acima, a segunda indexação de qualquer outra orientação. Como exemplo, pode-se deduzir que, para um tetraedro com vetor de orientação $(-3, 5, -1)$, o vetor de orientação da indexação alternativa será $(5, -3, -1)$, ou seja, as mesmas transformações que levam o vetor $(1, 3, 5)$ no vetor $(-3, 5, -1)$, também levam $(1, -5, -3)$ em $(5, -3, -1)$.

Uma maneira conveniente de aproveitar esta propriedade, é prover os registros do catálogo de orientações com um campo de apontadores de equivalência, preenchido com base nas relações representadas pela tabela 3.11.

3.8 Estrutura de Dados para Triangulações RA

A concepção de uma estrutura de dados deve levar em conta diversos aspectos inerentes à aplicação, ou às aplicações, a que ela dará suporte. O projetista sempre estará diante da necessidade de conciliar limitações de tempo e espaço para alcançar seus objetivos. Em geral, a inclusão de informações redundantes em uma estrutura de dados, é um caminho para diminuir o tempo de processamento necessário para a tarefa. No entanto, mesmo diante da disponibilidade de memória de baixo custo, esse recurso sempre tende a ser solicitado em seus limites. Seja porque o volume de dados cresce

proporcionalmente à disponibilidade de espaço, na busca incessante de melhorar a qualidade dos modelos, seja porque pretende-se que os sistemas de menor porte ofereçam acesso a uma variedade cada vez maior e mais sofisticada de serviços, o projeto de sistemas exige sempre considerar o consumo de memória, visando otimizá-lo, juntamente com o tempo de processamento.

O objetivo central deste trabalho é a geração de imagens a partir de dados volumétricos. Esta seção apresenta uma estrutura de dados para representar triangulações RA, que privilegia o atendimento aos requisitos necessários à tarefa de visualização. O projeto dessa estrutura partiu do pressuposto de que as operações relevantes nesse contexto são passíveis de implementação eficiente com base na organização hierárquica das triangulações RA. Essa consideração permite dispensar a representação explícita da maioria das informações topológicas e geométricas que normalmente são incluídas em estruturas irregulares, resultando em uma estrutura que tem concisão e flexibilidade como suas características marcantes. O acesso a um tetraedro (setor ou folha de uma árvore) através da hierarquia, permite deduzir quais são seus vértices a partir dos vértices de seu pai. Operações como busca e ordenação de células também são facilitadas.

Os capítulos 6 e 7, que mostram como implementar as técnicas de visualização direta sobre triangulações RA, dão destaque a algoritmos baseados na organização hierárquica dessa estrutura.

Os elementos constituintes da estrutura são: tetraedros (setores) e vértices. Os tetraedros da triangulação raiz são alocados em grupos de 6. Cada um desses grupos constitui um núcleo e é associado a um cubo da malha inicial. Esses cubos são organizados, a partir da origem da malha, segundo a ordem crescente de suas coordenadas nos eixos X, Y e Z. Isto permite que os tetraedros componentes do núcleo correspondente a um determinado cubo sejam identificados implicitamente, em função dos índices do cubo (ver seção 6.3.1).

Os vértices relativos à triangulação raiz também são organizados segundo sua ordem nos eixos coordenados e, portanto, suas relações com cubos e tetraedros da triangulação raiz também são obtidas implicitamente (ver seção 6.3.1).

Cada tetraedro folha, ao ser dividido, passa a ser considerado como setor e guarda o endereço do seu vértice divisor, bem como os endereços de seus filhos. Essas são as únicas relações topológicas representadas explicitamente. Observa-se que as informações associadas a tetraedros só existem para aqueles que são nós intermediários das árvores (setores), sendo, portanto, desnecessário representar as folhas na estrutura de dados. Vale lembrar que o número de nós intermediários de uma árvore binária é aproximadamente igual ao número de folhas.

As informações geométricas a serem representadas são as coordenadas dos vértices divisores de núcleos.

O algoritmo 3.3 é um exemplo de algoritmo de percurso de uma árvore da triangulação. Pode-se observar que os vértices dos filhos são deduzidos a partir de informações do pai, com base na sistemática estabelecida para rotulação dos vértices (seção 3.2).

```
PercorreÁrvore(T, V0, V1, V2, V3)
```

```
  tetraedro T
```

```
  vértice  V0, V1, V2, V3
```

```
  se ehFolha(T) então
```

```
    ProcessaFolha
```

```
  senão
```

```
    PercorreÁrvore(T.filho1, V1, V2, T.Vdivisor, V3)
```

```
    PercorreÁrvore(T.filho2, V2, V1, T.Vdivisor, V0)
```

```
    ProcessaSetor
```

Algoritmo 3.3 – Percorso de uma árvore de uma triangulação RA.
Os vértices dos filhos são informados pelo pai.

3.8.1 Recuperação das Relações de Adjacência

Embora as principais técnicas de visualização possam ser implementadas de maneira eficiente com essa organização dos dados, deve-se ressaltar uma consulta que é especialmente penalizada com essa estrutura: a identificação de um vizinho de um tetraedro. Essa é uma informação fundamental em diversos algoritmos e, por isso mesmo, normalmente é incluída nas estruturas de dados usadas para representação de triangulações. No presente contexto, sua falta é sentida já no processo de construção de uma triangulação RA, onde é largamente requisitada. O argumento para desconsiderar essa necessidade é calcado no fato de que o tempo consumido na construção da estrutura é bastante pequeno, quando comparado com as demais atividades envolvidas no trabalho de exploração dos dados. Acrescenta-se, ainda, a possibilidade de que, uma vez construída, a estrutura possa ser armazenada em memória não volátil, diminuindo ainda mais o custo relativo daquela tarefa.

Uma vez feita a opção por não prover a informação, no entanto, é importante buscar uma boa solução para recuperá-la quando necessário. Seja o problema de identificar o tetraedro *tviz*, vizinho de um dado tetraedro *t*, pela face *i*. Uma solução para esta questão deve levar em conta o fato de que o acesso a *t* se deu por um percurso recursivo da árvore a que ele pertence. Em linhas gerais, o algoritmo poderia ser: retornar na seqüência de ascendentes de *t* até encontrar, caso exista, um ancestral cujo outro filho também contenha a face em questão; procurar nesse outro ramo de descendência, o tetraedro que possui aquela face; caso não exista tal ascendente, identificar a outra árvore que contenha a face e procurar nela o tetraedro desejado. Este algoritmo tem complexidade $O(n)$ no pior caso e $O(\log n)$ no caso mais favorável (árvores balanceadas correspondentes a uma triangulação RA regular).

Outra solução que vem sendo experimentada com bons resultados, embora ainda não publicados, utiliza um algoritmo convencional de busca (“hashing”), para encontrar uma face a partir de seus vértices. Essa proposta exige acrescentar à estrutura os endereços dos vértices de cada tetraedro. Isto poderia ser feito apenas para a fase de construção da triangulação, descartando essas informações redundantes ao final.

Finalmente, estendendo essa última estratégia, pode-se utilizar, na montagem da estrutura, uma representação que inclua também os endereços das células vizinhas de cada tetraedro, dispensando, assim, qualquer algoritmo para esse fim. Da mesma forma, terminada essa fase, as informações redundantes seriam descartadas.

3.8.2 Requisitos de Espaço

A esta altura da discussão, já se podem estabelecer parâmetros quantitativos de comparação das triangulações RA com outras estruturas adaptativas, no que se refere ao consumo de memória. Uma das grandes vantagens das triangulações RA para visualização é que, implementando as técnicas com base na organização hierárquica dessas estruturas, obtém-se uma simplificação significativa das relações topológicas a serem representadas. Este é um fator que merece ser analisado, uma vez que as informações sobre a topologia são responsáveis pela maior parte do espaço ocupado pelas estruturas irregulares.

Buscando quantificar esses ganhos, as triangulações genéricas serão consideradas como referencial comparativo, conforme estabelecido no capítulo 2, por ser o tipo de estrutura adaptativa mais usado. A estrutura de dados mais freqüentemente adotada para representar triangulações genéricas inclui, como informações relativas à topologia, os endereços dos vértices de cada tetraedro, bem como de suas células vizinhas. Isso perfaz um consumo total de $8n$ palavras de memória, onde n é o número de tetraedros da estrutura.

Conforme exposto anteriormente, as triangulações RA armazenam, para cada setor, o endereço de seu vértice divisor e os de seus dois filhos. Apenas os setores são representados e eles existem em número aproximadamente igual ao de tetraedros folhas, que são as células da triangulação propriamente ditas. Portanto, o espaço ocupado por esses dados é de $3n$ palavras de memória.

Para avaliar o espaço ocupado pelas informações geométricas, que são as coordenadas dos vértices, é preciso considerar a relação entre o número de vértices e o número de tetraedros de uma triangulação tridimensional. Essa relação situa-se em torno de $1/6$ (um vértice para cada seis tetraedros). Esse parâmetro provém de uma avaliação empírica baseada em malhas com distribuição homogênea de seus vértices, portanto, malhas que não apresentam tendência especial de concentração de vértices nas bordas. Essa tendência, assim com outras distribuições particulares, resultariam em uma elevação daquela relação, que precisaria ser reavaliada.

Assumindo como parâmetro esse valor de $1/6$, e considerando que os dados geométricos de cada vértice são constituídos de suas três coordenadas, conclui-se que o requisito total de memória para triangulações genéricas situa-se em torno de $8,5n$ palavras de memória, contra $3,5n$ das triangulações RA. Esses resultados supõem que endereços de memória e números reais sejam representados em uma palavra de memória. Em termos percentuais, as triangulações RA ocupam pouco mais de 41% do espaço requerido por triangulações genéricas.

Não estão incluídas, nesta análise, outros dados envolvidos nos processos de visualização, que dependem das técnicas escolhidas para compor um particular sistema,

assim como de opções e variantes de implementação dessas técnicas. Por exemplo, um sistema voltado para visualização de dados escalares incluiria, para cada vértice, pelo menos o valor escalar em estudo. Dependendo da forma adotada para realizar o mapeamento de propriedades óticas ao campo escalar, pode ser conveniente incluir também valores de cor e opacidade no registro de cada vértice, para evitar a repetição de uma operação custosa de mapeamento para esses pontos. Pode ser útil, ainda, armazenar o vetor gradiente da função em cada vértice. A eventual necessidade dessas informações independe do tipo de estrutura, sendo, portanto, um acréscimo a ser considerado nas duas classes de triangulações em foco. Se todas essas informações forem incorporadas aos dados do programa, podem representar um acréscimo de até 8 palavras de memória para cada vértice, dependendo do grau de precisão com que esses dados sejam tratados. Considerando a relação entre número de vértices e número de células das estruturas, o acréscimo seria de $8/6n$ palavras, elevando o requisito total de memória das triangulações genéricas para aproximadamente $9,8n$, contra $4,8n$ palavras das triangulações RA.

Conclui-se que a utilização de triangulações RA como base para o processo de visualização proporciona uma economia de espaço em torno de 50%. O próximo capítulo mostrará que, mesmo com essa economia de memória, a estrutura oferece flexibilidade suficiente para implementar esquemas eficientes de multiresolução, à custa de um pequeno acréscimo no espaço requerido.

Capítulo 4

Multiresolução em Triangulações

Retangulares Adaptativas

O caráter hierárquico inerente às triangulações RA constitui uma base conveniente para a definição de modelos de multiresolução. Pode-se considerar o modelo do campo associado a cada nível de refinamento da triangulação como um *modelo simplificado*, uma vez que a estrutura resultante, ao desprezar os níveis inferiores, também é uma triangulação RA. No entanto, os desvios desse modelo simplificado em relação ao modelo completo podem ser altamente discrepantes de região para região, especialmente quando se leva em conta a adaptatividade da estrutura completa. Pode-se, então, incluir no modelo simplificado outros níveis mais baixos da triangulação, apenas para os ramos relativos a setores que estejam apresentando erro elevado, até que o erro global se situe em um patamar aceitável.

Embora essa forma de descrição sugira um percurso da estrutura em largura, é mais eficiente realizar a busca dos tetraedros que constituirão o modelo simplificado, através de um percurso em profundidade, aplicado a cada árvore da triangulação. Cada célula da triangulação raiz é a raiz de uma árvore BSP. Para percorrer uma dessas árvores, estabelece-se um critério de prioridade para visitação dos filhos. Esse percurso pode progredir até alcançar as folhas da árvore ou pode ser interrompido em algum nó intermediário (setor), de acordo com algum critério que se coadune com o objetivo visado. A inserção desse procedimento em um esquema de multiresolução é natural, bastando acrescentar ao critério de avanço o controle do erro admitido. Se um nó intermediário de uma árvore A tiver o erro associado (seção 4.1) abaixo do limite estabelecido, o percurso de sua subárvore será abortado, passando aquele setor a ser considerado como folha de A .

O resultado da aplicação deste processo a todas as árvores da estrutura original, é um conjunto de novas árvores, cujas folhas podem não formar uma triangulação consistente, comprometendo a continuidade da função representada pelo modelo simplificado assim obtido. Para garantir essa consistência, a decisão de interromper o percurso em um nó intermediário precisa levar em conta um critério adicional: sendo S o setor relativo a esse nó, além do erro associado a S , estar abaixo do limite admitido, os demais setores integrantes do núcleo a que ele pertence também deverão respeitar esse limite.¹ A verificação dos erros dos setores de um núcleo exige que a estrutura ofereça

¹ O núcleo do tetraedro t pode não existir, o que fica caracterizado pela presença de algum tetraedro de tipo diferente (e conseqüentemente de tamanho diferente) do tipo de t , entre os que compartilham sua maior aresta. Isto significa, no entanto, que o tetraedro t não era

facilidade para recuperar a constituição de cada núcleo. Uma possível solução para isso, proposta em [Zhou97], é criar uma estrutura auxiliar que armazene, para cada vértice, os endereços dos setores constituintes do núcleo que ele divide.

No entanto, a condição de que o maior erro associado aos setores de um núcleo, ou seja, o erro do núcleo, esteja abaixo do limite tolerado, ainda não é suficiente para evitar todos os casos de inconsistência da triangulação. A seção 4.1.1 mostra que é preciso levar em conta a interdependência de vértices de níveis consecutivos.

O material apresentado neste capítulo tem, no citado artigo de Y. Zhou e al., a referência mais próxima, embora aquele trabalho se situe no contexto de multiresolução para conjuntos de dados regulares. Os autores descrevem ali uma infra-estrutura, baseada em subdivisão simplicial, muito semelhante à triangulação RA. A presente abordagem, tira proveito daquelas idéias, ampliando sua aplicabilidade para dados com distribuição irregular. Será mostrado um esquema eficiente de multiresolução, implementado sobre a estrutura de dados apresentada no capítulo 3. São alcançados ganhos significativos do ponto de vista do espaço ocupado pela estrutura, vantagem que se mantém mesmo para o caso particular do tratamento de dados regulares.

4.1 Modelos Simplificados

O objetivo primordial da abordagem multiresolução é facilitar o acesso a versões simplificadas do conjunto de dados, que representem aproximações do modelo completo, dentro de determinados limites. É usual definir esses limites por parâmetros de erro que estabeleçam os desvios admissíveis nos modelos intermediários. Podem ser adotadas formas variadas de definição de erro. Neste trabalho serão consideradas duas metodologias, provenientes de uma mesma concepção básica, mas diferenciadas no grau de complexidade e precisão (seção 4.3). Outras alternativas são possíveis, mesmo que envolvam custos computacionais mais elevados, uma vez que, como será mostrado adiante, o cálculo de erros não se repete freqüentemente, constituindo uma componente de pré-processamento.

O critério de precisão adotado pressupõe que o processo de refinamento da estrutura retangular adaptativa já esteja concluído. A partir daí, toma-se como modelo do campo, uma função definida sobre a triangulação RA como um todo. Será considerada uma função linear por partes, interpoladora dos valores do campo escalar estimados nos vértices da malha. Esse modelo será a referência para avaliação do erro.

A proposta que se apresenta consiste em simular sucessivas podas da estrutura hierárquica para avaliar o erro que essa operação introduziria. Este processo equivale à remoção de vértices da estrutura original (triangulação RA completa), começando pelas folhas das árvores BSP e progredindo recursivamente até alcançar suas raízes. A cada vértice removido (supostamente) avalia-se o erro a ser associado a ele, como sendo o desvio máximo entre as funções lineares inerentes aos setores antes subdivididos pelo

subdividido na triangulação original, pois apenas núcleos podem ser subdivididos. Conclui-se que t já era folha da árvore e, portanto, a interrupção do percurso no nó em questão é obrigatória.

vértice, que passam a ser folhas da árvore podada, e o modelo do campo (modelo original que considera a triangulação completa).

Esta abordagem pode servir de base a diversos esquemas de obtenção de modelos simplificados, alguns dos quais serão descritos adiante. Ela é consistente com o requisito introduzido anteriormente de que cada nó da árvore tivesse um valor de erro associado. Nesta proposta, esse erro é obtido consultando o registro do vértice que subdivide esse setor para gerar seus filhos, e é assumido como sendo o desvio introduzido no modelo pelo truncamento da árvore a partir dos filhos daquele nó. Além de sua simplicidade, esta estratégia, embute uma economia de espaço ao evitar o armazenamento de erros associados diretamente aos setores, uma vez que o número de vértices é, normalmente, muito menor do que o número de setores nessas estruturas. A estrutura completa, incluindo os erros associados aos vértices, será referida como *triangulação RA multiresolução*. No entanto, a atribuição de erros a vértices descrita acima deve ainda ser revista, para chegar à configuração definitiva dessa estrutura.

4.1.1 Propagação de Erro

A aproximação para um modelo simplificado é especificada por um limite de precisão, parâmetro esse que é comparado com o erro associado a cada vértice candidato à remoção. Cabe, neste ponto, estender um pouco mais a observação das consequências de uma poda indiscriminada de uma árvore da estrutura. Se a remoção de um vértice for decidida apenas em função do erro associado a ele, pode ocorrer a geração de configurações de estrutura que apresentem inconsistência, ou seja, o resultado do processo pode não ser uma triangulação². Conforme observação anterior, a continuidade da função definida pelo modelo poderia ficar comprometida. A figura 4.1 ilustra, em duas dimensões, uma situação em que a remoção do ponto b não pode ser feita sem que o ponto a também seja removido, para que a estrutura obtida seja uma triangulação. Supondo que a avaliação de erro indicasse o erro em b menor do que em a , ainda assim o vértice b não poderia ser removido isoladamente, ou seja, se o vértice a tiver que estar presente em um modelo simplificado, o vértice b também deverá estar, assim como os vértices c , d e outros não marcados, de cuja existência a existência de a depende.

A condição descrita acima pode ocorrer com frequência em malhas adaptativas, onde aquela vinculação entre vértices se dá na transição entre regiões de alta e baixa complexidade, que podem ser caracterizadas por simplexos relativamente pequenos e grandes, respectivamente. Uma situação em que isto acontece é quando essa complexidade da estrutura já está associada à grandeza escalar em estudo. Seja, por exemplo, um caso em que, ainda na fase de construção da estrutura, o próprio critério determinante do maior aprofundamento de seu refinamento em uma determinada região, já tenha sido baseado no desvio dos modelos mais grosseiros em relação aos dados originais. Em outras palavras, em cada estágio do refinamento, a célula escolhida para ser subdividida tenha sido aquela em que o modelo atual tenha apresentado maior erro em relação àqueles dados. Isto aumenta a probabilidade de que, em um posterior

² Ver definição de triangulação no capítulo 3.

processo de simplificação, a remoção de um vértice associado a simplexes pequenos introduza um erro relativamente alto no modelo resultante.

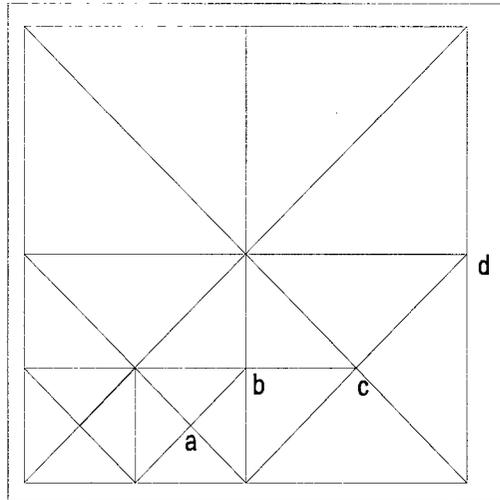


Figura 4.1 - Remoções induzidas: a remoção de d exige a remoção de c , que exige a de b , que por sua vez exige a de a .

Uma solução para organizar os dados de forma a facilitar o tratamento dessas situações, pode ser associar a cada vértice o erro máximo encontrado na seqüência de vértices de cujas remoções sua remoção dependeria. Na figura 4.1, se o erro em a é maior do que os erros em b , c e d , a remoção de qualquer um desses últimos está vinculada à de a . Isto indica, na estratégia proposta, que para todos esses vértices deverá ser atribuído o erro atribuído a a . Isto equivale, ainda, a dizer que os valores de erro associados aos vértices dessa seqüência (considerada na ordem crescente de tamanhos dos respectivos núcleos) devem ser não decrescentes. Esse novo valor do erro, atribuído ao vértice por indução, será chamado de *erro propagado*.

A implementação dessa proposta será discutida mais adiante, na seção 4.3. Porém, será considerado, desde já, que essa propagação de erro já tenha sido efetuada. Os esquemas de obtenção de modelos simplificados, que serão apresentados na próxima seção, valem-se dessa característica das triangulações RA multiresolução.

4.2 Extração de Modelos Simplificados

As várias circunstâncias em que os modelos simplificados são úteis apresentam condicionamentos específicos, que por sua vez apontam para diferentes esquemas de extração dos modelos, de molde a atender àqueles requisitos com eficiência. Entre esses aspectos condicionantes, devem-se ressaltar as características da função de distribuição de tolerância (para ser mais conciso, função de tolerância) e o uso local/remoto dos modelos obtidos.

A tolerância é normalmente representada por funções simples, de forma a que sua avaliação não seja muito onerosa para o processo como um todo. A própria função constante é a mais usada, gerando o que será chamado de *modelo homogêneo*. Outras funções são recomendadas, entre as quais ressaltam-se duas, especialmente convenientes para a composição de interfaces para manipulação interativa de um sistema de visualização.

A primeira dessas funções associa a cada ponto uma tolerância diretamente proporcional a sua distância em relação ao observador. A consideração implícita nesta opção é que o interesse do observador está concentrado na porção do volume situada mais próxima dele. Assim sendo, essa região deve ser exibida com detalhes, enquanto que as porções mais distantes são consideradas menos importantes, podendo ser exibidas com menor precisão.

A segunda função aqui considerada associa um valor de erro admissível baixo (possivelmente nulo) e constante a uma região indicada pelo usuário, ou eventualmente selecionada pelo próprio sistema. Fora dessa área em foco a tolerância é ampliada, associando-se um valor de erro admissível mais elevado e também constante.

Quanto ao uso local ou remoto dos modelos obtidos, deve-se observar que a transmissão de um modelo via rede é uma operação que tem requisitos próprios: o volume de dados a ser transmitido deve ser o menor possível e deve ser fácil reconstituir a estrutura no computador cliente.

Serão apresentados a seguir alguns esquemas, construídos sobre as triangulações RA multiresolução para atender a essas necessidades. São caracterizadas duas classes de modelos ou, mais propriamente, de métodos para extração de modelos: *dinâmicos* e *pré-computados*. Os modelos dinâmicos são assim denominados porque são extraídos durante o próprio percurso da estrutura para realização da operação objetivo (geração de uma imagem ou transmissão de modelos para outro computador). Os métodos dinâmicos descritos a seguir são aplicáveis à extração de modelos homogêneos. Sua extensão para outras funções de tolerância ainda está em estudo. Os modelos pré-computados são obtidos em um percurso da estrutura realizado especificamente para efetuar a poda das árvores. A ordem desse percurso é imprópria para as operações usuais, que são realizadas posteriormente.

4.2.1 Modelos Dinâmicos

Um modelo simplificado é definido pelo grau de precisão ϵ desejado. A princípio, o usuário pode escolher essa tolerância em uma escala contínua de valores. Uma vez que esteja disponível a triangulação RA multiresolução, proveniente do modelo completo, a identificação da porção de cada árvore que faz parte do modelo simplificado se dá por um percurso dessa árvore em profundidade. Ao visitar cada setor, o erro associado ao vértice que o divide é comparado com a precisão estabelecida. Se esse erro for maior do que ϵ , o percurso prossegue (o vértice pertence ao modelo simplificado); se for menor, o setor corrente passa a ser interpretado como folha da árvore nesse modelo (se ainda não se trata de uma folha da árvore completa), e o

percurso de sua descendência é abortado. As árvores podem ser visitadas em qualquer ordem; também é indiferente a ordem de visita aos filhos dos setores de cada árvore.

A consistência da triangulação assim constituída estará assegurada pelo critério estabelecido para associação de erros aos vértices da estrutura, o que também garante, por conseqüência, a continuidade da função escalar do campo.

A simplicidade desta proposta permite antever uma alta eficiência do processo e uma boa flexibilidade. Pode servir de base, por exemplo, para um esquema interativo de geração de imagens, onde a precisão possa ser estabelecida dinamicamente, ou seja, a qualquer momento o usuário pode escolher o valor de ϵ a ser usado a partir de então para gerar as próximas imagens. Isto equívale, ainda, a dizer que o modelo simplificado estará sendo extraído concomitantemente com a geração da imagem. Este esquema é especialmente conveniente para implementação com o método de geração de imagens por projeção. No capítulo 6, que trata especificamente daquele método, serão discutidas questões relativas à implementação desta proposta.

4.2.2 Modelos Dinâmicos em Série

Considere a situação em que um servidor deva disponibilizar modelos de resolução progressivamente mais fina, em uma rede de baixa velocidade, para visualização nos clientes³. Uma das vantagens desse esquema é oferecer rapidamente ao usuário imagens que, mesmo sendo grosseiras, permitam uma primeira observação dos dados, subsidiando uma eventual decisão de abortar a transmissão de maiores detalhes de um modelo, que já tenha sido constatado ser indesejável ou desnecessário. As sucessivas imagens com detalhamento progressivo devem ser apresentadas tão logo estejam disponíveis no cliente, sem que o usuário interfira para estabelecer parâmetros de precisão.

Uma forma trivial para atender a este tipo de demanda é considerar que cada nível hierárquico da estrutura constitui um modelo. O erro característico de um tal modelo é o maior erro associado aos vértices divisores dos setores daquele nível. O servidor inicia a transmissão pelo modelo mais simples (o de menor precisão) que será então tratado pelo cliente para geração da imagem correspondente. Assim que possível, será iniciada a transferência da porção complementar da estrutura referente ao modelo subsequente. Apenas os tetraedros que não estão presentes na primeira versão precisam ser transmitidos. Com este procedimento, as árvores do modelo parcial vão sendo expandidas progressivamente nos clientes até alcançarem a configuração completa do modelo global, permitindo a geração de imagens de qualidade cada vez melhor.

Os inconvenientes deste esquema são: a impossibilidade de se prever o erro do próximo modelo e a inexistência de caráter adaptativo nos primeiros modelos. A tendência é que esses primeiros modelos sejam muito pouco significativos.

³ Para que a estrutura de dados sobre a qual se implementam esses modelos, sejam eles de multiresolução ou não, possa ser transmitida entre computadores, é necessário que ela seja alocada em arranjos, evitando o uso de ponteiros fornecidos pelo sistema operacional, que serão diferentes nas diferentes máquinas. Dessa forma, a identificação dos elementos da estrutura se faz por seus índices nesses arranjos, permanecendo inalterados ao transitarem de uma máquina para outra.

Uma solução melhor, que corrige essas deficiências, é estabelecer previamente uma série de níveis de precisão. Esses níveis, que caracterizarão os modelos simplificados, podem ser definidos automaticamente, dividindo um dado valor de erro máximo, admissível para a primeira aproximação, pelo número de modelos parciais determinado para a série. Outra opção seria um especialista fazer uma exploração prévia dos dados, estabelecendo os níveis da série conforme sua avaliação do que seria relevante acrescentar em novos detalhes na transição entre dois níveis subseqüentes. Neste último caso, os níveis estabelecidos (valores de erro) seriam armazenados em um arranjo unidimensional. De qualquer forma, os sucessivos modelos seriam obtidos percorrendo as árvores até alcançar os limites de definição do próximo nível da lista. A verificação desses limites é feita pela comparação do erro admissível, característico daquele nível, com o erro associado aos vértices encontrados ao longo do percurso.

4.2.3 Modelos Pré-Computados

Este é um método geral para extração de modelos simplificados que pode ser aplicado a uma triangulação RA qualquer, mesmo que não tenha sido feita uma avaliação prévia de erros. O método também é flexível em relação ao tipo de função de tolerância escolhida. Tem como base a estratégia utilizada em [Zhou97] para visitar os vértices e os núcleos divididos por eles, a partir dos níveis inferiores da estrutura, onde estão as células mais refinadas. Esta proposta não foi aplicada naquele trabalho para esta finalidade, mas para avaliação de erros nos vértices, como se verá na seção 4.3.

A inclusão deste método deve-se à impossibilidade, até o momento, de aplicar eficientemente os métodos dinâmicos para obtenção de *modelos heterogêneos*, ou seja, modelos simplificados induzidos por uma função de tolerância não constante. Essa tarefa apresenta complicações para a manutenção da consistência da triangulação, as quais não têm solução assegurada pela simples utilização da triangulação RA multiresolução, como acontece nos modelos homogêneos.

Embora o trabalho de Zhou refira-se à extração de modelos simplificados com resolução variável, o método descrito por ele é semelhante ao método dinâmico da seção 4.2.1. Tal mecanismo, entretanto, não garante a consistência do modelo quando a função de tolerância não é constante.

Para ilustrar o problema, considere, por exemplo, a figura 4.2, que repete a configuração da figura 4.1 e marca uma região de foco (retângulo tracejado). Supondo que o erro avaliado em a (E_a) seja maior do que os de b , c e d , pela metodologia adotada para associar erros a vértices, o valor E_a será replicado nos outros três. Considere-se ainda que o erro admitido na região em foco seja menor do que E_a , enquanto que fora do retângulo o erro admissível seja maior do que E_a . O simples percurso das árvores em profundidade, com que se implementa o método dinâmico, poderia alcançar primeiramente a célula dividida pelo vértice d , o qual seria marcado como componente do modelo simplificado, uma vez que o erro associado a ele é relativamente baixo para a região onde está situado. Dessa forma, os níveis mais baixos da árvore seriam desconsiderados, não chegando a ser verificado o vértice c . No entanto, esse mesmo vértice poderia ser alcançado e incorporado ao modelo simplificado com os outros

tetraedros que ele compõe, pertencentes a outro ramo da árvore, o que caracterizaria uma descontinuidade da função representada por esse modelo.

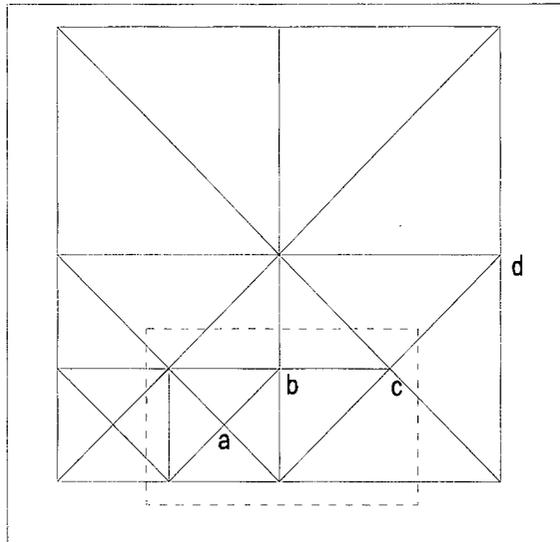


Figura 4.2 – Resolução heterogênea.
Região em foco com tolerância menor.

O exemplo acima mostra uma situação que pode ocorrer, seja qual for a função de tolerância considerada, desde que não seja uma função constante em todo o domínio, característica dos modelos com resolução homogênea. A causa do problema é o fato de que o percurso de um dos ramos da árvore alcançou uma região com tolerância elevada, antes de outra com tolerância menor, mas que tem relação de dependência com algum vértice da primeira.

A estratégia de solução que se apresenta é visitar cada vértice, começando do nível mais baixo da estrutura, para avaliar a necessidade de sua presença no modelo simplificado. Cada vértice visitado tem seu erro associado comparado com o erro admissível em sua posição, estipulado pela função de tolerância. São verificados todos os vértices de cada nível, antes de passar ao nível superior. Ao tratar um vértice v do nível imediatamente superior pode-se confirmar se algum vértice do nível anterior, marcado como obrigatório no modelo simplificado, depende da presença de v na estrutura. Essa confirmação depende de se ter acesso aos filhos dos setores divididos por v . Na organização de dados do citado trabalho, isso exige que antes se conheçam os próprios setores, para através deles se chegar a seus filhos. Essa informação é suprida por uma estrutura auxiliar que armazena, para cada vértice, a lista dos setores que ele divide. Essa estrutura tem tamanho $O(n)$, sendo n o número de tetraedros, pois serão citados todos os setores existentes, em número aproximadamente igual ao número de tetraedros (folhas) da triangulação.

Para implementar essa estratégia, falta estabelecer uma maneira eficiente de identificar quais são os vértices pertencentes a um mesmo nível. Essa informação poderia ser obtida por um percurso completo das árvores até o nível desejado. Repetir esse percurso a cada novo nível, resultaria em um algoritmo de custo elevado. Isso pode ser evitado, como é feito naquele trabalho, criando uma tabela de pontos divisores.

Trata-se de uma lista com um elemento para cada nível das árvores, elemento esse que é um ponteiro para a lista de vértices do nível respectivo. Essa tabela é preenchida com um único percurso da triangulação. Esta nova estrutura auxiliar é bem menor do que a primeira pois inclui apenas uma entrada para cada vértice, além de uma para cada nível. Devido ao ganho de eficiência que ela proporciona, justifica-se a sua manutenção.

4.3 Metodologias para Cálculo de Erro

Considera-se, aqui, que uma metodologia para cálculo de erro inclui a definição de um parâmetro para quantificar esse erro e uma estratégia de avaliação desse parâmetro.

Na proposta apresentada em [Zhou97], o parâmetro quantificador é estabelecido em função da técnica de visualização a ser utilizada. Em particular, aquele trabalho considera a técnica de extração de isosuperfícies. A inclusão ou exclusão de um vértice divisor de um núcleo, em geral provoca uma mudança de posição de pontos de interseção da isosuperfície com arestas dos tetraedros relacionados com aquele vértice. O erro associado a esse vértice é dado pela maior dessas variações. O inconveniente dessa estratégia é que um sistema que pretenda disponibilizar diferentes técnicas de visualização deverá calcular e armazenar diferentes parâmetros de erro.

Neste trabalho, optou-se por uma definição de erro associada ao modelo e não à técnica de visualização. O erro associado a um vértice também é o máximo entre os erros relativos aos tetraedros (setores) que ele divide. A diferença está no cálculo do erro em um setor, que é o erro cometido ao desprezar sua descendência. Esse será o desvio máximo entre a função linear definida pelos vértices desse setor e o modelo de referência do campo, constituído pela estrutura completa, que inclui todos os dados originais disponíveis em pontos pertencentes ao volume do setor. Resta ainda definir como calcular esse desvio. Serão apresentadas duas alternativas, de modo a oferecer opções com diferentes graus de complexidade e precisão. A primeira, descrita nesta mesma seção, é mais rigorosa e de custo computacional mais elevado. A segunda, descrita na seção 4.3.1 é uma simplificação da primeira.

A estratégia para avaliação do erro baseia-se na observação de que, por se tratarem de funções lineares definidas a partir dos vértices de uma hierarquia de simplexos, os extremos locais dessas funções, bem como os desvios máximos, ocorrerão exatamente nesses vértices. Sendo assim, o desvio máximo relativo a um setor, que será denominado *erro do setor*, ocorrerá sobre um dos vértices geradores de sua descendência. Para calcular o erro de um setor S , a ser associado a seu vértice divisor v , percorre-se a subárvore de S para visitar seus vértices (divisores dos setores de sua descendência, incluindo v) e calcular, para cada um deles, a diferença entre o valor conhecido do campo naquele ponto, armazenado no registro do vértice, e o valor que a função linear definida pelos vértices de S (os quatro vértices do seu tetraedro) associa a esse ponto. A maior de todas essas diferenças é o erro do setor, que é o erro imposto pelo truncamento da árvore a partir dos filhos de S .

O algoritmo 4.1, "*ErroDoNúcleo*", implementa o cálculo desse erro para todos os setores do núcleo dividido por um dado vértice. O maior deles fica associado a esse vértice. O algoritmo complementar "*ErroDoDescendente*", determina, para um setor

dado por seus quatro vértices, o maior erro nos vértices situados no interior de um seu descendente.

```

ErroDoNúcleo (v, erro)
    vértice v0, v1, v2, v3, v
    setor S
    real erro, e

    // N é o núcleo dividido por v
    v0, v3 ← vértices do eixo de N
    erro = || v.valor - (v0.valor + v3.valor)/2 ||
    para cada setor S de N
        v1, v2 ← vértices 1 e 2 de S
        ErroDoDescendente (v0, v1, v2, v3, S.filho1, e)
        se e > erro então erro = e
        ErroDoDescendente (v0, v1, v2, v3, S.filho2, e)
        se e > erro então erro = e

ErroDoDescendente (v0, v1, v2, v3, S, erro)
    setor S
    vértice v0, v1, v2, v3 // vértices do setor Sasc, ascendente de S
    real erro, e
    vértice v

    se existe S.vert_div então
        // o setor S possui filhos
        v = S.vert_div // vértice divisor de S
        erro = || v.valor - valor interpolado de Sasc na posição de v ||
        ErroDoDescendente (v0, v1, v2, v3, S.filho1, e)
        se e > erro então erro = e
        ErroDoDescendente (v0, v1, v2, v3, S.filho2, e)
        se e > erro então erro = e
    senão erro = 0

```

Algoritmo 4.1 – Cálculo do erro (dos setores) de um núcleo

Foi mostrada, na seção 4.1.1, a necessidade de propagar um valor elevado de erro, obtido em um vértice de nível baixo, para os vértices de níveis superiores que lhe dão sustentação. O trabalho de Zhou apresenta uma boa solução para esse problema. Visitando primeiramente os vértices dos níveis mais baixos, ao tratar um vértice v , já são conhecidos os valores de erro associados aos vértices divisores dos filhos dos setores pertencentes ao núcleo (N) dividido por v . São justamente esses os vértices que podem ter seu valor de erro transmitido (propagado) para v . O valor de erro efetivamente associado a v deverá ser o maior dentre os seguintes:

- . o erro do núcleo N , dado pelo algoritmo acima;
- . o erro associado, pelo algoritmo acima, a cada um dos vértices que subdividem os filhos dos setores de N , se houver algum deles realmente subdividido.

O percurso dos vértices e dos setores por eles divididos, é implementado com base nas mesmas estruturas auxiliares descritas na seção 4.2.3, quais sejam, a lista dos setores divididos por cada vértice, e as listas de vértices divisores de núcleos em cada nível.

Uma análise acurada do algoritmo 4.1 mostra que muitos vértices são visitados repetidamente para avaliar o erro em relação a um mesmo setor. Algumas estratégias podem ser utilizadas para evitar essa redundância de cálculos. Embora essas alternativas sejam relativamente simples, sua avaliação foi deixada para o futuro, uma vez que, além desse cálculo de erros constituir apenas uma fase de pré-processamento, aquela melhoria não chegaria a afetar a complexidade assintótica dessa tarefa.

Analisando a complexidade do processo de avaliação de erro como um todo, conclui-se que seu comportamento é $O(n^2)$ em tempo, no pior caso, onde n representa o número de células da triangulação (folhas da árvore). Isto porque, para cada vértice é necessário visitar todos os descendentes dos setores que ele divide, com exceção das folhas. O pior caso ocorrerá quando a árvore tiver sido gerada por um processo adaptativo tendencioso, com a opção de refinamento recaindo sempre sobre o mesmo filho. A condição mais favorável ocorre para triangulações homogêneas, ou seja, aquelas em que todas as células estão no mesmo nível de refinamento. Nesses casos o comportamento do algoritmo é $O(n \log n)$.

4.3.1 Simplificação do Cálculo de Erro

Com a avaliação de erro realizada pela metodologia descrita acima, pode-se assegurar que o erro introduzido em um modelo simplificado, ao desconsiderar a subárvore relativa a um dado setor, não supera o erro associado ao vértice que subdivide esse setor. Entretanto, a complexidade quadrática no tempo daquele procedimento, justifica a busca de estratégias alternativas, ainda que sacrificando a precisão em certa medida. Em seguida, apresenta-se uma abordagem simplificadora que, admitindo um cálculo aproximado para o erro, proporciona um algoritmo de comportamento linear no pior caso.

Recorrendo novamente a um caso bidimensional, sugerido pela figura 4.3, suponha a remoção do vértice a . Com isso as arestas ba e ad deixam de existir e passa a existir a aresta ec , sobre a qual a função fica definida pelos valores atribuídos aos vértices e e c .

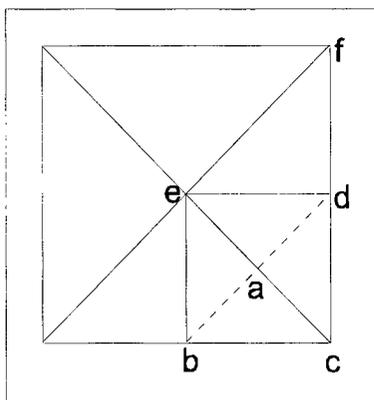


Figura 4.3 – Remoção do vértice a .

O erro associado ao vértice a será dado, neste caso, pela diferença entre o valor original da função nesse ponto, que permanece armazenado no registro do vértice, e o valor da função sobre ec calculado para o ponto a . Suponha agora a remoção do vértice d . O valor da função no ponto a não se altera, continuando definido pela aresta ec . Com a remoção do vértice e , quando então a aresta ec deixa de existir, é que o valor da função no ponto a se altera, impondo uma reavaliação do desvio nesse ponto. Como se observa, para o caso bidimensional em foco, o desvio no ponto a só precisa ser revisto no segundo nível de desrefinamento acima dele próprio. Em uma triangulação RA tridimensional essa condição se estende para o terceiro nível de desrefinamento, quando, como mostra a figura 4.4, deixa de existir a aresta que define o valor da função no ponto a . Neste caso, após a remoção de a , seriam removidos os vértices 2 e 3 sem alterar o valor da função em a , que seria mantido pela aresta 4-5. No terceiro nível de desrefinamento, removendo o vértice 4, é que haveria necessidade de reavaliar a função em a .

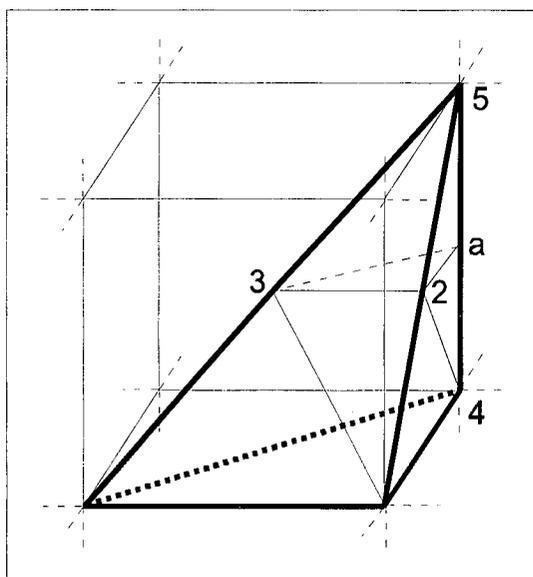


Figura 4.4 – Removendo o vértice a , a aresta 4-5 conserva o valor da função naquela posição, mesmo após a remoção dos vértices 2 e 3.

A hipótese simplificadora que se lança é que seja baixa a probabilidade de que o erro no ponto a possa prevalecer no terceiro nível de desrefinamento após a remoção desse vértice. Admitindo-se essa hipótese, pode-se negligenciar a reavaliação do erro nos níveis inferiores ao atual, transformando o cálculo do erro do núcleo, no simples cálculo do desvio na posição referente ao vértice que o divide. Esse desvio é a diferença entre o valor armazenado no vértice e o valor da função interpolado sobre o eixo do núcleo. O algoritmo resultante tem comportamento linear no tempo.

Outra consequência desta simplificação é que todas as considerações sobre propagação de erro feitas anteriormente se tornam desnecessárias. Com isso, a avaliação do erro nos vértices pode ser feita em qualquer ordem. Uma ordem definida por um percurso baseado na hierarquia da estrutura passa a ser uma solução mais conveniente, dispensando, assim, as estruturas auxiliares usadas para o percurso a partir das folhas. Essas estruturas continuam necessárias para a extração de modelos simplificados baseados em critério de tolerância variável. Em um sistema que utilize apenas o método dinâmico de extração desses modelos, definidos por função de tolerância constante, elas podem ser abolidas.

4.3.2 Avaliação Comparativa

A metodologia adotada por Zhou, em seu trabalho, para avaliação do erro relativo à poda da estrutura, mencionada no início da seção 4.3, foi estabelecida com base na variação de parâmetros morfológicos de isosuperfícies. O desvio relativo à remoção de um vértice considera apenas a forma da isosuperfície com e sem aquele vértice. Uma abordagem mais rigorosa deveria considerar o desvio do modelo sem o vértice, em relação ao modelo completo, que estaria, então, servindo como modelo de referência. Aquela estratégia, portanto, corresponde à alternativa simplificada da seção anterior que, por sinal, tem a mesma ordem de complexidade em tempo, com a vantagem de dispensar as estruturas auxiliares criadas por aqueles autores.

4.4 Requisitos de Espaço

Encerra-se esta abordagem com uma comparação entre a estrutura de dados descrita em [Zhou97] e a desenvolvida para este trabalho. Aquele estudo visa a uma aplicação muito particular, qual seja a representação de uma malha regular sob a forma de uma árvore binária de tetraedros. O resultado constitui uma árvore binária completa, gerada por um processo de subdivisão totalmente regular que produz uma estrutura hierárquica homogênea. Essa árvore pode, então, ser armazenada em um arranjo unidimensional, ficando dispensada de armazenar as informações relativas à estruturação da hierarquia, as quais ficam implicitamente definidas e são facilmente recuperadas. Cada registro desse arranjo representa um nó da árvore, que armazena exclusivamente endereços de vértices: os quatro que definem o tetraedro e seu vértice divisor. Os tetraedros folhas também são representados, apenas dispensando o campo relativo a vértice divisor.

Este trabalho, no entanto, trata de uma estrutura adaptativa, portanto irregular, do ponto de vista da densidade de informações nas diferentes regiões. Como consequência, a estrutura hierárquica resultante não é homogênea e não pode ser representada da daquela maneira. No entanto, sua regularidade construtiva permite um esquema de armazenamento ainda mais econômico do que aquele: cada registro relativo a um nó da árvore armazena apenas os endereços de seu vértice divisor e de seus dois filhos (capítulo 3). A economia está no menor número de ponteiros armazenados nos nós intermediários das árvores, uma vez que os endereços dos vértices do setor não são armazenados, e, principalmente, na total eliminação da representação de informações relativas às folhas dessas árvores. Obviamente, aquela aplicação específica também pode ser implementada sobre esta estrutura de dados.

Quanto às informações dos vértices, as duas estruturas são equivalentes. Ambas armazenam, para cada vértice, suas coordenadas e os endereços dos tetraedros que ele divide. Como já foi mencionado, essas listas de endereços, em conjunto, citam todos os setores da estrutura, totalizando, portanto, n elementos, onde n é o número de folhas das árvores. Deve-se acrescentar, ainda um ponteiro para inserir o vértice na lista de vértices do seu nível. A lista de ponteiros para essas listas de vértices (um para cada nível) será desprezada.

Para quantificar o consumo de memória, será considerada uma máquina que represente números reais e inteiros (incluindo endereçamento de memória) em uma palavra de memória. Também será assumida a relação de $1/6$ entre o número de vértices e o número de tetraedros dessas estruturas. Conclui-se que a estrutura usada por Zhou ocupa $10,7n$ palavras, ao passo que nossa estrutura ocupa $4,7n$ palavras de memória.

No capítulo 3 foram relacionadas diversas outras informações que podem vir a ser agregadas aos dados do problema. A essas, deve-se acrescentar, ainda, o erro em cada vértice, característico do contexto de multiresolução. Isso representa um acréscimo de $9/6n$ palavras, elevando aqueles totais para até $12,2n$ e $6,2n$.

Semelhante à conclusão daquele capítulo, também aqui a economia de espaço com o uso das triangulações RA, representadas com a estrutura deste trabalho, situa-se em torno de 50%.

Finalmente, deve-se considerar a pouca flexibilidade da estrutura de Zhou, do ponto de vista da extração de modelos simplificados. A representação de árvore binária em um arranjo linear só é eficiente, do ponto de vista de espaço, quando se trata de uma árvore completa. Um modelo simplificado caracteriza-se pelo desbalanceamento dessas árvores, o que leva a um desperdício de espaço naquela representação.

Capítulo 5

Visualização Volumétrica Direta

A visualização direta de dados volumétricos engloba uma variedade de técnicas para obtenção de imagens de um campo escalar tridimensional, sem extração ou construção de representações geométricas intermediárias. No estudo dos algoritmos envolvidos, destacam-se duas abordagens fundamentais, cuja diferença básica está na ordem de realização das operações envolvidas, que pode ser regida pela organização dos dados ou da imagem. Essas duas classes de algoritmos, que são designadas como técnicas de *projeção* e técnicas de *lançamento de raios*, receberão atenção particular em seções específicas deste capítulo, onde serão descritos algoritmos que as implementam sobre malhas irregulares. A ênfase em malhas irregulares, particularmente malhas de tetraedros (triangulações), se deve ao fato de que essas são as malhas adaptativas mais usadas, pela sua simplicidade e porque proporcionam os algoritmos mais eficientes, inclusive para implementação das técnicas de visualização. Isso faz dessas triangulações (genéricas) o melhor referencial, diante do objetivo de comparar o desempenho das técnicas citadas, quando aplicadas sobre triangulações retangulares adaptativas, com o seu desempenho sobre outras malhas adaptativas.

Os algoritmos apresentados neste capítulo, extraídos de trabalhos anteriores, fornecerão subsídios para os capítulos seguintes, onde serão estudadas implementações dessas técnicas sobre triangulações retangulares adaptativas, explorando as vantagens oferecidas por essas estruturas.

Antes, porém, a próxima seção será dedicada à descrição do modelo que estará sendo utilizado para estabelecer como os dados são traduzidos em propriedades óticas.

5.1 Modelos Óticos

Os modelos óticos mais amplamente empregados em visualização direta consideram o volume como uma nuvem de partículas cuja densidade varia em função do campo escalar. Existem muitos trabalhos publicados sobre simulação da interação da luz com esse tipo de meio. [Blinn82] [Kajiya84] [Max86a] [Max86b] [Rushmeier87] são exemplos de abordagens que buscam obter imagens realísticas, incluindo efeitos de absorção, reflexão, transmissão, difusão (“scattering”) e sombra, resultando em modelos complexos e de custo computacional proibitivo. Diante disso, os caminhos seguidos em direção à viabilização de sistemas desse gênero foram a simplificação daqueles modelos e a criação de outros, despreocupados com a realidade visual, mas que fossem capazes de proporcionar ferramentas efetivas na caracterização das propriedades relevantes dos dados em estudo, e que permitissem implementações mais eficientes. Essa segunda

opção é bem aceita pelos usuários que, em muitas situações, valorizam mais o requisito de interatividade do ambiente operacional. Isso se confirma, em especial, no campo da visualização científica, uma vez que, em grande parte das aplicações, não existem referenciais visuais para o que se pretende observar.

A seção a seguir descreve o modelo que será considerado neste trabalho. Para uma abordagem mais detalhada, ver [Max95a]. Nesse mesmo trabalho, Nelson Max apresenta modelos mais sofisticados, incluindo sombra e reflexão/espalhamento múltiplos, além de fazer uma ampla revisão bibliográfica sobre o assunto.

5.2 O Modelo Adotado

Os algoritmos que serão apresentados ao longo deste texto levarão em conta, entre os citados acima, os efeitos de absorção e reflexão. Além disso, recorrendo ao modelo de iluminação proposto por P. Sabella [Sabella88], que introduziu a caracterização do campo como emissor de densidade, considerar-se-á que as partículas emitem brilho próprio, ao mesmo tempo que absorvem parte da energia que chega até elas.

Esses são os elementos mais frequentemente adotados no âmbito das aplicações científicas da visualização volumétrica direta. O modelo de Sabella fornece o mesmo resultado que se obteria considerando iluminação ambiente, em que cada partícula recebesse a mesma intensidade luminosa de todas as direções [Max90]. Observa-se ainda que o efeito de reflexão, incluído para enriquecer a visualização direta de isosuperfícies, tem implicações semelhantes ao de emissão. Como será mostrado adiante com mais detalhes, o modelo matemático para ambos os efeitos associa a cada partícula uma contribuição positiva à energia total percebida pelo observador.

[Max95a] desenvolve esse modelo a partir do caso limite, em que as partículas teriam tamanho infinitesimal. Se s é o parâmetro de distância ao longo de um raio que parte do fundo em direção ao observador, em cada segmento infinitesimal ds ocorrerá absorção, emissão e reflexão. As partículas são consideradas esferas idênticas de raio r e área projetada $A = \pi r^2$. Seja ρ o número de partículas por unidade de volume. Considerando uma fatia cilíndrica de espessura Δs e área da base B , seu volume será $B\Delta s$ e conterá $N = \rho B\Delta s$ partículas. Se Δs é suficientemente pequeno para que seja baixa a probabilidade de que as projeções das partículas sobre a base se sobreponham, pode-se aproximar a área total dessas projeções por $A_t = NA$. Então, a fração de B ocupada pelas projeções das partículas será $A_t/B = \rho A\Delta s$.

Nesse modelo, a capacidade do meio de absorver energia luminosa é definida estabelecendo que cada partícula absorve toda a luz que chega até ela. Assim, $\rho A\Delta s$ também representa a fração de energia luminosa que será absorvida. No limite, como Δs tende a zero, também tende a zero a probabilidade de superposição das projeções das partículas, resultando a equação diferencial

$$\frac{dI}{ds} = -\rho(s)AI(s) = -\tau(s)I(s)$$

onde $I(s)$ é a intensidade luminosa que alcança a posição s e $\tau(s) = \rho(s)A$ é chamado *coeficiente de atenuação*.

Quanto ao efeito de emissão, considera-se que as partículas brilham com intensidade C por unidade de área projetada, contribuindo com CA_t para a energia total que sai da fatia. O acréscimo de energia por unidade de área é $CA_t/B=C\rho A\Delta s$, resultando, no limite, a equação diferencial

$$\frac{dI}{ds} = C(s)\rho(s)A = C(s)\tau(s) = g(s)$$

onde $g(s)$ é chamado *termo fonte*.

Considerando simultaneamente ambos os efeitos, a equação é

$$\frac{dI}{ds} = g(s) - \tau(s)I(s)$$

cujas solução fornece, para a intensidade luminosa que alcança o observador, a expressão

$$I = I_0 e^{-\int_a^b \tau(t) dt} + \int_a^b g(s) e^{-\int_s^b \tau(t) dt} ds \quad (1)$$

onde I_0 corresponde à iluminação de fundo; a e b são os valores do parâmetro nos pontos em que o raio entra e sai do volume, respectivamente. O fator

$$e^{-\int_{s_1}^{s_2} \tau(s) ds}$$

é definido como *transparência* do meio entre as posições s_1 e s_2 . Assim, as expressões

$$T(s) = e^{-\int_a^s \tau(t) dt} \quad T'(s) = e^{-\int_s^b \tau(t) dt}$$

correspondem, respectivamente, à transparência do volume ao longo do raio desde a posição de entrada no volume até a posição s , e à transparência entre s e a saída do volume em direção ao observador. Pode-se, então, escrever, para maior clareza

$$I = I_0 T_v + \int_a^b g(s) T'(s) ds \quad (2)$$

onde T_v é a transparência do volume entre a e b , dada por

$$T_v = e^{-\int_a^b \tau(s) ds}$$

5.2.1 Funções de Transferência

Uma tarefa delicada que o usuário de um software de visualização volumétrica precisa realizar é a *classificação dos dados*. Nessa hora, os conhecimentos sobre o fenômeno em estudo e mesmo sobre a maneira como as imagens são “extraídas” dos dados originais, serão de grande importância para o bom aproveitamento dos recursos disponíveis nos programas desse gênero. Para uma técnica de visualização direta, classificar os dados significa escolher *funções de transferência* que mapeiem os valores escalares em propriedades óticas. As funções τ e C , mencionadas na seção anterior, denominadas respectivamente *coeficiente de atenuação* e *cromaticidade*, são exemplos de *funções de transferência*.

O tratamento feito até aqui considera a luz monocromática, calculando um único valor de intensidade luminosa, a partir de funções τ e C também únicas. No entanto, a mesma abordagem se estende a sistemas policromáticos, bastando repetir as equações obtidas para cada comprimento de onda relevante. Por exemplo, no sistema RGB são calculadas as intensidades I_r , I_g e I_b , a partir de seis funções de transferência τ_r , τ_g , τ_b ,

C_r , C_g e C_b . Genericamente, a expressão da intensidade luminosa para os diferentes comprimentos de onda, λ , fica sendo

$$I_\lambda = I_{0\lambda} T_{v\lambda} + \int_a^b g_\lambda(s) T'_\lambda(s) ds. \quad (3)$$

A utilidade de toda essa flexibilidade é discutível, uma vez que a escolha dessas funções é uma tarefa não trivial, devido à dificuldade de se prever o resultado de sua aplicação combinada. Por outro lado, o cálculo das três componentes de intensidade luminosa a partir das seis funções de transferência tem um custo computacional elevado. Assim, é usual adotar restrições que, apesar de diminuírem essa flexibilidade, proporcionam ganhos significativos de eficiência, sem reduzir, em termos práticos, o potencial de resolução da técnica, no que se refere a sua capacidade de explicitar as características relevantes do fenômeno sob análise.

A figura 5.1 mostra exemplos de funções de transferência, τ e C_r , para um campo escalar f . É usual defini-las como funções lineares por partes.

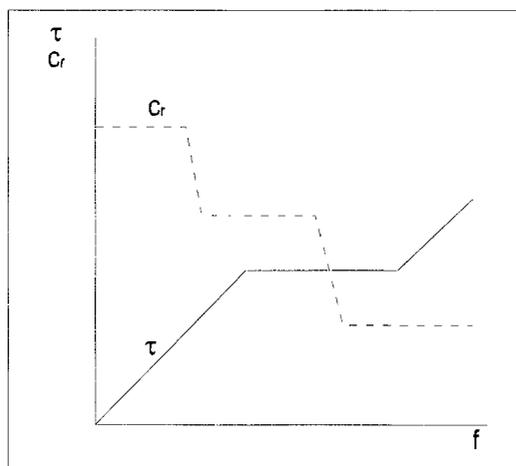


Figura 5.1 – Funções de transferência

5.2.2 Simplificações do Modelo

Uma primeira opção simplificadora é considerar a cromaticidade C constante ao longo do raio, para cada comprimento de onda. Assim, o termo fonte definido acima fica sendo $g(s) = C\tau(s)$ e a parcela em que ele está presente, na expressão da intensidade luminosa, assume a forma

$$\begin{aligned} \int_a^b g(s) e^{-\int_s^b \tau(t) dt} ds &= \int_a^b C\tau(s) e^{-\int_s^b \tau(t) dt} ds = \\ &= C \int_a^b \frac{d}{ds} e^{-\int_s^b \tau(t) dt} ds = C \left(1 - e^{-\int_a^b \tau(t) dt} \right) = C(1 - T_v) \end{aligned}$$

A expressão final, em sua forma mais concisa e com referência explícita aos diferentes comprimentos de onda, será

$$I_\lambda = I_{0\lambda} T_{v\lambda} + C_\lambda (1 - T_{v\lambda}). \quad (4)$$

A grandeza $\alpha=1-T$ é a *opacidade*, que também é uma característica do meio variável com o comprimento de onda.

Pode-se, ainda, dar mais um passo no sentido de diminuir o custo computacional do processo, considerando constante, além da cromaticidade, também o coeficiente de atenuação. Dessa forma, a transparência será dada por

$$T_{v\lambda} = e^{-\tau_\lambda l},$$

onde $l=b-a$.

As restrições de que a cromaticidade e/ou o coeficiente de atenuação sejam constantes ao longo do raio não influenciam tanto os resultados quanto pode parecer a princípio. Como o domínio é discretizado em células que são tratadas separadamente, para se conseguir o ganho de eficiência esperado basta considerar esses parâmetros constantes no interior de cada célula. A integração, que precisa ser feita em toda a extensão da interseção do raio com o volume, será avaliada em cada segmento do raio correspondente à sua interseção com uma célula. O resultado final será obtido pela “composição” [Porter84] desses resultados parciais através da equação (4). Dessa forma, aqueles parâmetros podem variar de célula para célula, sem perder as vantagens descritas acima.

As constantes C_λ e τ_λ para uma dada célula podem ser definidas, por exemplo, como as médias de seus valores nos pontos de interseção do raio com a célula, obtidos por interpolação linear nas faces correspondentes. Se τ_λ for uma função linear do campo escalar e esse campo variar também linearmente no interior da célula, essa opção fornecerá para transparência o mesmo valor que seria obtido pela integração. Isso já não é verdade para C_λ . Qualquer que seja essa função, desde que não seja realmente constante, aproximá-la por uma constante implicará em erro, uma vez que isso corresponde a dar à cor da porção mais próxima do observador o mesmo peso que à da porção mais distante. A cor da frente deveria ter maior peso, já que a opacidade ao longo do raio obscurece mais a cor do fundo.

Uma terceira consideração simplificadora é assumir que o coeficiente de atenuação seja independente do comprimento de onda. Com isso, a exponencial envolvida no cálculo da transparência será computada apenas um vez para cada célula.

Finalmente, um artifício numérico que pode ser aplicado, se τ_λ for considerado constante, baseia-se na expansão em série da função

$$f(x) = 1 - e^{-x} = x - x^2 / 2 + \dots$$

Como $\alpha=1-e^{-\tau l}$ pode-se aproximá-lo por $\alpha=\tau l - (\tau l)^2/2$, ou mesmo por $\alpha=\tau l$, se $\tau l \ll 1$.

Naturalmente, todas essas simplificações constituem imprecisões introduzidas nos resultados. As experiências mostram, no entanto, que vale a pena considerá-las, pelo menos como recurso para obtenção de um esquema iterativo eficiente.

5.2.3 Reflexão em Superfícies

De uma maneira geral, a introdução do efeito de reflexão no modelo ótico se dá através do termo fonte $g(s)$, que passa a ser

$$g(s) = C(s)\tau(s) + R(s)$$

onde $R(s)$ representa a intensidade de luz refletida, na direção do observador, ou seja, a contribuição para a intensidade luminosa do raio, devida à reflexão na posição s .

Para o modelo adotado neste trabalho, a inclusão desse efeito se justifica como recurso para enriquecer a imagem com a exibição de superfícies de nível.

Neste trabalho, o efeito de reflexão será considerado somente em pontos situados sobre a superfície desejada. Assim, a parcela $R(s)$ introduzida na expressão do termo fonte, só terá que ser calculada em relação a esses pontos. Isso pode ser feito de forma simples. Supondo que se queira exibir a isosuperfície de nível j , o procedimento de integração pode ser adaptado para checar a presença desse nível na célula a ser processada. Em caso positivo, a célula será tratada como se fosse dividida pela superfície. O segmento do raio contido no seu interior também será dividido, devendo ser computadas, separadamente, três contribuições: a do segmento situado atrás da superfície, a contribuição por reflexão na superfície e a do segmento frontal. A função de transferência relativa ao coeficiente de atenuação será dotada de um pulso de largura nula e altura τ_j (função delta), localizado, no eixo de abscissas, na posição correspondente ao valor escalar igual a j .

Outro aspecto em que se observam diferenças, do ponto de vista do tratamento do efeito de reflexão no modelo ótico para visualização volumétrica, é quanto à inclusão ou não de especularidade no modelo de iluminação. A opção mais simples é tratar a superfície como lambertiana, considerando apenas reflexão difusa. A inclusão de reflexão especular proporciona melhoria da qualidade da imagem, facilitando a observação de detalhes, mas também significa aumento de custo. A localização da fonte de luz no infinito e a utilização de projeção ortográfica proporcionam simplificações nos cálculos, que podem viabilizar a utilização de um modelo de iluminação mais completo. Também, pode-se considerar essa opção como um recurso a mais em um sistema de visualização, objetivando apresentações finais de melhor qualidade.

A contribuição da reflexão difusa para a intensidade luminosa é incluída no modelo de iluminação como

$$I_d = I_i k_d \cos \theta$$

onde I_i é a intensidade da luz incidente, k_d é o coeficiente de reflexão difusa e θ é o ângulo entre a direção da luz incidente e a normal à superfície. As grandezas envolvidas, com exceção do ângulo θ , são dependentes do comprimento de onda. O coeficiente de reflexão pode ser definido em função do coeficiente de atenuação ou simplesmente como uma constante. Ainda outra alternativa é calculá-lo em função do módulo do gradiente [Drebin88].

A normal à superfície em um ponto é dada pela direção do gradiente, que é determinada a partir da função de interpolação utilizada para definir os valores do campo escalar no interior das células. As funções de interpolação mais usadas não apresentam continuidade do gradiente na transição entre células, o que pode ser visível na imagem. Se isso for considerado inconveniente, pode-se optar por uma alteração nas normais de forma a se obter transições suaves. [Williams92a] sugere adotar para cada vértice um gradiente calculado como a média dos gradientes nos baricentros das células que compartilham esse vértice. A normal à superfície em um ponto no interior de uma célula seria obtida, então, por interpolação dos gradientes atribuídos aos vértices.

Esse esquema pode ser implementado de forma a tratar mais de um nível ao mesmo tempo, exibindo as isosuperfícies correspondentes em uma mesma imagem, como superfícies “concêntricas” em que as mais externas sejam semitransparentes.

5.3 Método de Lançamento de Raios

A maneira mais natural de se obter uma imagem de um conjunto de dados volumétricos é “lançar” raios do olho do observador através de cada pixel da tela onde a imagem deve ser formada. As posições do olho e de um pixel definem a direção de um raio, cuja trajetória através das células do volume é acompanhada por um procedimento que calcula a contribuição que cada célula fornece para a cor final do pixel, de acordo com o modelo ótico adotado. As sucessivas contribuições vão sendo “compostas” [Porter84] em ordem crescente da distância ao observador (frente para trás, FpT), até que o raio intercepte um objeto opaco, que a opacidade acumulada alcance um dado valor limite próximo de um ou até que o raio saia pela parte posterior do volume. Dessa forma, evita-se o trabalho de tratar as células mais distantes do observador, após ser alcançado o limite de opacidade.

Alternativamente, a composição pode ser realizada no sentido inverso, partindo do fundo em direção ao observador (trás para frente, TpF). Essa opção tem um pequeno ganho, em relação à anterior, porque aquele método exige o esforço extra de manutenção de uma variável para acumular opacidade (ou transparência). Em compensação, essa última alternativa não oferece a possibilidade de interromper o percurso em função do limite de opacidade. Os trechos de código a seguir, explicitam essas características [Max95a].

Algoritmo FpT:

```

I = 0; /* I - intensidade luminosa */
T = 1; /* T - transparência */
i = n; /* n - número de células */
while (T > limite_transp && i >= 1) {
    I = T*g[i] + I; /* g[i] - termo fonte da célula i */
    T = T*t[i]; /* t[i] - transparência da célula i */
    --i;
}
I = I + T*I0; /* I0 - intensidade luminosa de fundo */

```

Algoritmo TpF:

```

I = I0;
for (i=1; i<=n; ++i)
    I = t[i] * I + g[i];

```

Os algoritmos seqüenciais de lançamento de raio realizam seu trabalho pixel a pixel, normalmente na ordem das linhas de varredura da imagem, o que justifica referenciá-los como algoritmos baseados no espaço da imagem. Entretanto, pode ser de interesse processar os pixels em ordem aleatória, permitindo ao usuário ter uma visão

geral mais imediata e acompanhar o refinamento da imagem à medida em que os pixels em falta vão sendo calculados.

Existem vários algoritmos publicados que implementam essa técnica. [Levoy88], [Upson88], [Levoy90a] e [Levoy90b] tratam da aplicação da técnica a malhas regulares. Para malhas irregulares, [Garrity90] descreve o algoritmo básico, enquanto [Giertsen92] e [Silva96a] apresentam implementações que buscam ganhar eficiência explorando de maneira mais ampla a coerência existente no espaço do objeto, usando algoritmos de varredura. Finalmente, ainda para malhas irregulares, [Giertsen93], [Ma95] e [Silva96b] propõem estratégias de subdivisão do processamento em arquiteturas paralelas.

5.3.1 Algoritmo Básico

A implementação do método de lançamento de raios para malhas irregulares, proposta por Garrity em [Garrity90], inicia por um pré-processamento em que são identificadas as faces externas do volume. Para cada raio lançado é necessário determinar, entre essas faces, qual é interceptada primeiro, o ponto de interseção, bem como a célula a que essa face pertence. Determina-se, então, a outra interseção do raio com essa célula, ou seja, a face e ponto de saída. De posse dessas informações, pode-se calcular a contribuição dessa célula para a cor do pixel associado ao raio. A estrutura de dados usada para representar a malha deve prover informações de vizinhança que permitam identificar facilmente a célula seguinte em que o raio penetra. Como essas células têm uma face em comum, os cálculos realizados anteriormente são aproveitados, de modo que apenas uma interseção precisa ser calculada para cada célula, com exceção da primeira, naturalmente. Esse procedimento continua até que outra face externa seja encontrada. Nesse estágio, se a malha é convexa, o raio terá abandonado definitivamente o volume, finalizando seu processamento; senão, será necessário verificar se alguma outra face externa é interceptada, por onde o raio estaria penetrando novamente o volume.

Em seu trabalho, Garrity descreve uma maneira prática para determinar essas interseções, válida para qualquer célula poliédrica convexa. No entanto, como estamos tratando especificamente de tetraedros, esses cálculos podem ser bastante simplificados com a utilização de *coordenadas baricêntricas* [Persiano96] (ver Apêndice A). Além da vantagem na determinação de interseções, essa abordagem também simplifica o cálculo da contribuição da célula para a cor do pixel, uma vez que as coordenadas baricêntricas do ponto de interseção do raio com a face representam os pesos dos vértices da face em uma interpolação linear relativa àquele ponto [Farin90]. Os cálculos envolvidos encontram-se detalhados nas próximas seções.

Determinação da Face de Entrada

Para reduzir o custo de determinação da primeira célula (e conseqüentemente da primeira face) interceptada por um raio é conveniente implementar um mecanismo de classificação geométrica das faces externas. Isso pode ser feito, por exemplo, através de estruturas auxiliares tais como uma grade regular ou uma estrutura hierárquica como “octree”. No entanto, acreditamos ser mais simples e até mais eficiente explorar a

própria estrutura topológica da malha para realizar essa determinação, levando em conta que a coerência espacial entre pixels tem alta probabilidade de se refletir como coerência espacial entre os pontos de entrada dos respectivos raios no volume. Se o presente raio corresponde a um pixel vizinho do anterior, a primeira célula interceptada por ele provavelmente será a mesma ou uma vizinha daquela por onde seu antecessor penetrou no volume. Obviamente, esse esquema não se aplica se os pixels forem tratados em ordem aleatória, conforme alternativa apresentada anteriormente.

Interseções com Faces Internas

Ao longo de sua trajetória, que será denominada *percurso radial*, um raio lançado intercepta uma seqüência de células da estrutura. A identificação de cada uma dessas células e a determinação de seus pontos de interseção com o raio formam, juntamente com o cálculo de sua contribuição para a cor do pixel, um núcleo de operações que constituem o procedimento de maior peso do método, do ponto de vista do custo computacional. Esta seção apresenta uma forma eficiente de realizar essa tarefa para células tetraédricas, válida tanto para projeção paralela quanto para perspectiva.

Se **A**, **B**, **C** e **D** são pontos não coplanares do espaço euclidiano ϵ^3 , formando um simplexo (tetraedro) não degenerado, qualquer ponto **P** do espaço pode ser representado relativamente a esses quatro pontos de referência por uma quádrupla (a, b, c, d) de números reais (Apêndice A), tais que

$$a + b + c + d = 1 \text{ e}$$

$$\mathbf{P} = a\mathbf{A} + b\mathbf{B} + c\mathbf{C} + d\mathbf{D}.$$

Esses números são denominados *coordenadas baricêntricas* de **P** em relação àquele tetraedro.

Uma direção $\mathbf{L} = \mathbf{Q} - \mathbf{P}$, onde $\mathbf{Q}, \mathbf{P} \in \epsilon^3$, também se representa por uma quádrupla de números reais (a_L, b_L, c_L, d_L) , tais que

$$a_L + b_L + c_L + d_L = 0 \text{ e}$$

$$\mathbf{L} = a_L\mathbf{A} + b_L\mathbf{B} + c_L\mathbf{C} + d_L\mathbf{D}.$$

Supondo que **L** seja a direção de um raio e que já seja conhecido o ponto **P**, primeira interseção desse raio com uma célula, a questão a ser resolvida é determinar $\mathbf{Q} = \mathbf{P} + \lambda\mathbf{L}$, ponto de interseção do raio com a face por onde ele sai do tetraedro (figura 5.2).

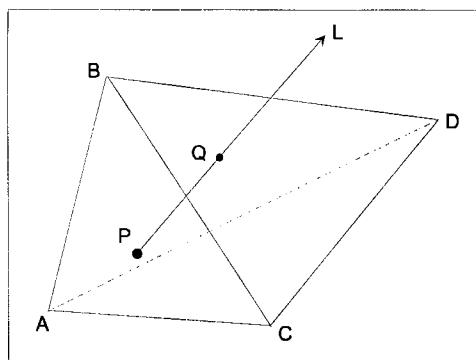


Figura 5.2 – Interseções de um raio com as faces de um tetraedro

Como não se sabe qual é a face de saída, é necessário considerar todas elas, exceto, evidentemente, a face de entrada. Escolhendo novamente a face A como exemplo, a interseção com o plano dessa face é o ponto do raio que tem coordenada a nula, ou seja,

$$a_Q = a_P + \lambda a_L = 0,$$

ou ainda, o ponto correspondente ao parâmetro

$$\lambda = -\frac{a_P}{a_L}.$$

Observando que, no exemplo da figura 5.2, a face de entrada é D, o valor de λ referente à interseção com a face de saída será dado por

$$\lambda = \text{o mínimo entre os valores positivos em } \left\{ -\frac{a_P}{a_L}, -\frac{b_P}{b_L}, -\frac{c_P}{c_L} \right\}.$$

De posse das coordenadas baricêntricas de **P** e **Q** e do parâmetro λ , que corresponde à distância entre esses dois pontos, pode-se fazer as interpolações e calcular a contribuição do tetraedro para a cor do pixel, conforme o modelo ótico adotado. Na verdade, com exceção da primeira célula interceptada pelo raio, apenas as interpolações relativas a **Q** serão necessárias, aproveitando-se os resultados da célula vizinha relativos a **P**.

Transição entre células

Seguindo seu percurso, ao sair da célula que acaba de ser processada, o raio penetra uma nova célula que deverá receber o mesmo tratamento. Sejam t e t' a antiga e a nova célula, respectivamente. Como para t , as informações geométricas necessárias para processar t' são as coordenadas baricêntricas do ponto de entrada na nova célula e da direção do raio. Essas informações precisam estar representadas como coordenadas relativas a t' , simplexo que define intrinsecamente um novo sistema de coordenadas. Essa transição, que corresponde a uma mudança de sistema de coordenadas, é simplificada pelo fato de que t e t' têm uma face em comum (ver Apêndice A).

Designando por **E** o quarto vértice do novo tetraedro, o primeiro passo dessa mudança é determinar as coordenadas desse ponto em relação a t . Sejam a_E , b_E , c_E e d_E essas coordenadas, ou seja, mantendo as letras **A**, **B**, **C** e **D** para representar os vértices de t , **E** pode ser expresso como

$$\mathbf{E} = a_E \mathbf{A} + b_E \mathbf{B} + c_E \mathbf{C} + d_E \mathbf{D}.$$

Como mostra o Apêndice A, a determinação dessas coordenadas envolve o cálculo de volumes de tetraedros, o que é uma operação bastante cara, se se considerar a frequência com que ela precisa ser realizada nesse processo. No entanto isso pode ser feito como pré-processamento, guardando os valores obtidos associados às faces. O custo desse armazenamento é de apenas quatro números reais por face, uma vez que as coordenadas baricêntricas de um ponto **A** em relação a um tetraedro **BCDE** são facilmente calculadas a partir das coordenadas do ponto **E** em relação ao tetraedro **ABCD**. Essas relações são apresentadas a seguir e sua dedução também se encontra no Apêndice A.

Sejam

$$\mathbf{E} = a_E \mathbf{A} + b_E \mathbf{B} + c_E \mathbf{C} + d_E \mathbf{D} \quad \text{e} \quad \mathbf{A} = b_A \mathbf{B} + c_A \mathbf{C} + d_A \mathbf{D} + e_A \mathbf{E}.$$

Conhecendo-se as coordenadas a_E , b_E , c_E e d_E , as coordenadas b_A , c_A , d_A e e_A são dadas pelas expressões:

$$b_A = -\frac{b_E}{a_E} \quad c_A = -\frac{c_E}{a_E} \quad d_A = -\frac{d_E}{a_E} \quad e_A = \frac{1}{a_E}.$$

Dando seqüência à transição entre células, para obter a representação da direção \mathbf{L} no novo sistema, pode-se escrever:

$$\mathbf{L} = a_L \mathbf{A} + b_L \mathbf{B} + c_L \mathbf{C} + d_L \mathbf{D} = \bar{b}_L \mathbf{B} + \bar{c}_L \mathbf{C} + \bar{d}_L \mathbf{D} + \bar{e}_L \mathbf{E}.$$

Substituindo \mathbf{E} por sua expressão em função de \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} , vem

$$\mathbf{L} = \bar{e}_L a_E \mathbf{A} + (\bar{b}_L + \bar{e}_L b_E) \mathbf{B} + (\bar{c}_L + \bar{e}_L c_E) \mathbf{C} + (\bar{d}_L + \bar{e}_L d_E) \mathbf{D}.$$

Assim, as coordenadas da direção em relação ao novo tetraedro são

$$\bar{e}_L = \frac{a_L}{a_E} \quad \bar{b}_L = b_L - \bar{e}_L b_E \quad \bar{c}_L = c_L - \bar{e}_L c_E \quad \bar{d}_L = d_L - \bar{e}_L d_E.$$

Finalmente, essa mudança se completa simplesmente considerando que o ponto \mathbf{P} , por onde o raio entra na nova célula, é o mesmo ponto \mathbf{Q} , por onde ele sai da antiga. Sendo assim, para obter a definição desse ponto em relação ao novo tetraedro, basta rearranjar suas coordenadas de acordo com as relações de vizinhança estabelecidas na seção 3.6.

Os cálculos envolvidos nessa mudança de sistema de coordenadas podem introduzir erros acumulativos na representação da direção. Para evitar maiores prejuízos decorrentes disso, pode-se adotar a medida de recalculá-la periodicamente a partir de sua definição original. Considerando que a direção tenha sido definida pela diferença entre dois pontos, essa medida é implementada calculando as coordenadas baricêntricas desses pontos em relação a um tetraedro intermediário, encontrado no percurso, e recalculando sua diferença.

5.3.2 Degenerações

É importante ressaltar a necessidade de tratar os casos de degeneração, que podem ser interseções degeneradas ou células degeneradas. O primeiro tipo ocorre quando o raio passa muito próximo a uma aresta ou a um vértice da malha. Nesse caso, a comparação do parâmetro de distância pode levar a erro numérico e à escolha incorreta da face de saída. O segundo tipo ocorre quando existe célula com dois vértices coincidentes ou com um vértice no interior de uma face à qual ele não pertence. Para malhas simpliciais, uma célula degenerada é um tetraedro com os quatro vértices coplanares, o que desqualifica o critério de distância a interseções com planos de faces para identificação da face de saída. Essas anomalias podem induzir a erros na determinação de células interceptadas. Sugestões para tratá-las são oferecidas no trabalho de Garrity.

5.3.3 Coordenadas Baricêntricas X Coordenadas Cartesianas

Esta seção relaciona alguns fatos que justificam a opção pela abordagem baseada em coordenadas baricêntricas para implementar o método de lançamento de raios sobre malhas simpliciais. Em seu trabalho original [Garrity90], voltado para malhas de poliedros convexos quaisquer, o autor determina a face de saída calculando as interseções entre raio e planos com base em suas coordenadas cartesianas. Dessa forma, a avaliação do parâmetro de distância que permite identificar a interseção mais próxima envolve, para cada plano, uma quantidade de operações de ponto flutuante acima de 2,5 (mais precisamente 33/12) vezes maior do que o necessário na abordagem adotada neste trabalho. Isto ocorre mesmo na hipótese mais econômica daquela versão, qual seja a que considera conhecidas as normais das faces, e mesmo levando em conta as operações de mudança de base inerentes à versão com coordenadas baricêntricas.

Um segundo aspecto a ser ressaltado está ligado aos casos de degeneração já citados. As soluções propostas em [Garrity90] para tratar essas exceções, baseadas em coordenadas cartesianas, também envolvem operações pesadas com vetores (vários produtos escalares) para obter outros parâmetros que confirmem que a interseção ocorre dentro de uma determinada face. Essa verificação é praticamente gratuita em coordenadas baricêntricas: basta verificar se as coordenadas da interseção, relativas aos vértices da face, são todas não negativas.

Vale ainda lembrar, para completeza dessas observações, o fato já assinalado anteriormente, de que as coordenadas baricêntricas da interseção são também os coeficientes da interpolação linear entre os vértices da face, operação necessária à avaliação da contribuição da célula para a cor do pixel. Novamente, o cálculo desses coeficientes representa um custo significativo dentro do processo.

Finalmente, o lado negativo da abordagem com coordenadas baricêntricas. Foi apresentada na seção anterior a sugestão de armazenar, associadas a cada face, as coordenadas baricêntricas (quatro números reais) do quarto vértice de um dos tetraedros a que ela pertence, em relação ao outro tetraedro. Esta medida visa economizar operações aritméticas com números reais no cálculo de interseções. Esta proposta constitui um acréscimo, ainda que pequeno, no consumo de memória em relação ao método sugerido por Garrity. Em seu trabalho original, esse autor propõe o armazenamento da normal ao plano da face (três números reais por face) para facilitar a determinação de sua interseção com o raio. A diferença é, portanto, de um número real por face.

5.4 Método de Projeção

Enquanto no método de lançamento de raios o processamento se dá pixel a pixel, o método de projeção caracteriza-se pelo processamento célula a célula, ordenadas segundo um critério conveniente. Para cada célula tratada determinam-se os pixels sobre os quais ela se projeta e a sua contribuição para a cor e opacidade de cada um deles. O interesse que essa abordagem desperta é motivado, principalmente, por duas características que ela apresenta. Em primeiro lugar, permite uma exploração mais efetiva da coerência espacial, caracterizada pela projeção de uma célula sobre vários

pixels vizinhos. Isso proporciona algoritmos eficientes e oportunidades de tirar vantagem de técnicas de computação gráfica implementadas em hardware. Em segundo lugar, quando as células são projetadas em ordem decrescente de sua distância ao observador, a imagem é gerada por camadas que vão se superpondo, permitindo uma melhor observação, ainda que temporária, de regiões que ficam parcialmente ocultadas na imagem final.

O aproveitamento dessas vantagens potenciais, no entanto, depende de alguns fatores. A eficácia da exploração dessa forma de coerência espacial é tanto maior quanto maior for o número de pixels cobertos pela projeção de cada célula. Assim sendo, os maiores benefícios serão obtidos com a aplicação do método a volumes de dados de porte médio, ou em tomadas parciais ampliadas de volumes de maior porte (“zoom”). Outros aspectos que necessitam análise, e que receberão mais atenção adiante, são imprecisões devidas à utilização de hardware gráfico e à eventual impossibilidade de ordenar por visibilidade as células de determinadas malhas.

Vários trabalhos dedicados a esse método têm sido publicados. No campo das aplicações de malhas regulares, ressaltam-se [Westover90], [Wilhelms91] e [Lacroute94].

Quanto a algoritmos voltados para malhas irregulares, [Max90] e [Shirley90] são dois trabalhos fundamentais. O primeiro considera o volume subdividido em células poliédricas quaisquer, desde que convexas, e apresenta um procedimento para rasterização de cada célula, avaliando a contribuição para cada pixel através de integração analítica. Esse procedimento, descrito no artigo para o caso de projeção ortogonal, rasteriza duas vezes a região referente à projeção da célula: uma considerando as faces visíveis da célula, e outra para as faces invisíveis. Em cada uma dessas fases calcula-se, por interpolação linear, o valor da função escalar e do parâmetro do raio no ponto em que ele intercepta as faces envolvidas. A partir dos dados assim obtidos, pode-se computar o valor do pixel, considerando o modelo ótico descrito acima. Esse método produz imagens de alta precisão, mas o seu custo computacional é muito elevado, o que o torna pouco atrativo.

O trabalho de Shirley e Tuchman [Shirley90] substitui a rasterização das células volumétricas pela rasterização de polígonos, que é um processo muito mais barato e já com implementações em hardware disponíveis comercialmente. A idéia é imitar o efeito das células poliédricas através de polígonos transparentes que preenchem a região ocupada pela projeção da célula. Cada um desses polígonos, que tem em comum com cada vizinho apenas uma aresta de contato, recebe em seus vértices parâmetros de cor e opacidade determinados de forma a que sua inserção no processo de composição da imagem tenha um efeito similar ao que seria produzido pela parte da célula que ele representa. Essa abordagem, que é estendida e aprimorada nos trabalhos de Williams [Williams92a], Max/al [Max93] e Stein/al [Stein94], é um dos alicerces deste trabalho, pelo que receberá atenção especial nas seções a seguir.

Uma terceira linha de abordagem para métodos de projeção aplicados a malhas irregulares é devida a Yagel/al [Yagel96b]. Para uma dada direção de visualização, os vértices da malha são transformados para o espaço da imagem. Depois a malha é interceptada por uma série de planos paralelos ao plano de projeção, resultando em malhas bidimensionais de polígonos que são, então, rasterizados e compostos para produzir a imagem final. O uso intensivo de recursos de hardware gráfico, nas etapas de

transformação, rasterização e composição, torna a técnica muito eficiente. São apresentadas, também, extensões do método provendo características de adaptatividade e permitindo geração de imagens com qualidade progressiva. Vale a leitura do relatório técnico [Yagel93], que apresenta alguns esquemas interessantes relativos à implementação voltada para células tetraédricas, que não foram mantidos na versão definitiva do artigo.

5.4.1 Algoritmo de Projeção de Tetraedros

O algoritmo proposto em [Shirley90] considera células tetraédricas, dentro das quais a função escalar é interpolada linearmente, e realiza os seguintes passos:

1. *projetar os vértices da triangulação*
2. *ordenar os tetraedros segundo a visibilidade*
3. *classificar cada tetraedro de acordo com o perfil de sua projeção*
4. *decompor as projeções dos tetraedros em triângulos*
5. *calcular valores de cor e opacidade para os vértices dos triângulos*
6. *rasterizar os triângulos, na ordem estabelecida no passo 2, interpolando linearmente os valores de cor e opacidade dos vértices.*

Essas etapas são discutidas com mais detalhes a seguir. A abordagem estará voltada para projeção paralela, nos aspectos em que for relevante distinguir da projeção em perspectiva. Este enfoque está em consonância com o objetivo perseguido neste trabalho, que é comparar o desempenho de técnicas implementadas sobre triangulações genéricas em relação ao desempenho alcançado em triangulações RA. A projeção em perspectiva não oferece expectativa de ganhos significativos no algoritmo de projeção de tetraedros. Como será visto com mais detalhes em capítulo específico, a eficiência dessa técnica na projeção paralela de triangulações RA se deve à regularidade da geometria das projeções das células desse tipo de estrutura. Essas projeções se enquadram em um pequeno número de casos, a menos de um fator de escala, o que viabiliza a construção de tabelas. Essa característica não está presente no caso de projeção em perspectiva, onde uma mesma célula tem o formato de sua projeção modificado quando sua posição em relação ao centro de projeção é alterada.

Passo um : Projeção dos vértices da triangulação

calcular as coordenadas dos vértices da malha no sistema de coordenadas do observador X_{ob}, Y_{ob}, Z_{ob} {transformação de visualização} projetar esses vértices no plano X_{ob}, Y_{ob} , paralelamente ao eixo Z_{ob} (direção de observação)

Passo dois : Ordenação das células

ordenar os tetraedros da triangulação de trás para a frente

Esta fase será discutida com mais detalhes em seção específica, onde serão apresentadas abordagens alternativas.

Passo três : Classificação dos tetraedros

para cada tetraedro

marcar faces visíveis (+), faces invisíveis (-) e faces paralelas à direção de projeção (0)

identificar o caso em que o tetraedro se enquadra, conforme a figura 5.3

se 2 faces são paralelas à direção de projeção então

caso 4

senão se 1 face é paralela então

caso 3

senão se 2 faces são visíveis então

caso 2

senão

caso 1

A ordem dos testes nesse algoritmo obedece apenas a conveniência de expressão textual. A ordem para implementação efetiva deve dar preferência ao caso dois, que é o mais freqüente, seguido do caso um, do caso três, que ocorre muito poucas vezes em uma triangulação genérica, e finalmente do caso quatro, cuja ocorrência tem baixíssima probabilidade.

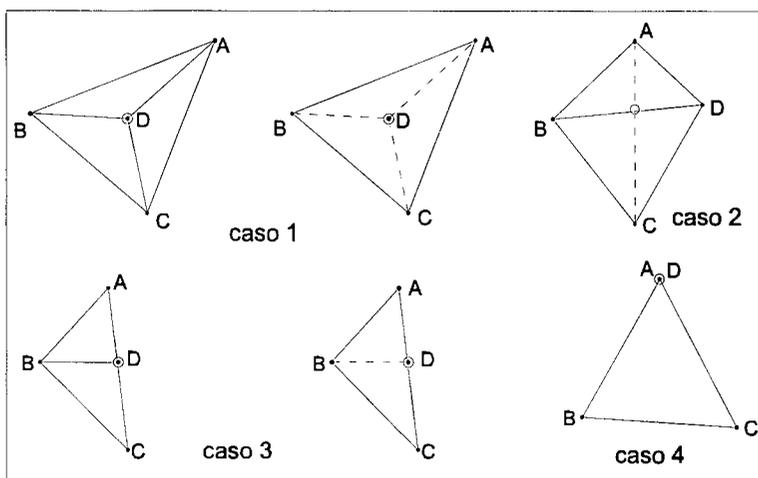


Figura 5.3 – As 4 condições possíveis para a projeção de um tetraedro

Observa-se que em cada um dos casos acima, apenas o vértice da figura plana marcado com círculo corresponde a uma espessura não nula no tetraedro. Por isto aquele vértice será designado por *vértice ENN*.

Passo quatro : Decomposição das projeções dos tetraedros

Este passo será descrito como algoritmo 5.1, pela necessidade de se fazer referência a ele mais adiante.

para cada tetraedro

sejam P1 e P2 os pontos do tetraedro delimitantes do segmento correspondente que se projeta sobre o vértice ENN

se caso 1 então

*{P1 é o vértice D do tetraedro e P2 pertence à face oposta}
calcular as coordenadas baricêntricas de P2 em relação a
essa face*

calcular $Z_{ob,P2}$ e $f(P2)$, por interpolação linear na face

senão se caso 2 então

*determinar o vértice ENN, interseção dos segmentos AC e
BD na figura plana*

*calcular as coordenadas baricêntricas do vértice ENN em
relação às arestas AC e BD do tetraedro*

*calcular $Z_{ob,P1}$, $f(P1)$, $Z_{ob,P2}$ e $f(P2)$, por interpolação linear
nas arestas AC e BD, respectivamente*

senão se caso 3 então

*{P1 é o vértice D do tetraedro e P2 pertence à aresta AC}
calcular as coordenadas baricêntricas de P2 em relação a
essa aresta*

calcular $Z_{ob,P2}$ e $f(P2)$, por interpolação linear na aresta

senão

*{caso 4 → P1 e P2 são os vértices com projeções
coincidentes}*

calcular a distância entre P1 e P2 $\{d_{P1-P2} = |Z_{ob,P1} - Z_{ob,P2}|\}$

Algoritmo 5.1

Quanto ao quinto passo do algoritmo, o método assume que é aceitável a interpolação linear de cor e opacidade dentro dos triângulos resultantes da decomposição dos tetraedros. Assim, basta determinar os valores dessas grandezas nos vértices desses triângulos, ficando a interpolação para a etapa de rasterização.

Passo cinco : Cálculo de cor e opacidade nos vértices dos triângulos

para cada tetraedro

{vértice ENN}

para cada comprimento de onda λ

$$C_{\lambda,ENN} = C_{\lambda}((f(P1)+f(P2))/2)$$

$$\tau_{ENN} = \tau((f(P1)+f(P2))/2)$$

$$\alpha_{ENN} = 1 - \exp(-\tau_{ENN}d_{P1-P2})$$

{demais vértices (v) da projeção}

para cada comprimento de onda λ

$$C_{\lambda} = C_{\lambda}(f(v))$$

$$\alpha = 0$$

Passo seis : Rasterização dos triângulos

para cada triângulo {percurso de trás para a frente}

para cada pixel

interpolando linearmente C e α

$$I_{\text{novo}} = I_{\text{velho}}(1-\alpha) + C\alpha \quad \{I \rightarrow \text{intensidade luminosa}\}$$

Esse último passo deve ser realizado preferencialmente por hardware gráfico.

5.4.2 Mapeamento de Textura

O algoritmo de projeção de tetraedros, na sua forma original descrita na seção anterior, constitui um método aproximativo, uma vez que a interpolação linear para a opacidade dentro dos triângulos fornece, para cada pixel, um valor diferente do valor exato que seria obtido pela integração ao longo do segmento que se projeta sobre o pixel. Além de inexata, a imagem produzida apresenta o efeito conhecido como bandas de Mach [Gomes94], como mostram Stein/al [Stein94]. Esse efeito se manifesta na fronteira entre regiões de uma imagem, onde ocorra descontinuidade da função de intensidade luminosa ou de sua derivada, e deve-se a uma tendência do sistema visual humano de acentuar as diferenças entre características óticas de regiões vizinhas. Neste caso ele ocorre ao longo das projeções das arestas das células, tornando-as visualmente destacadas e fazendo com que a subdivisão em tetraedros seja evidenciada. No caso de uma face paralela à direção de observação, esse efeito fica ainda mais acentuado.

O problema é devido, em parte, à linearidade da função no interior de cada tetraedro, estabelecida pelo modelo proposto para o campo, o que em geral já é suficiente para provocar a descontinuidade de derivada nas fronteiras entre células. Essa condição é herdada pelas propriedades óticas mapeadas sobre o campo. No entanto, o citado trabalho de Stein/al mostra que o algoritmo provoca esse tipo de distorção, mesmo que as funções cromaticidade e coeficiente de atenuação sejam constantes em todo o volume. Aqueles autores revêm o método e propõem uma alteração que leva a uma melhoria significativa na qualidade das imagens obtidas, sem perder as vantagens da rasterização dos triângulos por hardware. O objetivo que se persegue é fazer a rasterização de cada triângulo utilizando, para cada pixel que ele cobre, o valor correto de opacidade referente à influência do tetraedro correspondente, evitando a interpolação linear da opacidade nos triângulos.

Aproveitando a notação usada acima e designando por $Q1$ e $Q2$ os pontos extremos do segmento interseção entre o tetraedro e o raio que se projeta sobre um pixel, a opacidade será dada por

$$\alpha = 1 - e^{-\bar{\tau}d_{Q1-Q2}}$$

onde

$$\bar{\tau} = \tau \left(\frac{f(Q1) + f(Q2)}{2} \right).$$

Como a função exponencial não é implementada normalmente no hardware das plataformas gráficas, a introdução dessa expressão no processo é feita indiretamente. A

idéia explorada por Stein/al foi construir uma tabela da função de opacidade e inserí-la no mecanismo de mapeamento de textura do hardware gráfico [Watt93] [Neider93].

Nos casos em que for razoável considerar τ constante no interior de um tetraedro, constrói-se uma tabela unidimensional da função $1 - e^{-u}$, onde a coordenada $u = \tau d$. Como a distância d , comprimento do segmento, interno ao tetraedro, do raio que incide sobre o *vértice ENN*, varia linearmente dentro de cada triângulo, o valor de u também terá variação linear, podendo ser interpolado pelo hardware. Os valores atribuídos aos vértices do triângulo são estabelecidos como na seção anterior: $u = \tau d_{p1-p2}$ para o *vértice ENN* e $u = 0$, para os demais.

Se se considerar τ variando linearmente no tetraedro, o produto τd será uma função quadrática em cada triângulo. Para conservar a vinculação com o mecanismo de interpolação linear, uma vez que interpolação quadrática ainda não é comumente implementada em hardware, constrói-se uma tabela bidimensional da função $1 - e^{-\tau d}$, onde as coordenadas τ e d são interpoladas pelo hardware antes de serem usadas como índices para consultar a tabela.

As figuras 5.4 e 5.5 mostra imagens ilustrativas do diferencial de qualidade proporcionado pela utilização de mapeamento de textura no método de projeção. São imagens de um arranjo 2x2x2 de cubos, divididos em tetraedros, extraídas de [Stein94]. A primeira foi obtida com aplicação de interpolação linear das opacidades dos vértices dos triângulos, deixando nítidas as bandas de Mach, conforme previsto. Além de evidenciarem o erro de aproximação embutido nos cálculos, essas estruturas lineares podem prejudicar a visualização de características relevantes do fenômeno em estudo. Já a Segunda imagem, cuja única diferença metodológica em relação à primeira é a utilização de mapeamento de textura para melhorar a precisão do cálculo dos valores de opacidade, é nitidamente superior.

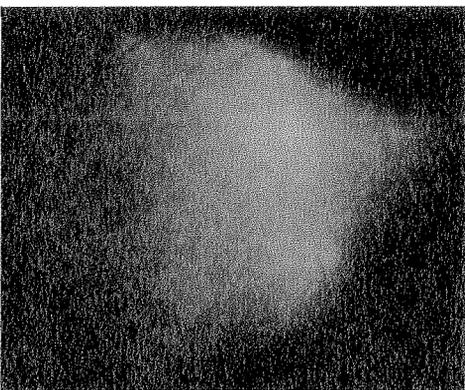


Figura 5.4 – Interpolação linear de opacidade.

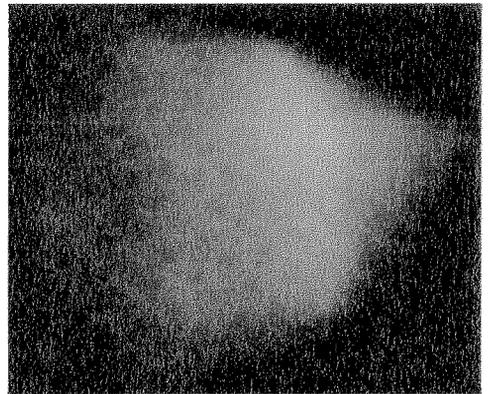


Figura 5.5 – Cálculo de opacidade com mapeamento de textura.

5.4.3 Considerações Sobre o Modelo Ótico no Método de Projeção

Ainda esse mesmo trabalho de Stein/al, mostra que com o algoritmo de projeção de tetraedros pode-se empregar um modelo ótico mais elaborado, com um acréscimo relativamente pequeno no custo computacional. Em lugar da cromaticidade constante em cada tetraedro, como estabelecido na forma simplificada do modelo ótico adotado neste trabalho, pode-se considerar um comportamento linear para a cromaticidade no interior de uma célula. Conforme já foi mencionado, a adoção de um valor constante de cromaticidade para o segmento interseção de um raio com uma célula, igual à média dos valores nos extremos desse segmento, é uma medida aproximativa que introduz erro na imagem resultante. Esse erro se deve à opção, implícita nessa estratégia, de desprezar o fato de que a opacidade atenua mais a influência da cor do fundo do que a da cor da frente do segmento. [Williams92b] faz um tratamento mais rigoroso dessa questão, de forma a obter a expressão analítica correta da contribuição de um tetraedro para a intensidade luminosa de um pixel, considerando lineares, dentro de cada célula, tanto as funções de transferência quanto a própria função escalar sob investigação. Porém, essa expressão é de computação pesada o que torna inviável sua utilização para cada pixel em um sistema interativo. Os resultados obtidos por aqueles autores mostram, no entanto, que um acréscimo significativo de qualidade da imagem pode ser obtido pela simples aplicação dessa expressão ao *vértice ENN*, conservando a interpolação linear por hardware para o interior dos triângulos.

5.4.4 Ordenação das Células

Na descrição do método de projeção para visualização volumétrica direta foi colocada a necessidade de ordenar as células por visibilidade. O processamento (projeção) de uma célula K só deve ser feito depois de já terem sido processadas todas aquelas cujas projeções têm interseção com a de K e que estejam mais distantes do observador do que K . Esse requisito está associado com a operação de composição das contribuições de cor e opacidade das células.

Pode-se definir ordenação por visibilidade em relação a um ponto de observação ou a uma direção de observação. Esta última, embora menos natural, porque um observador real sempre está localizado em um ponto do espaço, não é menos usual em sistemas de visualização, devido à importância de projeções paralelas em muitas aplicações, particularmente em visualização científica. Neste texto, como já mencionado, o estudo do método de visualização por projeção está enfocado em projeção paralela, por sua conveniência diante das triangulações retangulares adaptativas, como será mostrado no capítulo próprio. Isto dirige esta análise para ordenação segundo uma direção de observação ou *ordenação direcional*.

Formalmente, dada uma direção d definida no espaço tridimensional, a ordenação direcional induzida por d é uma relação \succ tal que duas células C^1 e C^2 satisfazem $C^1 \succ C^2$ se o interior de C^1 e uma translação de C^2 por λd são disjuntos para qualquer $\lambda > 0$. Em outras palavras, $C^1 \succ C^2$ se C^2 se projeta na direção d atrás de C^1 . Um *percurso ordenado* de uma malha em uma direção d é qualquer percurso de suas células, tal que cada célula é visitada antes de qualquer outra maior (de acordo com a relação \succ).

Uma ordenação direcional para uma malha regular é facilmente definida e implementada. O mesmo não ocorre com as malhas irregulares, que exigem algoritmos bem mais sofisticados. Encontram-se na literatura alguns trabalhos que propõem soluções para esse problema⁴, entre os quais ressaltamos [Max90], [Williams92c], [Williams98] e [Cignoni98].

Os dois primeiros artigos têm uma linha de abordagem semelhante, tratando de malhas conexas, convexas e acíclicas (ver adiante), compostas por poliedros convexos de faces planas. Outra característica comum aos dois algoritmos é a exigência de que a vizinhança entre duas células da malha seja definida por uma face (inteira) comum a ambos os poliedros. A idéia aplicada em ambos os trabalhos começa pela construção de um grafo de adjacências das células da malha. Dada a direção de observação, define-se a orientação para cada aresta do grafo partindo da célula obstruída para a célula visível. Isso pode ser feito tomando como referência o plano da face compartilhada pelas duas células correspondentes à aresta. Esse plano define dois semi-espacos e a aresta estará orientada no sentido da célula em cujo semi-espaco o observador está situado. A ordem das células resulta, então, da ordenação topológica desse grafo. O grafo de adjacências e as equações dos planos das faces são computados em uma fase de pré-processamento e armazenados com a malha. Todas as demais fases do processo, repetidas cada vez que os parâmetros de visualização são redefinidos, são realizadas em tempo linear em relação ao tamanho da malha. [Williams92c] mostra que o espaco de armazenamento exigido também tem relação linear com o tamanho da malha.

[Cignoni98] observa que a implementação dessa classe de algoritmos exige um cuidado especial com imprecisão numérica, que em certos casos pode excluir toda uma região da malha do processo de ordenação.

Já em seu trabalho de 1998, Williams adota uma estratégia mais geral, fazendo uma revisão de um algoritmo proposto em [Stein94], corrigindo-o e tornando-o mais eficiente. Três requisitos dos seus antecessores ([Max90] e [Williams92c]) são relaxados: a malha pode ser constituída de porções desconexas; pode ser não convexa; e células vizinhas podem compartilhar parcialmente uma face. Isso dá flexibilidade para certos tipos de aplicações, por exemplo envolvendo modelos sólidos com partes móveis. Naturalmente o algoritmo resultante é computacionalmente mais pesado, apresentando complexidade de pior caso $O(n^2)$ em tempo.

O trabalho de Cignoni apresenta um algoritmo especialmente projetado para ordenação de um tipo particular de malhas de tetraedros, as triangulações de Delaunay (ver próxima seção), baseado na classificação das potências⁵ do ponto de observação em relação às esferas circunscritas aos tetraedros. Embora tenha complexidade de tempo $O(n \log n)$, devido à ordenação de uma lista linear de números reais, sendo portanto teoricamente menos eficiente do que os algoritmos baseados em ordenação topológica, resultados experimentais colocam esse algoritmo em vantagem para o tratamento de

⁴ Os trabalhos citados tratam de ordenação por visibilidade em relação a um ponto de observação. A adaptação para ordenação segundo uma direção de observação é trivial.

⁵ A potência de um ponto p em relação a uma esfera $E \subset E^d$, é definida como a diferença entre o quadrado da distância de p ao centro de E e o quadrado do raio de E . Esse valor é menor ou maior do que zero se p está dentro ou fora da esfera, respectivamente. Nesse último caso, a potência é igual ao quadrado da distância de p ao ponto de tangência a E de uma reta (qualquer) que passe por p .

conjuntos de dados de pequeno e médio porte (da ordem de algumas centenas de milhares de células). Assim como o primeiro algoritmo, o espaço de armazenamento exigido por este último também é $O(n)$. Mais precisamente, o algoritmo de Cignoni necessita armazenar apenas 4 números reais (o centro e o raio de uma esfera) para cada célula, sendo o mais econômico deles. Sua principal limitação é a incapacidade de resolver casos de degeneração da triangulação de Delaunay, onde tetraedros compartilham a mesma esfera circunscrita. Outras características que merecem destaque são: simplicidade de implementação e estabilidade numérica.

No presente trabalho será adotada como referência para avaliação comparativa aquela primeira abordagem, por ser mais eficiente, por apresentar uma vasta gama de aplicações práticas e porque suas limitações, citadas na comparação com a segunda, podem ser contornadas em parte. Uma malha não convexa pode ser transformada em convexa pela inclusão de novas células. Se todas as cavidades ou vazios da malha forem convexos, eles próprios podem ser tratados como células assinaladas como imaginárias, tornando a malha convexa. Se existirem cavidades ou vazios não convexos, eles podem ser subdivididos em tetraedros (ou outros poliedros convexos), que também serão tratados como células imaginárias. Essas células acrescentadas não são incluídas na saída do processo de ordenação. Quanto à restrição referente ao tratamento de malhas desconexas, ela pode ser superada de forma similar, incluindo células-pontes também tratadas como imaginárias. Já para a última limitação relacionada, a exigência de que a vizinhança entre células seja sempre definida por faces inteiras, não há na literatura sobre o assunto nenhuma proposta capaz de contorná-la com algoritmo linear. Portanto, essa restrição permanece para aplicação dessa estratégia.

Ciclos em Malhas de Poliedros

Nem sempre é possível obter uma ordenação direcional para uma malha de poliedros. Mesmo entre poliedros convexos pode ocorrer ciclo obstrutivo, ou seja, pode existir um conjunto de células C^1, C^2, \dots, C^k tal que $C^1 \succ C^2 \succ \dots \succ C^k \succ C^1$, como mostra a figura 5.6 para um conjunto de tetraedros.

Esta situação cria um impasse para o qual a única solução possível é a subdivisão de pelo menos uma célula, de forma a quebrar o ciclo. Essa medida, no entanto, apresenta vários inconvenientes, entre os quais: a possibilidade de se ter que subdividir um grande número de células, inclusive não pertencentes a ciclos; geração de células com formatos inesperados, por exemplo a subdivisão de tetraedros em uma malha simplicial pode resultar em células não tetraédricas; geração de novos vértices, com a necessidade de estabelecer valores do campo para eles.

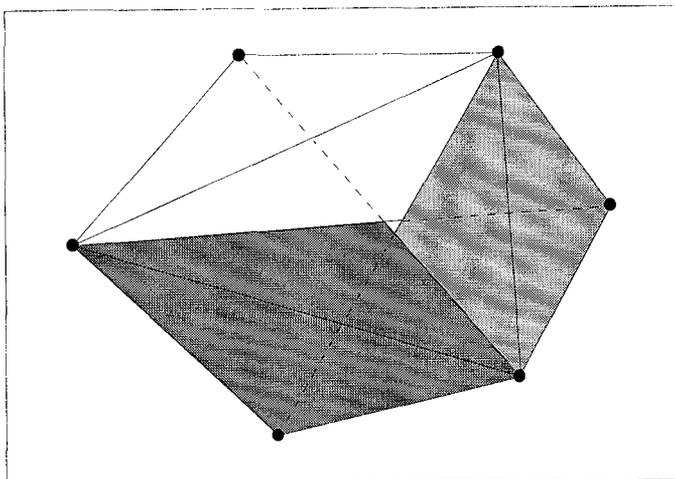


Figura 5.6 – Ciclo de tetraedros obstrutivos

Em uma malha acíclica em relação a uma direção d , a relação \succ induzida por d (seção 5.4.4) define uma ordem parcial para suas células, ficando garantida a ordenabilidade segundo aquela direção. Para simplificar a linguagem, será considerada uma *malha acíclica*, daqui para a frente, aquela que for acíclica em relação a qualquer direção. Temos notícia de iniciativa em busca de uma estratégia para verificar essa condição, mas o resultado foi um algoritmo muito custoso, mesmo para ser utilizado como pré-processamento. Não encontramos nenhuma referência a artigo publicado sobre o assunto.

Um resultado muito útil no sentido de evitar ciclos é proporcionado pelo trabalho de H. Edelsbrunner [Edelsbrunner89] que apresenta uma classe de malhas garantidamente acíclicas: são os complexos celulares do espaço euclidiano d -dimensional cujas faces são obtidas pela projeção ortogonal das faces de um polítopo convexo do espaço $(d+1)$ -dimensional. Entre essas estruturas, as mais conhecidas são as *triangulações de Delaunay* [Bowyer81] [Watson81]. Uma triangulação de Delaunay de um conjunto S de pontos do espaço euclidiano tridimensional é uma triangulação tal que a esfera circunscrita aos vértices de qualquer tetraedro da estrutura não contém nenhum ponto de S em seu interior. Esta forma de subdivisão do domínio de um problema tem sido amplamente utilizada em aplicações em computação científica. Para uma distribuição homogênea dos pontos de S , considerando ser n o tamanho desse conjunto, a triangulação de Delaunay tem tamanho $O(n)$ e o tempo esperado para sua construção é $O(n^{4/3})$ [Joe91]. No entanto, em distribuições não homogêneas a estrutura pode alcançar tamanho e tempo de construção $O(n^2)$. Apesar disso, mesmo sendo as malhas adaptativas essencialmente não homogêneas, o critério de Delaunay tem sido amplamente adotado em esquemas de geração dessas malhas. [George91] e [Weatherill94], por exemplo, são trabalhos voltados para a geração de malhas para elementos finitos. No entanto, a aplicação mais “natural” dessa estrutura talvez seja no tratamento de dados esparsos (“scattered data”), onde os pontos amostrais não têm um relacionamento topológico pré-estabelecido. Constrói-se, então, uma triangulação de Delaunay utilizando esses pontos como vértices, estrutura esta que passa a ser considerada como discretização do domínio do problema para diversas técnicas de simulação e/ou análise.

Uma estratégia que pode ser útil em determinadas aplicações, para garantir a inexistência de ciclos em uma malha, é refazer a subdivisão do domínio com a triangulação de Delaunay dos vértices da malha original. No entanto, se essa malha original é não convexa, sua fronteira não coincidirá com a da nova subdivisão, uma vez que a fronteira de uma triangulação de Delaunay é o fecho convexo de seus vértices. Para garantir a presença da fronteira original entre as faces dos tetraedros da triangulação de Delaunay, usa-se acrescentar pontos localizados sobre as faces faltantes da fronteira original e inseri-los na triangulação, fazendo as necessárias modificações para que essa estrutura continue atendendo ao critério geométrico que a caracteriza. Isso é feito até que nenhum tetraedro corte qualquer das faces da fronteira original. Todos os tetraedros, resultantes desse processo, localizados fora da fronteira da malha original, são marcados como imaginários. Não existe garantia de convergência para esse processo, mas ele tem sido aplicado com sucesso em muitos casos [Meshkat91] (citado em [Williams92a]).

Outra abordagem possível, bastante explorada para domínios não convexos, é a construção de triangulações de Delaunay com restrições. Nessas estruturas, a inclusão das faces da fronteira original é viabilizada pelo relaxamento do critério de Delaunay (critério das esferas vazias) em torno dessas faces. O resultado não é rigorosamente uma triangulação de Delaunay, podendo haver ciclos obstrutivos. Pode-se esperar que a verificação da existência ou não de ciclos nesse tipo de estrutura seja menos custosa do que em malhas genéricas, ainda que sejam triangulações.

Será mostrado mais adiante neste texto que as triangulações retangulares adaptativas são acíclicas, ainda que não atendam ao critério de Delaunay.

Capítulo 6

O Método de Projeção para Triangulações RA

Conforme mencionado anteriormente, a geração de imagens a partir de dados volumétricos estruturados em uma grade regular é muito facilitada, por terem todas as células o mesmo formato e por estarem organizadas segundo um arranjo espacial regular. No método de projeção para obtenção de imagem por projeção paralela, todos os cálculos intermediários que dependem apenas da geometria da projeção das células podem ser evitados, uma vez que essas projeções têm exatamente a mesma geometria [Wilhelms91]. Assim, pode-se reduzir drasticamente o custo do processo como um todo. Além disso, grades regulares são facilmente ordenadas segundo a direção de projeção.

Também foi mencionado que triangulações genéricas exigem um esforço computacional maior. Isso ocorre justamente porque normalmente cada célula tem um formato particular, exigindo a realização de cálculos geométricos específicos. Além disso, nem toda triangulação tridimensional admite ordenação direcional, podendo ser necessário subdividir tetraedros para superar essa dificuldade, uma vez que essa ordenação é um requisito essencial do método de projeção.

As triangulações RA ocupam uma posição intermediária entre essas duas formas de organização de dados. Como foi visto no capítulo 3, essas triangulações contêm tetraedros com no máximo 72 orientações, número insignificante se comparado com o número de tetraedros que costumam constituir os volumes nas aplicações reais, freqüentemente da ordem de centenas de milhares até milhões. Isso torna viável a aplicação de um método semelhante ao descrito em [Wilhelms91] para malhas regulares. O que é acrescentado aqui, além da construção de algumas tabelas com informações relativas às 72 configurações de projeções, é a necessidade de um ajuste de escala para cada célula, por conta dos diferentes tamanhos que células semelhantes podem ter em uma estrutura adaptativa.

De um modo geral, a projeção de um tetraedro de uma triangulação RA depende de sua posição, de seu nível (que se reflete no tamanho) e sua orientação. A orientação afeta não apenas a geometria da projeção, mas também tem reflexos nos cálculos envolvidos na avaliação dos parâmetros de cromaticidade e opacidade, como será mostrado.

Vale enfatizar que as considerações aqui desenvolvidas referem-se a projeção paralela, uma vez que no caso de perspectiva a geometria das projeções depende da

posição relativa da célula e do centro de projeção, o que inviabiliza a construção de tabelas.

Serão descritos a seguir, na seção 6.1, os elementos constituintes da estrutura de dados para o algoritmo de projeção para triangulações RA. A seção 6.2 será destinada a analisar cada passo do algoritmo descrito na seção 5.4.1 (para triangulações genéricas), do ponto de vista das implicações de sua implementação sobre triangulações RA.

6.1 Estrutura de Dados

A estrutura de dados é constituída de quatro partes: a primeira delas, o catálogo de orientações na descrição que se segue, é uma tabela fixa, independente dos dados particulares da aplicação ou dos parâmetros de visualização. A segunda são os dados do volume propriamente dito, sendo portanto característica particular de cada aplicação, mas independente dos parâmetros de visualização. As duas últimas são novamente independentes dos dados do volume, mas dependem dos parâmetros de visualização, necessitando ser recalculadas para cada nova direção de projeção.

6.1.1 Catálogo de Orientações

Este catálogo descreve características geométricas e topológicas de cada orientação de tetraedro de triangulações retangulares. A seção 3.7 apresenta um estudo detalhado de aspectos como a constituição e a implementação do catálogo através de uma tabela de tetraedros padrão associados às diferentes orientações.

6.1.2 Dados do Volume

O volume é representado por uma triangulação RA, incluindo sua estrutura topológica e a lista de seus vértices. A triangulação pode ser armazenada como um arranjo de árvores BSP (ver capítulo 3), que têm como raízes tetraedros do tipo A constituintes da subdivisão de nível mais alto do domínio. Os nós internos dessas árvores (setores) contêm apenas apontadores para seus dois filhos. Cada folha é um tetraedro da triangulação RA e armazena sua orientação, seu nível de refinamento e as referências a seus quatro vértices e a seus vizinhos.

A lista de vértices contém, em cada entrada, as coordenadas e o valor do campo escalar no ponto correspondente.

6.1.3 Guia para Percurso Ordenado

Consiste de uma tabela auxiliar para o processo de ordenação das células da triangulação por visibilidade (ver seção 6.3). Construída a partir do catálogo de orientações, é válida para uma particular direção de projeção. Para cada orientação

possível, esta tabela fornece ao algoritmo de percurso ordenado a indicação do filho (de um tetraedro com a mesma orientação) que deve ser processado primeiro. Para compor a tabela, consulta-se o catálogo de orientações para obter as orientações dos filhos. Aplica-se, então, um teste geométrico para decidir qual deles tem sua visibilidade parcial ou totalmente obstruída pelo irmão, em relação à direção de projeção.

6.1.4 Catálogo de Projeções

Trata-se de uma tabela que contém, para cada tetraedro padrão do catálogo de orientações, a indicação da classe (perfil) da projeção gerada segundo a direção considerada (ver figura 5.2), o que define a configuração da subdivisão em triângulos dessa projeção. Também é necessário registrar, para cada vértice dessa subdivisão, qual é o vértice correspondente no tetraedro e, para o vértice de espessura não nula, a espessura correspondente e as coordenadas baricêntricas dos pontos da fronteira do tetraedro que se projetam sobre ele.

Estão relacionadas, a seguir, de forma explícita para cada configuração de projeção, as informações relevantes a serem tabeladas. A notação utilizada é a mesma estabelecida no capítulo 5, seção 5.4.1.

- . Caso 1: . identificação dos vértices A, B, C e D
 - . coordenadas baricêntricas de D (vértice ENN) em relação ao triângulo ABC
 - . tamanho do segmento P1-P2
- . Caso 2: . identificação dos vértices A, B, C e D
 - . coordenadas baricêntricas do vértice ENN em relação à aresta AC e à aresta BD
 - . tamanho do segmento P1-P2
- . Caso 3: . identificação dos vértices A, B, C e D
 - . coordenadas baricêntricas de D (vértice ENN) em relação à aresta AC
 - . tamanho do segmento P1-P2
- . Caso 4: . identificação do vértice A=D (vértice ENN)
 - . tamanho do segmento P1-P2

A identificação dos vértices consiste em estabelecer a correspondência entre a notação adotada nos esquemas representativos das projeções dos tetraedros (A, B, C e D), e a numeração usada para referenciar os vértices de um tetraedro na estrutura de triangulação. Isso se faz necessário porque duas instâncias de um dado tipo de tetraedro podem ser classificadas em um mesmo *caso*, porém, apresentando um posicionamento relativo diferente entre os vértices de suas projeções, devido a diferenças de orientação. Essa identificação será responsável por garantir o correto relacionamento entre os vértices dos tetraedros e as coordenadas baricêntricas tabeladas.

Para construir esse catálogo aplicam-se, para cada tetraedro padrão, os procedimentos do método de projeção descritos no capítulo 5. Também aqui pode-se aproveitar a equivalência entre orientações indexadas (seção 3.7.3). Como os pares de orientações equivalentes possuem a mesma geometria, pode-se evitar a duplicação de cálculos aritméticos.

Também compõem este catálogo as projeções dos vértices da triangulação.

6.2 Algoritmo de Projeção de Tetraedros

Nesta seção serão consideradas as questões relativas à modificação do algoritmo de projeção de tetraedros para operar sobre triangulações RA, explorando as características dessas estruturas com o objetivo de obter uma implementação eficiente do método. Cada passo do algoritmo apresentado na seção 5.4.1 será tratado separadamente, visando ressaltar as diferenças relevantes e a simplificação dos cálculos proporcionada pelas tabelas, além de facilitar uma análise comparativa do custo computacional das duas implementações. O catálogo de orientações é considerado pré-existente, uma vez que sua construção independe da direção de projeção.

Passo zero : Construção de tabelas

Esta etapa constitui um pré-processamento relativo a cada nova direção de projeção, onde se inclui a construção do guia para percurso ordenado e o catálogo de projeções, que será completado no próximo passo.

Passo um : Projeção dos vértices da triangulação

A operação de projeção dos vértices de uma triangulação RA é essencialmente a mesma para uma triangulação genérica, exigindo um esforço computacional equivalente.

Passo dois : Ordenação das células

A ordenação direcional das células de uma triangulação RA pode ser realizada sem nenhum teste geométrico, com o auxílio do Guia para Percurso Ordenado. A discussão desse processo, com apresentação de um algoritmo, será deixada para a seção 6.3, para uma abordagem mais detalhada.

Passo três : Classificação dos tetraedros

Para cada tetraedro da estrutura a ser projetado, a classificação consistirá apenas de uma consulta ao catálogo de projeções.

Passo quatro : Decomposição das projeções dos tetraedros

Este passo do algoritmo fica isento das operações aritméticas relativas ao cálculo de interseção de arestas, requerido no caso número dois dos perfis de projeção, para determinar o vértice ENN. Também não é necessário calcular as coordenadas baricêntricas do vértice ENN, a serem usadas como coeficientes de interpolação, porque já estão tabeladas. Finalmente, também é desnecessário calcular as coordenadas Z correspondentes aos pontos P1 e P2, pontos das faces da célula que se projetam sobre o vértice ENN. Estas últimas coordenadas seriam usadas para determinar, pela diferença entre elas, a distância entre P1 e P2, que é a espessura da célula correspondente ao vértice ENN. Esta espessura também já está disponível na tabela, dependendo apenas da aplicação de um fator de escala, calculado em função do nível de refinamento do

particular tetraedro. A isenção do cálculo daquelas coordenadas Z evita algumas interpolações sobre faces e/ou arestas do tetraedro. Calculada a espessura, restam apenas as interpolações dos valores da função no ponto P2, quase sempre necessária, e, eventualmente, no ponto P1. Os coeficientes para estas interpolações são as coordenadas baricêntricas do vértice ENN em relação às projeções da face ou da(s) aresta(s) relevante(s), que também são recuperadas do catálogo de projeções.

Estas simplificações ressaltam da comparação entre os algoritmos 6.1 e 5.1.

para cada tetraedro
sejam P1 e P2 os pontos do tetraedro delimitantes do segmento
correspondente ao vértice ENN
recuperar as coordenadas baricêntricas do vértice ENN
se caso 1 então
{P1 é o vértice D do tetraedro e P2 pertence à face oposta}
calcular f(P2) por interpolação linear na face
senão se caso 2 então
{o vértice ENN é a interseção dos segmentos AC e BD na
figura plana}
calcular f(P1) e f(P2) por interpolação linear nas arestas
AC e BD do tetraedro
senão se caso 3 então
{P1 é o vértice D do tetraedro e P2 pertence à aresta AC}
calcular f(P2) por interpolação linear na aresta
senão
{caso 4 → P1 e P2 são os vértices com projeções
coincidentes}

recuperar a espessura referente ao vértice ENN {d_{P1-P2(tab)}}
calcular a distância entre P1 e P2
{d_{P1-P2} = d_{P1-P2(tab)} / 2^{niv_ref div 3}, para niv_ref começando em zero}

Algoritmo 6.1

Passo cinco : Cálculo de cor e opacidade nos vértices dos triângulos

Também esta etapa envolve os mesmos cálculos, seja para uma triangulação RA ou para uma triangulação genérica.

Passo seis : Rasterização dos triângulos

Etapa realizada pelo hardware, novamente sem diferença de desempenho computacional entre os diferentes tipos de triangulações.

6.3 Ordenação das Células

Reafirmando o conceito expresso no capítulo 5, um percurso ordenado (ou ordenação direcional) das células de uma triangulação (ou de um complexo celular qualquer) em relação a uma direção d , é um percurso tal que cada célula é visitada antes de qualquer outra que obstrua sua visibilidade naquela direção.

Também já foi mostrado naquele capítulo que malhas de poliedros, ainda que convexas e mesmo malhas simpliciais, nem sempre admitem ordenação direcional, podendo ocorrer ciclos obstrutivos. Em contraposição, foi citado um trabalho [Edelsbrunner89] que demonstra que as triangulações de Delaunay são acíclicas, ou seja, sempre podem ser ordenadas em qualquer direção. Na seqüência, demonstra-se que as triangulações RA também têm essa propriedade e apresenta-se um algoritmo que fornece um percurso ordenado para uma dada direção.

Definição

Um complexo celular T , composto por células convexas, permite *remoção livre* de uma de suas células constituintes, segundo uma dada direção, se nenhuma outra célula de T obstrui essa operação.

Teorema

Sejam T_1 e T_2 duas triangulações tridimensionais. Se T_1 é acíclica e T_2 é derivada de T_1 pela bisseção de uma de suas células por um corte plano, então T_2 também é uma triangulação acíclica.

Demonstração

Seja d uma direção qualquer e t uma célula qualquer de T_1 . Sendo T_1 uma triangulação acíclica, é possível remover todas as células de T_1 que obstruam parcial ou totalmente a visibilidade de t segundo a direção d , apenas por remoções livres. Seja T_t a triangulação resultante desse processo. Portanto, T_t permite a remoção livre de t , segundo aquela direção.

Seja P um plano que divide t em dois poliedros. Como esses poliedros são convexas, sempre existirá uma ordem entre eles em que ambos possam ser removidos livremente do complexo celular resultante daquele corte, segundo a direção d , qualquer que seja o plano P . Isso caracteriza a inexistência de ciclo obstrutivo envolvendo esses poliedros e as células remanescentes de T_t . Por outro lado, a subdivisão de t não afeta a mobilidade das células retiradas para transformação de T_1 em T_t , o que garante também que não existe ciclo obstrutivo entre os novos poliedros e aquelas células removidas.

Então a subdivisão de t não cria ciclo obstrutivo em uma estrutura originalmente acíclica. Como T_2 é derivada de T_1 exatamente por um corte deste gênero, conclui-se que T_2 também é uma estrutura acíclica. ♦

Corolário

As triangulações RA são acíclicas.

Demonstração

O termo triangulação retangular tem sido usado neste trabalho para designar uma subdivisão simplicial de um domínio constituído de um arranjo regular de cubos. Inicialmente, essa subdivisão é uma triangulação regular em que cada cubo está decomposto em 6 tetraedros do tipo A. Todos os 6 tetraedros em que se subdivide um cubo têm a mesma esfera circunscrita, qual seja a esfera cujo centro coincide com o centro do cubo e cujo raio mede $l\sqrt{3}/2$, onde l é o comprimento da aresta do cubo. É fácil verificar que qualquer outro vértice dessa estrutura, excluindo os vértices do próprio cubo, dista pelo menos $l\sqrt{11}/2$ do centro dessa esfera, ou seja, não existe nenhum vértice da estrutura dentro da esfera. Conclui-se, então, que essa subdivisão simplicial é uma triangulação de Delaunay, sendo portanto acíclica.

A geração de uma triangulação adaptativa, a partir dessa configuração regular, se dá por uma sucessão de operações de refinamento. Esse refinamento consiste da subdivisão de um núcleo, que por sua vez constitui-se de uma série de aplicações do procedimento bissecção-ao-meio (cortes planos) aos tetraedros desse núcleo. Portanto, o teorema permite concluir que as triangulações RA são acíclicas, e que elas possuem sempre ordenação direcional, qualquer que seja a direção de observação. ♦

6.3.1 Algoritmo de ordenação

A organização hierárquica das triangulações RA e a regularidade da geometria da descendência nas árvores que as constituem, proporcionam uma excelente solução para o problema de ordenação das células por visibilidade.

Em linhas gerais, a ordenação de uma triangulação RA em uma direção d , deve começar pelo percurso ordenado de sua triangulação raiz. Para cada célula dessa triangulação que tiver descendentes, promove-se o percurso ordenado de sua árvore.

Para percorrer a triangulação raiz, considera-se inicialmente a subdivisão do domínio em cubos, definindo a ordem em que eles devem ser visitados. O processamento de cada cubo consiste de testes geométricos para estabelecer a ordem de visita de seus tetraedros, raízes das árvores BSP. Por sua vez, o processamento de cada um desses tetraedros consiste de uma chamada ao procedimento de percurso de sua árvore.

Para percorrer uma árvore BSP de uma triangulação RA, novamente aplicam-se testes geométricos, para determinar a relação de ordem entre os dois filhos da raiz, e invoca-se, recursivamente, o procedimento de percurso para suas respectivas sub-árvores, começando pela do filho que ganhou prioridade na ordenação.

Os testes geométricos referidos acima não precisam ser realizados para cada cubo ou nó de árvore visitado. O teste relativo a nó de árvore é substituído por uma consulta ao Guia para Percurso Ordenado, construído para cada direção de projeção. Os testes para ordenação dos tetraedros constituintes de um cubo, também podem ser substituídos por consulta a uma tabela, constituída de oito entradas referentes às duas diferentes indexações de cada uma das quatro posições possíveis para a subdivisão do cubo, ou

seja, referente a duas indexações para cada diagonal do cubo. Esta última tabela também precisa ser montada em função da direção de projeção.

Implementação do Algoritmo de Ordenação

Seja uma malha de cubos composta por N_i cubos na direção do eixo X, N_j cubos na direção de Y e N_k cubos na direção de Z. Seja ($Orig_i$, $Orig_j$, $Orig_k$) a *origem da malha* (ponto origem) e o *cubo origem* aquele que possui índices $i=0$, $j=0$, $k=0$.

Cada cubo da malha pode ser obtido por uma composição de reflexões do cubo origem: o cubo de índice (i,j,k) resulta de uma reflexão em relação a um plano paralelo ao plano yz, se i é ímpar, composta com uma reflexão em relação a um plano paralelo ao plano xz, se j é ímpar, e finalmente em relação a um plano paralelo ao plano xy, se k é ímpar.

Eventualmente, recorre-se a um sistema de coordenadas local de um cubo. Trata-se de um sistema cartesiano cuja origem, *origem do cubo*, é o vértice do cubo que corresponde à imagem da origem da malha, pelas reflexões pertinentes. Naturalmente, a origem do cubo origem coincide com a origem da malha. As direções (orientadas) de referência desse sistema local são as reflexões das direções de referência globais (eixos X, Y e Z), com as respectivas escalas adaptadas de forma a que as coordenadas da origem do cubo sejam $(0, 0, 0)$ e todas as dimensões do cubo sejam unitárias.

Como os cubos são subdivididos em seis tetraedros (setores), faz-se necessário estabelecer uma forma de indexação destes. É conveniente adotar os mesmos índices de permutação P, descritos no capítulo 3. Aquela enumeração serve para indexar o tetraedro no array correspondente ao cubo. O índice de sinais de um setor de um cubo depende apenas dos índices desse cubo. Assim, todos os setores do cubo de índice (i,j,k) possuem os sinais

$(+,.,.)$ se i é par e $(-.,.)$ caso contrário;
 $(.,+.)$ se j é par e $(.,-.)$ caso contrário;
 $(.,.,+)$ se k é par e $(.,.,-)$ caso contrário.

Quando necessário, os vértices dos cubos são enumerados conforme a tabela 6.1, onde suas coordenadas são expressas localmente:

índice do vértice	coordenadas
0	(0,0,0)
1	(1,0,0)
2	(0,1,0)
3	(1,1,0)
4	(0,0,1)
5	(1,0,1)
6	(0,1,1)
7	(1,1,1)

Tabela 6.1 – Vértices de cubo em coordenadas locais

O algoritmo de ordenação começa estabelecendo uma ordem de percurso para os cubos, de acordo com a direção. Ao percorrer um cubo, estabelece a ordem de percurso dos setores-raízes (tetraedros raízes de árvores BSP) desse cubo, também respeitando a direção dada. Finalmente, percorre ordenadamente a árvore associada a cada um desses setores.

Para iniciar o percurso da malha de cubos, determina-se qual dos oito vértices que a delimitam é o primeiro a ser encontrado na direção proposta. Identificado o cubo a que pertence esse vértice, ele é o primeiro a ser tratado, sendo os demais processados por camadas (com índice i constante) que são percorridas por colunas (com índice j constante).

A inspeção das configurações possíveis de setores em cubos, revela que o percurso dos setores-raízes de um cubo pode ser realizado numa seqüência predeterminada da forma:

$$jj, jj+1, jj+2, jj-1, jj-2, jj-3,$$

a partir de um determinado setor inicial de índice jj .

Em outras palavras, identificado o setor inicial (o mais distante ou o mais próximo dos 6 tetraedros, dependendo da orientação do percurso), visitá-lo; depois visitar dois tetraedros em seqüência para qualquer lado; finalmente, visitar a seqüência de 3 tetraedros pelo outro lado.

O setor inicial, no entanto, varia com a orientação do cubo considerado, entendendo-se como orientação de um cubo as direções dos eixos do seu sistema local de referência. No total, os cubos apresentam-se em 8 orientações distintas. Assim sendo, é útil construir uma pequena tabela que indique o índice do setor inicial de percurso, segundo a particular direção considerada, para cada uma das 8 orientações de cubos. Descreve-se, a seguir, uma maneira simples de preencher cada entrada desta tabela, válida se aplicada após a transformação de visualização, quando o eixo Z já é paralelo à direção em questão. O processo começa pela escolha de um setor inicial qualquer de um cubo com aquela orientação. Este setor é então comparado com um de seus vizinhos no cubo: aquele cujo vértice que não pertence à face comum tem coordenada Z maior (ou menor, dependendo da orientação do percurso) é escolhido. Este será comparado da mesma forma com seu outro vizinho no cubo. Em caso de empate, que ocorre quando a face de contato é paralela à direção considerada, deve-se manter o tetraedro atual. A identificação dos vértices que não pertencem à face comum fica facilitada pelas seguintes observações: são os vértices de índice 1 ou 2, e esses índices são iguais em ambos os tetraedros. Esta abordagem exige no máximo 3 comparações entre pares de setores para identificar o setor inicial.

Estabelecida a ordem de visitação dos setores-raízes de um cubo, o percurso ordenado da árvore associada a cada um deles requer a determinação da orientação de seu tetraedro. Relembrando o capítulo 3, na sistemática ali estabelecida para identificação das orientações de tetraedros, o índice de uma orientação é dado pela expressão:

$$48t + 8P + S$$

onde t é o tipo do tetraedro ($A=0, B=1, C=2$), e P e S são seus índices de permutação e de sinais (tabelas 3.4 e 3.5, respectivamente). Por se tratarem de setores-raízes, são

todos tetraedros do tipo A, o que reduz a expressão a suas duas últimas parcelas. O índice de permutação coincide com o índice do setor no cubo e o de sinais é calculado para o primeiro cubo – note-se que todos os tetraedros de um mesmo cubo possuem o mesmo índice de sinais – e transmitido para os demais por reflexão.

O outro requisito para permitir um percurso eficiente da árvore de um setor é a construção do Guia para Percurso Ordenado (seção 6.1.3). Consiste de uma tabela que indica, para cada orientação de tetraedro, qual o filho que deve ser visitado primeiro, de acordo com a direção de projeção. Para sua construção, consulta-se o Catálogo de Orientações para obter um tetraedro padrão para cada orientação. O teste geométrico consiste da comparação da direção de projeção com o vetor que vai do vértice 0 ao vértice 3 do tetraedro. Se o produto escalar desses dois vetores for positivo, indicando que o vértice 0 está na frente do vértice 3, o filho 1, aquele que herda o vértice 3 do pai, será visitado primeiro. Caso contrário, o filho 2, herdeiro do vértice 0 do pai, terá a preferência.

6.4 Avaliação

Para fins de avaliação do ganho de eficiência do algoritmo de projeção de tetraedros implementado para triangulações RA, em relação a sua implementação para triangulações genéricas, é conveniente distribuir os passos dos algoritmos das seções 5.4.1 e 6.2 em três grupos: os passos 0, 1, 3, 4 e 5 serão tratados em conjunto como fase de projeção (embora não sejam consecutivos); o passo 2 como fase de ordenação; e o passo 6 como fase de rasterização. Essa separação é motivada por serem processos de naturezas distintas, com peculiaridades que tornam nebulosa a análise em conjunto. Existem algoritmos para essas três fases com a mesma ordem de complexidade computacional, $O(n)$ em tempo. No entanto, a parcela de tempo que cada uma dessas etapas ocupa em relação ao total do processo pode variar muito, de acordo com as características da plataforma computacional, devido às diferenças de natureza das operações predominantes em cada uma delas.

Assumindo que a rasterização seja feita pelo hardware, o estrangulamento do processo estará nas duas primeiras fases, realizadas por software, que são justamente as etapas onde se concentram os ganhos com a adoção das triangulações RA. Comparando o tempo de execução dessas duas fases, conforme relata [Cignoni95], a projeção é uma operação muito mais custosa do que a ordenação. Os resultados quantitativos para os dois conjuntos de dados estudados naquele trabalho, fornecem uma relação em torno de 1:10 entre os tempos para ordenação e para projeção, respectivamente.

Com base nesses dados foi adotada, como medida simplificadora dos experimentos realizados no âmbito deste trabalho, a opção de concentrar a atenção na fase de projeção. Foram feitas implementações para triangulações genéricas e para triangulações RA. Os resultados obtidos são avaliados comparativamente.

Quanto à ordenação, apresentamos a seguir uma análise teórica e considerações sobre os resultados práticos esperados, em comparação com outros métodos já citados.

6.4.1 Análise do Procedimento de Ordenação

O algoritmo de ordenação para triangulações RA tem complexidade $O(n)$ em tempo, uma vez que se trata de um simples percurso de uma árvore binária. Portanto, do ponto de vista teórico, ele é equivalente ao algoritmo baseado em ordenação topológica, apresentado no capítulo 5. No entanto, a inexistência de testes geométricos cria a expectativa de um desempenho altamente vantajoso. Mesmo na faixa de estruturas de pequeno e médio porte, onde a ordenação numérica (ordenação de potências) se mostrou, nos resultados práticos, mais eficiente do que a topológica, apesar da desvantagem teórica (capítulo 5), esse algoritmo específico para triangulações RA deverá ganhar a preferência. Esta previsão se justifica porque, além de ser um algoritmo ótimo do ponto de vista teórico, o processamento de cada nó interno da árvore, no percurso ordenado, envolve apenas a verificação de um campo lógico que indica qual dos dois filhos do setor tem prioridade.

Vale ressaltar que a abordagem para ordenação de triangulações RA terá seguramente boas características do ponto de vista de estabilidade numérica. Esta afirmação se baseia nos seguintes fatos: inexistência de células degeneradas e; os testes geométricos são limitados aos tetraedros padrão, podendo ser realizados com aritmética de maior precisão, sem onerar muito o processo como um todo.

Quanto aos requisitos de armazenamento, considerando que as árvores BSP já são parte integrante da estrutura de dados, esse algoritmo tem um pequeno consumo fixo, referente às tabelas, e um consumo variável associado ao processo recursivo. No pior caso, o espaço de armazenamento será linear, ficando também aqui equivalente à ordenação topológica. Isto ocorrerá em configurações com características muito particulares, correspondendo a condições de adaptatividade que induzam um desequilíbrio absoluto da árvore BSP, fazendo-a crescer sempre no mesmo ramo. O melhor caso corresponde a uma estrutura com refinamento homogêneo, portanto sem nenhum caráter adaptativo, onde o requisito de memória será $O(\log_2 n)$.

6.4.2 Resultados do Procedimento de Projeção

Os experimentos realizados estão baseados em implementações do método de projeção, feitas para triangulações genéricas e para triangulações RA. O campo escalar utilizado é a função distância a um eixo, a partir da qual foram produzidos diversos conjuntos de dados. Esses dados são amostras da função em pontos distribuídos de forma homogênea, tendo como domínio um cubo que tem o eixo de referência da função como eixo de simetria, paralelo a quatro de suas arestas. Dois desses conjuntos (D1 e D2) constituídos por 100 e 1000 pontos sorteados aleatoriamente, foram usados para construir triangulações de Delaunay, sobre as quais foi aplicado o método de projeção para triangulações genéricas. As estruturas resultantes têm 495 e 6165 tetraedros, respectivamente. Os outros conjuntos (RA1, RA2 e RA3) referem-se a triangulações RA regulares (todas as células são do mesmo tipo e tamanho), com níveis de refinamento 8, 9 e 10, constituídas por tetraedros dos tipos C, A e B, respectivamente. A primeira tem 1536 tetraedros, a segunda 3072 e a última 6144.

As figuras 6.1 e 6.2, são imagens obtidas a partir das triangulações D1 e D2, enquanto que as figuras 6.3 e 6.4 são provenientes das triangulações RA. Todas elas apresentam marcas lineares correspondentes a arestas de células, observáveis mais facilmente nas figuras 6.2 e 6.4. Essa característica é típica do método de projeção de tetraedros de Shirley e Tuchman, implementado sem utilização do mecanismo de mapeamento de textura para o cálculo de opacidade (seção 5.4.2). Deve-se ressaltar, no entanto, que as imagens obtidas por projeção de triangulações RA, segundo algumas direções particulares (seção 6.5), apresentam essas marcas mais acentuadas, como mostra a figura 6.4, cuja direção de projeção é paralela a um dos eixos coordenados. Isso se deve à superposição das projeções de arestas de diversas células, e é razoável esperar que essas imagens sejam beneficiadas de forma mais notória pela inclusão do referido mecanismo, para melhorar a precisão do cálculo de opacidade.

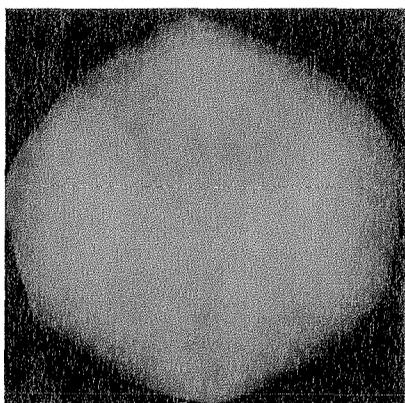


Figura 6.1 – Triangulação de Delaunay com 6165 tetraedros.

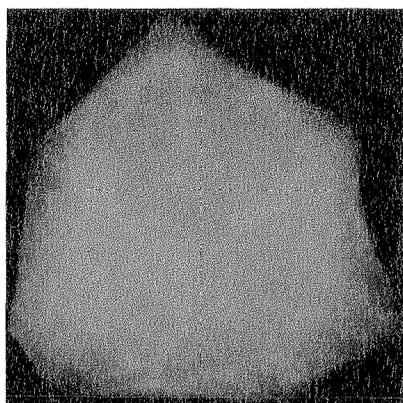


Figura 6.2 – Triangulação de Delaunay com 495 tetraedros.

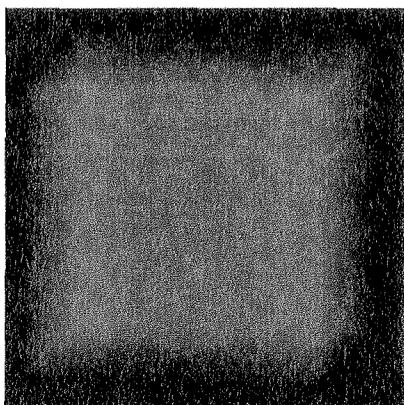


Figura 6.3 – Triangulação RA com 3072 tetraedros.

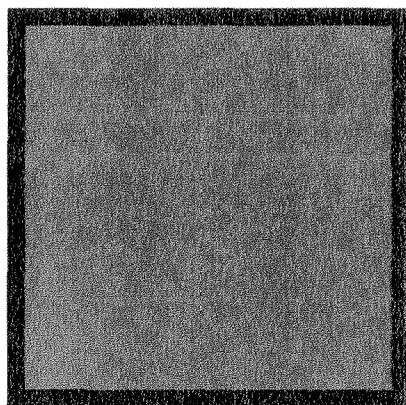


Figura 6.4 – Triangulação RA com 6144 tetraedros, observada segundo uma direção paralela a um dos eixos coordenados

A seguir apresenta-se um quadro comparativo de tempos de processamento em diferentes plataformas computacionais (tabela 6.2). Essas medidas referem-se aos cálculos relativos à execução das projeções dos tetraedros, excluindo, portanto, as fases

de ordenação, cujo algoritmo específico para triangulações RA ainda não foi implementado, e de rasterização dos triângulos. Os experimentos foram baseados nas estruturas D2, RA1, RA2 e RA3, citadas acima, sendo o tempo em segundos referente à projeção de um milhão de tetraedros. Para as triangulações RA, os resultados tabelados são as médias das três estruturas. Duas classes de experimentos (1 e 2) são identificadas, cuja diferença está na forma de cálculo da opacidade.

Não se pretende, com esses resultados, oferecer uma comparação rigorosa de desempenho dos métodos, uma vez que as implementações apresentam diferenças estruturais significativas. A triangulação RA foi implementada com alocação dinâmica de memória, sendo que o acesso a cada item de um tetraedro ou de um vértice é feito através de chamada a uma função de consulta ou de manipulação. Já a estrutura implementada para triangulações genéricas está baseada em alocação estática, com manipulações e consultas feitas diretamente sobre os vetores de vértices e de tetraedros. No entanto, essas diferenças penalizam a versão para triangulações RA, o que permite esperar por uma ampliação dos ganhos já apurados, na medida em que sejam feitas implementações mais equilibradas.

	1: $\alpha=1-e^{-\tau l}$ 2: $\alpha=\tau l$	triangulação genérica (D)	triangulação RA	RA/D
IBM risc6000	1	8,4	5,2	0,62
	2	7,1	3,9	0,55
Sun Sparc20	1	11,0	7,2	0,65
	2	9,1	5,2	0,58
Pentium II	1	4,4	2,3	0,52
	2	3,5	2,0	0,57

Tabela 6.2 – Quadro comparativo de tempos de execução do procedimento de projeção para um milhão de tetraedros (em segundos).

6.5 Direções Privilegiadas de Projeção

As projeções dos tetraedros de uma triangulação RA, segundo certas direções especiais, apresentam características de regularidade ainda mais fortes. São elas as direções dos eixos coordenados, das diagonais do cubo e das diagonais das faces do cubo. Por serem também as direções das arestas dos tetraedros, aumenta significativamente a incidência do *caso 4* da classificação de projeções, que é o caso de tratamento menos custoso, não exigindo cálculo de interseções ou interpolações (ver capítulo 5). A segunda classificação mais frequente passa a ser o *caso 3*. O *caso 2*, que é o de maior incidência em direções genéricas, e o *caso 1*, os dois mais custosos, são, ainda em ordem, os menos frequentes nessas direções particulares. Como será mostrado, mesmo essas últimas configurações terão seus cálculos simplificados.

Essas direções privilegiadas são em número de 26. Nesta seção, serão analisadas as particularidades das configurações de projeção produzidas por essas direções e como se pode tirar proveito delas para aumentar a eficiência do processo. No quadro a seguir (figura 6.5) essas configurações são catalogadas para serem referenciadas adiante.

As configurações ilustradas na figura, com respeito aos casos da figura 5.2, correspondem a:

- . q1 caso 4
- . q2 caso 3, com interseção no centro da aresta
- . q3 caso 2, com interseção no centro das arestas
- . q4 caso 2, com interseção a 1/3 de ambas as arestas
- . q5 caso 1, com coeficientes de interpolação nas proporções 1:1:2

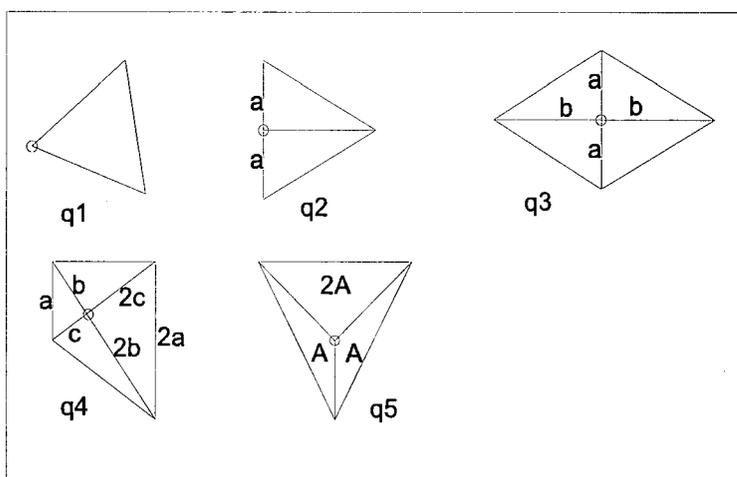


Figura 6.5 – As 5 configurações possíveis de projeções de tetraedros de uma triangulação RA em direções privilegiadas

Considerando o algoritmo de projeção de tetraedros na sua forma apresentada no capítulo 5, a importância de tomar essas direções particulares concentra-se nos passos quatro (decomposição das projeções) e cinco (cálculo de cor e opacidade nos vértices dos triângulos).

Recordando os cálculos a serem realizados nesses passos, uma primeira observação é que não será necessário computar nenhuma interseção entre arestas, uma vez que essas interseções ocorrem em posições pré-definidas. Também pelo mesmo motivo, a determinação de coordenadas baricêntricas relativas a arestas e faces é desnecessária. Isso porém é pouco significativo, uma vez que esses cálculos são feitos apenas para compor a tabela dos 72 tipos x orientações característicos das triangulações RA. O que tem maior peso, no que se refere ao número de operações de ponto flutuante realizadas para um tetraedro, são as interpolações de f , que são operações específicas de cada célula do volume. Observando o quadro acima identifica-se que, onde são necessárias, essas interpolações são feitas com as proporções 1:1 (configurações q2 e q3), 1:2 (configuração q4) e 1:1:2 (configuração q5). Uma análise acurada do algoritmo, no que tange a essas operações, mostra que é possível reorganizá-las em função dessas particulares proporções, de forma a obter alguma economia, ainda que pequena.

No entanto, os reflexos mais relevantes da adoção de direções especiais dizem respeito à proporção de ocorrência dos diferentes casos de configurações de projeções. As tabelas a seguir, incluídas para quantificar esse aspecto, foram obtidas pelo cruzamento dos 72 tetraedros padrão contra cada uma das direções especiais possíveis, contando a incidência das 5 classes de projeções.

	q1	q2	q3	q4	q5
Tipo A	144				
Tipo B	96	48			
Tipo C	96	48			
%	78	22			

Tabela 6.3 – Projeções nas 6 direções associadas aos eixos coordenados

	q1	q2	q3	q4	q5
Tipo A	48	96		48	
Tipo B	144		48		
Tipo C	96	96			
%	50	33,3	8,3	8,3	

Tabela 6.4 – Projeções nas 8 direções associadas às diagonais do cubo

	q1	q2	q3	q4	q5
Tipo A	96	96	96		
Tipo B	48	144		96	
Tipo C	96	48		96	48
%	27,8	33,3	11,1	22,2	5,5

Tabela 6.5 – Projeções nas 12 direções associadas a diagonais de faces

	q1	q2	q3	q4	q5
Ocorrências	864	576	144	240	48
%	46,1	30,8	7,7	12,8	2,6

Tabela 6.6 – Quadro global de ocorrências das cinco configurações

Essas tabelas mostram a predominância da configuração *q1*, cujo tratamento não envolve nenhuma interpolação. Em segundo lugar, mas ainda bem mais numerosa do que as demais, a configuração *q2*, cujo tratamento envolve apenas uma interpolação com razão um entre os coeficientes. As outras configurações, que envolvem cálculos mais custosos, ainda que simplificados em relação a direções genéricas, ocorrem em pequena proporção.

[Williams92a] reporta uma incidência de aproximadamente 40% de casos 1 e 60% de casos 2 em seus experimentos com triangulações genéricas. Os casos 3 e 4, tratados como degenerações dos primeiros, ocorrem muito raramente. Esses números devem permanecer válidos para as triangulações RA fora das direções explicitadas acima.

Uma forma prática de aproveitar essa peculiaridade da estrutura é implementar um modo especial de interação restrito a projeções nessas direções privilegiadas. Esse modo seria útil para uma rápida exploração do volume, permitindo ao usuário identificar regiões a serem exploradas mais minuciosamente em outro modo de interação que ofereça flexibilidade para especificar de forma contínua a direção desejada.

6.6 Projeção de Modelos Simplificados

Um modelo simplificado, obtido a partir de uma triangulação RA multiresolução, conforme os critérios estabelecidos no capítulo 4, é um modelo completo e consistente, uma vez que a estrutura sobre a qual está definido é uma triangulação RA e a função representada é uma função contínua.

Sendo assim, pode-se aplicar sobre ele qualquer técnica estabelecida sobre esse tipo de estrutura, incluindo, naturalmente, a técnica de projeção apresentada neste capítulo. Observa-se, no entanto, que a obtenção de modelos dinâmicos se dá por um processo intimamente vinculado ao algoritmo de ordenação das células de uma triangulação RA: o percurso recursivo das árvores que compõem a estrutura, característico daquele algoritmo é uma particularização do percurso utilizado para a definição de modelos dinâmicos. Isso permite que, estabelecido o valor de tolerância para o qual se quer obter um modelo simplificado, aplicando-se aquela estratégia de percurso, cada tetraedro identificado como folha do modelo possa ser imediatamente projetado.

Embora esta estratégia seja particularmente eficiente para projeção paralela, devido à aceleração que as triangulações RA proporcionam para essa modalidade de geração de imagens, sua aplicabilidade também se estende à projeção em perspectiva. O algoritmo de ordenação é perfeitamente adaptável para classificar as células em função da visibilidade em relação a um centro de projeção, ainda que perdendo a possibilidade de estabelecer o critério de classificação em função de uma tabela pré-computada. Neste caso, a decisão quanto à visibilidade de um irmão em relação ao outro exigirá cálculos geométricos específicos (seção 6.3.1).

Capítulo 7

Lançamento de Raios em Triangulações

RA

O método de lançamento de raios (“ray casting”) já foi descrito no capítulo 5, onde foi apresentada em detalhes uma abordagem específica para malhas simpliciais (triangulações). A técnica ali tratada permite uma implementação eficiente desse método para triangulações genéricas. Pode-se adaptar aquela metodologia para triangulações RA, tirando vantagem da possibilidade de substituir cálculos dispendiosos por consultas a tabelas pré-computadas [Soares97]. Como no método de projeção, os maiores ganhos com essa abordagem também seriam obtidos com projeção paralela. Uma desvantagem dessa solução é que, para alcançar eficiência, seria necessário acrescentar à estrutura de dados um grande volume de informações topológicas, relativas às células vizinhas e aos vértices de cada célula. Outra limitação é sua desconexão com os mecanismos de multiresolução apresentados, uma vez que a organização hierárquica da estrutura não é levada em conta.

A abordagem deste capítulo, ao contrário, baseia-se justamente na hierarquia das triangulações RA. A representação de informações topológicas é minimizada com a utilização da estrutura de dados descrita no capítulo 3. O resultado é um algoritmo que é, no mínimo, tão eficiente quanto o anterior, do ponto de vista do tempo de execução. Suas vantagens mais significativas são a economia de espaço e a perfeita sintonia com a estratégia de multiresolução. A técnica básica é a mesma, ou seja, para cada pixel da janela da tela onde se pretende gerar a imagem, um raio é lançado, partindo do olho do observador e passando pelo centro do pixel. Ao longo de sua trajetória, esse raio intercepta uma série de células, cujas contribuições para a cor do pixel devem ser determinadas e combinadas, levando em conta a ordem dessas interseções. O algoritmo que se apresenta a seguir é válido tanto para projeção paralela quanto para perspectiva.

7.1 Algoritmo Hierárquico

O algoritmo para tratamento de um raio começa identificando e ordenando os cubos interceptados por ele. Por sua vez, para cada cubo, determinam-se seus tetraedros interceptados (triangulação raiz), que também são ordenados. A árvore relativa a cada um desses tetraedros (setor raiz), será tratada a seu turno. Para isso, calculam-se os pontos de entrada e saída do raio no setor. Naturalmente, esse processo aproveita os cálculos relativos às faces comuns entre setores vizinhos da triangulação raiz. Esses

pontos extremos do segmento do raio no setor, serão parâmetros para um procedimento que realizará a integração da contribuição daquele setor para a cor do pixel. Se já se tratar de uma folha da árvore, a integração é efetuada imediatamente. Senão, inicia-se um processo recursivo, que determinará a interseção do raio com a face que separa os filhos do setor, se existir tal interseção, e repetirá a chamada para requisitar os cálculos relativos a seu(s) filho(s). A figura 7.1 ilustra o processo com um caso bidimensional.

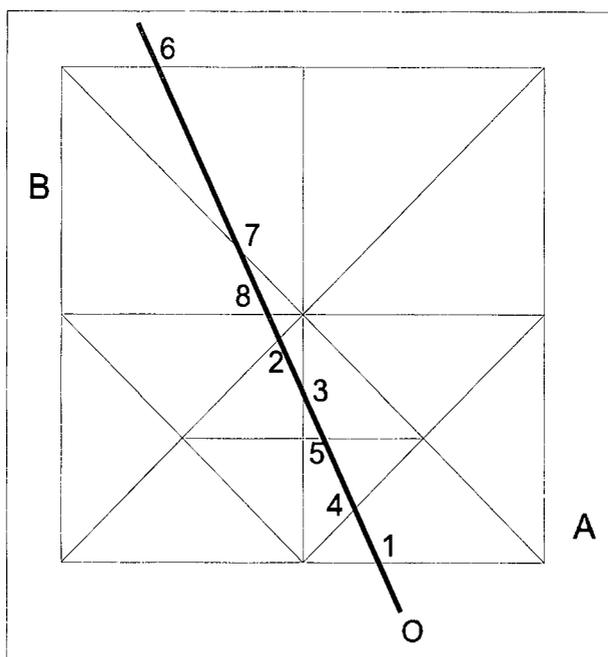


Figura 7.1 – Seqüência de interseções de um raio com os setores de uma malha, considerando que a origem do raio é o ponto “O”, e que a árvore “A” é percorrida antes da árvore “B”.

Sempre que ocorrer a interseção de dois irmãos, visita-se primeiro aquele mais próximo do observador (ou mais distante, dependendo da ordem escolhida para o algoritmo). As folhas da árvore efetivamente visitadas constituem uma seqüência de tetraedros contíguos que contêm o segmento do raio interior ao volume.

A geometria da malha de cubos e da triangulação raiz, como aliás de toda a triangulação RA, não é representada explicitamente. No entanto, a regularidade do nível superior dessa estrutura, torna essas informações facilmente dedutíveis. Além disso, é uma premissa da modelagem com triangulações RA, que a malha de cubos é pouco refinada, ficando o detalhamento por conta da adaptatividade da malha de tetraedros. Com isso, a determinação das interseções de um raio com a malha de cubos e com os tetraedros da triangulação raiz pode ser considerada pouco significativa, do ponto de vista de seus custos, apesar da inexistência de informações topológicas explícitas, na representação adotada, impor a necessidade de se testarem, eventualmente, os seis tetraedros de cada cubo interceptado.

Diante disso, o enfoque desta abordagem estará centrado no tratamento de cada árvore binária da estrutura separadamente. Os cálculos geométricos envolvidos serão explicitados a seguir, para permitir uma comparação de seus custos com aqueles associados às triangulações genéricas. Por tratar-se de subdivisão simplicial, a utilização de coordenadas baricêntricas permanece sendo a opção mais conveniente para os cálculos relativos à geometria das células das triangulações RA.

A primeira constatação importante é que o número de interseções a serem calculadas é muito inferior àquele realizado no tratamento de triangulações genéricas, descrito no capítulo 5. Considerando já conhecido o ponto de entrada de um raio em um tetraedro, a estratégia ali empregada calcula os parâmetros que definem as interseções desse raio com os planos das outras três faces desse tetraedro, para depois concluir qual delas é o ponto de saída. No método para triangulações RA, as interseções calculadas ao longo do percurso de uma árvore se referem a faces bem definidas, quais sejam, as faces que dividem setores. Conhecidos os pontos **P** e **Q**, interseções do raio com um setor *S*, a existência ou não da interseção do raio com a face que divide *S* é verificada previamente, comparando as coordenadas baricêntricas de **P** e **Q**, relativas aos vértices extremos da maior aresta de *S*. Sejam **A** e **D** esses vértices, e sejam a_p , d_p , a_Q e d_Q as respectivas coordenadas. Se $a_p > d_p$ e $a_Q < d_Q$ ou vice-versa, significa que **P** e **Q** pertencem a filhos diferentes de *S* e o raio intercepta a face comum entre os dois. Assim sendo, o parâmetro que define a interseção deve ser calculado.

Seja **I** esse ponto de interseção e **A**, **B**, **C** e **D** os vértices do setor (figura 7.2).

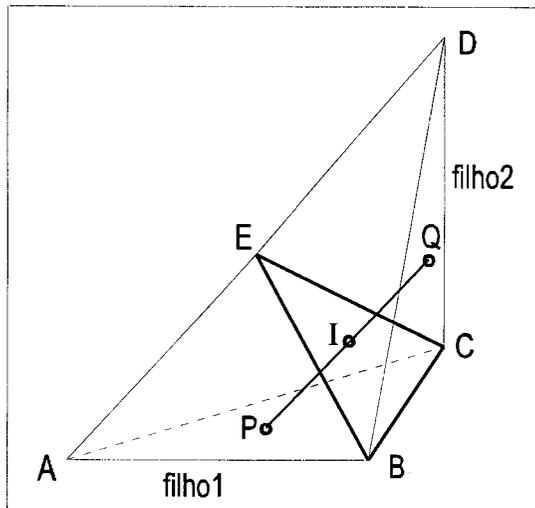


Figura 7.2

Um ponto genérico **R** do segmento **PQ** é dado pela equação

$$\mathbf{R} = (1-\lambda)\mathbf{P} + \lambda\mathbf{Q}.$$

Em particular o ponto **I** deve ter coordenadas em relação a **A** e a **D** iguais. Portanto,

$$(1-\lambda)a_p + \lambda a_Q = (1-\lambda)d_p + \lambda d_Q$$

de onde vem que

$$\lambda = \frac{a_p - d_p}{a_p - d_p - (a_Q - d_Q)}.$$

Para aplicar o mesmo processo, recursivamente, aos filhos do setor, é necessário converter as coordenadas dos pontos **P**, **Q** e **I** para os sistemas relativos a esses filhos. A chamada para o filho 1 requer os pontos **P** e **I**, enquanto a chamada para o filho 2 requer os pontos **Q** e **I**.

Converter as coordenadas de um ponto genérico **R** para o filho 1, corresponde a mudar o tetraedro de referência de **A, B, C** e **D** para **A, B, C** e **E**, na figura. Segundo o apêndice A, considerando-se que as coordenadas de **E** em relação a **A, B, C** e **D**, no tetraedro original, são iguais a $\frac{1}{2}$, 0, 0 e $\frac{1}{2}$, respectivamente, tem-se: b_R e c_R não se alteram, enquanto

$$e_R = 2 d_R \quad \text{e} \quad a_R = a_R - d_R.$$

Na situação particular em que o ponto se situe sobre a face que divide os filhos (como é o caso do ponto **I**), quando então $a_I = d_I$, em relação ao tetraedro original, tem-se que, em relação ao novo tetraedro, $a_I = 0$. Resta, portanto, como única operação aritmética necessária, calcular $e_I = 2 d_I$.

Para o ponto **P**, que se situa sobre a face **A, B, C**, tem-se que $d_P = 0$, em relação ao tetraedro original. Portanto, em relação ao novo tetraedro, $e_P = 0$ e a_P também não se altera.

As conversões relativas ao filho 2 são análogas.

Comparando esses cálculos com os do capítulo 5, relativos a triangulações genéricas, observa-se que, além de se calcular λ um menor número de vezes, também as conversões de coordenadas necessárias são mais simples. No total são economizadas diversas operações de ponto flutuante, incluindo algumas divisões, que são muito mais caras do que multiplicações, somas e subtrações, nos processadores modernos.

As demais operações necessárias para calcular as contribuições das células para a cor do pixel são as mesmas nos dois casos.

Embora não tenham sido feitas avaliações computacionais para comparar o desempenho do método de lançamento de raios com triangulações RA em relação a triangulações genéricas, a exposição acima indica que existem ganhos de eficiência. Além disso, vale enfatizar outros aspectos que permitem concluir pelo alto valor da abordagem aqui apresentada, no contexto de visualização volumétrica. Em primeiro lugar, por tratar-se de um algoritmo sintonizado com a filosofia de exploração dos modelos dinâmicos de multiresolução. O aprofundamento no percurso das árvores da estrutura pode ser interrompido em função de um critério de tolerância, estabelecido imediatamente antes do início da geração da imagem. O algoritmo encontrará os limites do modelo simplificado, com os necessários requisitos de consistência estrutural e continuidade da função, sem nenhum esforço adicional, além da comparação do erro associado aos vértices alcançados com aquele parâmetro de tolerância. Uma outra característica desse algoritmo a ser ressaltada, é que ele utiliza a mesma estrutura de dados adotada para o método de projeção, unificando as duas técnicas em torno de uma representação notoriamente mais econômica do que as estruturas adaptativas usuais.

7.2 Degenerações

Analisando os casos de degeneração mencionados no capítulo 5, uma interseção degenerada, que pode levar a um erro na determinação da face de saída, pode ser tratada, no caso de triangulação RA, da mesma maneira proposta para triangulações genéricas: sendo positivas as coordenadas baricêntricas de um ponto do plano em relação aos vértices de uma face do mesmo plano, fica confirmado que o ponto pertence àquela face.

O caso de célula degenerada, conforme já observado, não ocorre em triangulações RA.

Capítulo 8

Conclusão

Este trabalho mostrou a aplicabilidade das triangulações RA como base para a modelagem de campos escalares tridimensionais, em um sistema de visualização científica. A abordagem do problema iniciou por um aprofundamento do estudo dessas estruturas. Suas características geométricas e combinatórias, algumas já mencionadas em outros trabalhos, outras detectadas ao longo deste esforço, foram analisadas detalhadamente, resultando no estabelecimento de uma codificação para os elementos da estrutura. Essa codificação, associada ao caráter hierárquico das triangulações RA, possibilitou a definição de uma estrutura de dados extremamente compacta. Além de econômica, do ponto de vista do espaço ocupado, essa estrutura mostrou-se muito conveniente para a definição de algoritmos capazes de implementar, de forma eficiente, as técnicas de visualização tratadas.

A abordagem foi voltada para as técnicas de visualização direta, sendo estudada a implementação dos algoritmos de lançamento de raios (“ray casting”) e de projeção. Essas duas técnicas, em conjunto, proporcionam condições para se projetar um sistema que contemple requisitos variados: o método de projeção possibilita a geração rápida de imagens, em um esquema interativo de exploração dos dados, enquanto o método de lançamento de raios permite a obtenção de imagens de alta qualidade.

Foram também definidas as triangulações RA multiresolução, que facilitam a extração de modelos simplificados, baseados em critério de precisão, os quais são obtidos pela eliminação dos níveis inferiores de refinamento da estrutura. Criou-se uma metodologia para associar erros aos vértices, e indiretamente aos tetraedros intermediários da hierarquia. Essa metodologia inclui um mecanismo de propagação de erro que garante a consistência de modelos simplificados, definidos dinamicamente ao mesmo tempo que se percorre a estrutura para gerar uma imagem, sem necessidade de visitar os níveis inferiores. Também não é necessário criar uma representação à parte para o modelo simplificado. As triangulações RA multiresolução facilitam, ainda, a que um servidor atenda de forma eficiente a uma requisição de modelos progressivos emitida por um cliente. Para tanto, o servidor extrai uma série de modelos simplificados dinâmicos. A codificação para transmissão pode ser feita de forma extremamente compacta, e, diferente de outros esquemas que também utilizam estruturas bastante compactas para transmissão, esses modelos não exigem cálculos geométricos para sua recomposição no cliente.

Foi implementado o algoritmo de projeção para triangulações RA, juntamente com uma versão equivalente para triangulações genéricas, para fins de comparação empírica dos ganhos de eficiência. Os resultados obtidos indicam uma vantagem significativa para as triangulações RA, cuja execução gastou, em média, cerca de 60% do tempo

gasto pela outra. Essas medidas não constituem uma base rigorosa de comparação, uma vez que existem diferenças na forma de implementação desses algoritmos, quanto à metodologia de alocação da memória requisitada para representação das estruturas. No entanto, mostrou-se que tais diferenças penalizam a implementação relativa a triangulações RA, que deve, afinal, ampliar aquela margem de diferença.

Do ponto de vista do consumo de memória, a estrutura de dados utilizada para as triangulações RA requer em torno de 50% do espaço ocupado pela estrutura mais econômica descrita na literatura, para outras triangulações. São consideradas aqui apenas as estruturas que permitem ordenar as células com eficiência. A relação acima refere-se a uma estrutura específica para triangulações de Delaunay [Cignoni98b]. Qualquer outra exige um espaço adicional da ordem de $O(n)$ para ordenação.

Conclui-se, assim, que a utilização das triangulações RA como base para sistemas de visualização volumétrica se justifica, do ponto de vista de economia, eficiência e flexibilidade. Resta considerar as condições em que a estrutura se aplica como modelo de representação de um campo escalar. É preciso lembrar que, a menos de situações particulares, as informações conhecidas do campo não se situam nos nós de uma triangulação RA. Assim sendo, sua aplicação depende de uma reamostragem da função, o que implica necessariamente na incorporação de erros aos dados. A argumentação a favor dessa abordagem está calcada na experiência já consolidada de reamostragem de campos escalares, cujos dados originais têm distribuição irregular, para os nós de uma malha regular, visando aproveitar as conveniências dessa última para a realização das tarefas necessárias. O modelo usado para a reamostragem deve ser escolhido de acordo com cada aplicação. As triangulações RA entram nesse contexto substituindo as malhas regulares como estruturas auxiliares para o tratamento de dados irregulares. A vantagem oferecida é a adaptatividade, que proporciona uma representação não apenas econômica, mas também de boa qualidade, pois permite manter sob controle os erros de aproximação.

Desde o início deste trabalho, esperava-se que fosse existir uma zona nebulosa em que haveria dúvida quanto à conveniência de adotar uma ou outra dessas estruturas auxiliares. Por exemplo, para conjuntos de dados cuja heterogeneidade da distribuição é pouco acentuada. No entanto, com a crescente importância da característica de multiresolução, impulsionada, entre outros fatores, pela grande proliferação de aplicações via rede, a perspectiva de um aproveitamento efetivo das triangulações RA está muito ampliada. Para corroborar esta opinião, pode ser citada uma iniciativa recente, que utiliza subdivisões simpliciais para prover multiresolução até mesmo a campos cuja amostragem original seja regular [Zhou97].

A principal limitação para a aplicação das triangulações RA, bem como das malhas regulares, está no fato de que elas não garantem o posicionamento de seus nós sobre fronteiras irregulares. Esse aspecto assume especial importância em simulação computacional, em casos onde as regiões críticas estejam localizadas nessas fronteiras. A adaptatividade atenua esse inconveniente, colocando mais uma vez as triangulações RA em posição intermediária entre malhas regulares e triangulações genéricas. De qualquer forma, para a finalidade específica de visualização, aquela característica é menos restritiva. A possibilidade de explorar os dados usando diferentes técnicas de geração de imagens, ou mesmo combinações dessas técnicas, e ainda com variados

graus de precisão, proporciona ao cientista uma grande riqueza de recursos que permitem chegar a uma interpretação segura dos dados.

Trabalhos Futuros

Algumas questões ficaram em aberto neste trabalho. A implementação das técnicas estudadas é uma necessidade que se apresenta de imediato, incluindo uma revisão da estrutura de dados, uma vez que a versão implementada não é a que foi descrita.

A técnica de extração de isosuperfícies foi apenas citada. Pode-se, no entanto, vislumbrar diversas maneiras de incorporá-la a um sistema de visualização baseado em triangulações RA. A primeira proposta que se apresenta, é inspirada na filosofia de multiresolução: procurar as células interceptadas pela superfície desejada, concomitantemente com a busca dos elementos constitutivos de um modelo simplificado. Para cada uma dessas células, calcula-se a geometria da porção da superfície no seu interior, que fica definida por um conjunto de 1 ou 2 triângulos, os quais podem ser imediatamente projetados e/ou incorporados a uma descrição da superfície. Desse modo, pelo menos a princípio, será gerada apenas uma lista de triângulos, sem relações de conectividade. No entanto, se a aplicação exigir, essa proposta pode ser refinada, com base na própria organização hierárquica da estrutura, para que aquelas relações sejam identificadas de forma eficiente [Persiano94].

O algoritmo de ordenação das células, descrito no contexto do método de visualização direta por projeção dos tetraedros, também tem grande valor para exibição de múltiplas isosuperfícies em uma mesma imagem. Essa operação requer que as superfícies mais externas sejam transparentes e que suas cores sejam combinadas com as cores das mais internas. Para alcançar esse efeito, é necessário que as partes mais distantes do observador sejam projetadas antes das mais próximas, o que pode ser feito com auxílio daquele algoritmo. A extrema eficiência da ordenação por visibilidade das células de uma triangulação RA, permite prever que a sobrecarga dessa tarefa sobre o processo de exibição das isosuperfícies será menor do que a que ocorre em outras estruturas irregulares.

Outra questão que está sendo estudada, já com alguns algoritmos propostos, é a paralelização dos métodos apresentados. Novamente a organização hierárquica se mostra muito conveniente, permitindo implementar facilmente diversos esquemas de decomposição do domínio. Particularmente interessante é a facilidade de se obter uma decomposição constituída de blocos convexos de tamanhos aproximadamente iguais.

A extensão do esquema dinâmico de obtenção de modelos simplificados a modelos especificados por funções de tolerância heterogêneas, é outra questão que merece atenção. Também a experimentação da infra-estrutura apresentada, com modelos óticos mais elaborados, deve ser explorada.

Finalmente, um campo de investigação promissor, é a possibilidade de utilizar as triangulações RA como malhas dinâmicas. A facilidade de aumentar e diminuir o grau de refinamento da malha em uma região do domínio, é um recurso desejável em aplicações que envolvam, por exemplo, a movimentação de objetos, em torno dos quais se situam regiões críticas. Seja, por exemplo, um corpo em movimento em um fluido. A região crítica desse fenômeno varia continuamente. Vale considerar as triangulações RA como opção para esse tipo de aplicações, uma vez que o processo de refiná-las e

desfazer o refinamento não envolve, a princípio, cálculos onerosos. Sua aplicabilidade, no entanto, depende da possibilidade de servirem de base para as simulações a serem realizadas, pois o próprio dinamismo da malha torna inviável a estratégia de reamostragem.

Apêndice A

Coordenadas Baricêntricas

A.1 Sistemas de coordenadas baricêntricas

Parte desta seção é uma transcrição de [Persiano96].

Combinação baricêntrica⁶

A combinação baricêntrica de dois pontos \mathbf{O} e \mathbf{P} com coeficientes $u \in \mathbb{R}$, denotada $(1-u)\mathbf{O} + u\mathbf{P}$, é o ponto \mathbf{Q} dado por $\mathbf{Q} = \mathbf{O} + u(\mathbf{P} - \mathbf{O})$. Note que $\mathbf{P} - \mathbf{O}$ é um vetor, assim como $u(\mathbf{P} - \mathbf{O})$, e portanto \mathbf{Q} tem sentido de ponto do espaço euclidiano.

A figura A.1 ilustra a operação de combinação baricêntrica para diferentes valores do real u . O ponto \mathbf{Q} é sempre um ponto da reta que passa por \mathbf{O} e \mathbf{P} . Quando $0 \leq u \leq 1$, a combinação é um ponto do segmento \mathbf{OP} .

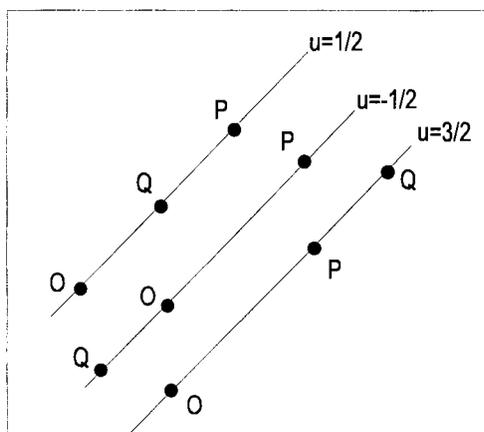


Figura A.1 - Combinações baricênticas de dois pontos do plano

Pode-se realizar combinação baricêntrica de mais de dois pontos. Se $u_i, i=0, 1, \dots, k$, são reais cuja soma é igual a 1, então a combinação baricêntrica dos pontos \mathbf{P}^i com esses coeficientes também resulta em um ponto, e é dada por:

⁶ A qualificação de “baricêntrica” deve-se à interpretação física da combinação como o centro de massa da distribuição de uma massa unitária onde cada parcela de massa u_i localiza-se no ponto \mathbf{P}^i .

$$u_0 \mathbf{P}^0 + u_1 \mathbf{P}^1 + \dots + u_k \mathbf{P}^k = \mathbf{P}^0 + \sum_{i=1}^k u_i (\mathbf{P}^i - \mathbf{P}^0) = \mathbf{P}^j + \sum_{\substack{i=0 \\ i \neq j}}^k u_i (\mathbf{P}^i - \mathbf{P}^j)$$

Um caso particular de combinação baricêntrica é a combinação convexa, caracterizada por serem todos os coeficientes não negativos. A combinação convexa de qualquer conjunto finito de pontos está necessariamente dentro da envoltória convexa desses pontos.

Outro tipo de combinação de pontos que faz sentido envolve coeficientes com soma nula. Nesse caso a combinação tem sentido de um vetor. Seja $(\mathbf{P}^i, i=0,1,\dots,m)$ um conjunto de pontos e sejam \mathbf{O} e \mathbf{Q} outros pontos definidos por:

$$\mathbf{O} = \sum_{i=0}^m u_i \mathbf{P}^i, \text{ com } \sum_{i=0}^m u_i = 1 \text{ e}$$

$$\mathbf{Q} = \sum_{i=0}^m v_i \mathbf{P}^i, \text{ com } \sum_{i=0}^m v_i = 1.$$

Então o vetor $\mathbf{Q} - \mathbf{O}$ é dado por

$$\mathbf{Q} - \mathbf{O} = (v_0 - u_0) \mathbf{P}^0 + (v_1 - u_1) \mathbf{P}^1 + \dots + (v_k - u_k) \mathbf{P}^k = w_0 \mathbf{P}^0 + w_1 \mathbf{P}^1 + \dots + w_k \mathbf{P}^k$$

onde

$$\sum_{i=0}^k w_i = \sum_{i=0}^k (v_i - u_i) = \sum_{i=0}^k v_i - \sum_{i=0}^k u_i = 1 - 1 = 0$$

Assim, combinações envolvendo pontos de um espaço euclidiano estão bem definidas somente quando a soma dos coeficientes é unitária ou nula. Combinações envolvendo pontos e vetores também fazem sentido, desde que os coeficientes dos pontos tenham soma nula ou unitária. Os coeficientes dos vetores podem ser quaisquer.

Coordenadas baricêntricas

Assim como as coordenadas cartesianas advém da noção de combinação linear de vetores linearmente independentes, pode-se associar coordenadas a pontos baseando-se na combinação baricêntrica de pontos não coplanares em ϵ^3 . Assim se $\mathbf{A}, \mathbf{B}, \mathbf{C}$ e \mathbf{D} são pontos não coplanares, qualquer ponto \mathbf{P} em ϵ^3 pode ser representado como combinação baricêntrica desses quatro pontos: $\mathbf{P} = a\mathbf{A} + b\mathbf{B} + c\mathbf{C} + d\mathbf{D}$, e dizemos que a quádrupla (a,b,c,d) são as coordenadas baricêntricas de \mathbf{P} em relação ao tetraedro formado por \mathbf{ABCD} . Note que, em particular, os pontos $\mathbf{A}, \mathbf{B}, \mathbf{C}$ e \mathbf{D} possuem, respectivamente, as coordenadas $(1,0,0,0), (0,1,0,0), (0,0,1,0)$ e $(0,0,0,1)$ em relação a esse tetraedro. Apesar de serem usadas quatro coordenadas para representar pontos em ϵ^3 , devido à sua soma ser sempre unitária, somente três delas são independentes.

As coordenadas baricêntricas podem ser calculadas pelas seguintes relações:

$$a = \frac{\text{vol}(\mathbf{PBCD})}{\text{vol}(\mathbf{ABCD})}, \quad b = \frac{\text{vol}(\mathbf{APCD})}{\text{vol}(\mathbf{ABCD})}, \quad c = \frac{\text{vol}(\mathbf{ABPD})}{\text{vol}(\mathbf{ABCD})}, \quad d = \frac{\text{vol}(\mathbf{ABCP})}{\text{vol}(\mathbf{ABCD})},$$

onde $\text{vol}(\mathbf{XYZW})$ representa o volume do tetraedro \mathbf{XYZW} com sinal [O'Rourke95].

Os pontos pertencentes ao plano da face oposta a um dos vértices, por exemplo \mathbf{A} (pode-se designar essa face também por \mathbf{A}), têm todos a coordenada baricêntrica relativa a esse vértice nula (no caso $a=0$). O plano paralelo a essa face, que passa pelo

ponto **A**, tem todos os seus pontos com coordenada $a=1$. Assim, se todas as coordenadas baricêntricas de um ponto estão no intervalo 0-1, o ponto está situado no interior do tetraedro. A figura A.2 ilustra essas observações para o triângulo (simplexo do ϵ^2).

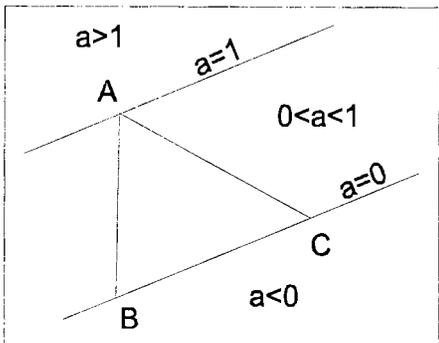


Figura A.2 - Coordenadas baricêntricas no plano

A.2 Mudança de sistema de coordenadas baricêntricas

Seja **P** um ponto qualquer do ϵ^3 , do qual se conhecem as coordenadas baricêntricas (a_p, b_p, c_p, d_p) em relação a um tetraedro **ABCD**. Sejam **E, F, G e H** também pontos do ϵ^3 , cujas coordenadas em relação a **ABCD** são (a_E, b_E, c_E, d_E) , (a_F, b_F, c_F, d_F) , (a_G, b_G, c_G, d_G) e (a_H, b_H, c_H, d_H) . Estes pontos definem um tetraedro, que por sua vez estabelece um outro sistema de referência. A questão que se coloca é determinar as coordenadas (e_p, f_p, g_p, h_p) de **P** no sistema definido por esse novo simplexo.

Pode-se escrever

$$\begin{aligned}
 \mathbf{P} &= a_p\mathbf{A} + b_p\mathbf{B} + c_p\mathbf{C} + d_p\mathbf{D} \\
 &= e_p\mathbf{E} + f_p\mathbf{F} + g_p\mathbf{G} + h_p\mathbf{H} \\
 &= e_p(a_E\mathbf{A} + b_E\mathbf{B} + c_E\mathbf{C} + d_E\mathbf{D}) + \\
 &\quad f_p(a_F\mathbf{A} + b_F\mathbf{B} + c_F\mathbf{C} + d_F\mathbf{D}) + \\
 &\quad g_p(a_G\mathbf{A} + b_G\mathbf{B} + c_G\mathbf{C} + d_G\mathbf{D}) + \\
 &\quad h_p(a_H\mathbf{A} + b_H\mathbf{B} + c_H\mathbf{C} + d_H\mathbf{D})
 \end{aligned}$$

de onde se extrai um sistema de equações lineares cujas incógnitas são (e_p, f_p, g_p, h_p) . Portanto, a solução desse sistema responde a questão proposta.

É importante observar, no entanto, que frequentemente se encontra a situação em que os tetraedros que definem os dois sistemas de referência têm vértices em comum. Cada vértice comum permite resolver trivialmente uma das incógnitas, reduzindo assim a ordem do sistema de equações. Seja, por exemplo, **E = A**. As coordenadas de **E** em relação ao tetraedro **ABCD** são, portanto, $(1,0,0,0)$, que será uma das colunas da matriz de coeficientes do sistema (neste caso, a coluna associada à incógnita e_p). Com isso pode-se resolver um sistema 3x3 e calcular e_p por substituição.

No caso em que os dois tetraedros tenham uma face em comum, a ordem do sistema de equações ficará reduzida a um. Sem perda de generalidade, suponhamos que **F = B, G = C e H = D**. A expressão resultante será

$$\begin{aligned}
\mathbf{P} &= a_p \mathbf{A} + b_p \mathbf{B} + c_p \mathbf{C} + d_p \mathbf{D} \\
&= e_p (a_E \mathbf{A} + b_E \mathbf{B} + c_E \mathbf{C} + d_E \mathbf{D}) + \\
&\quad f_p (\mathbf{B}) + \\
&\quad g_p (\mathbf{C}) + \\
&\quad h_p (\mathbf{D})
\end{aligned}$$

de onde vem que

$$a_E e_p = a_p \quad f_p + b_E e_p = b_p \quad g_p + c_E e_p = c_p \quad h_p + d_E e_p = d_p. \quad (I)$$

A.3 Aplicação ao método de lançamento de raios

A condição de compartilhamento de face, analisada na seção anterior, ocorre, por exemplo, no percurso radial do método de lançamento de raios, onde os sistemas de coordenadas relevantes estão associados a células vizinhas de uma subdivisão simplicial. Uma questão particularmente importante para aquele método, é a determinação das coordenadas baricêntricas do vértice **A**, em relação ao tetraedro **BCDE**, sendo conhecidas as coordenadas do vértice **E** em relação ao tetraedro **ABCD**, considerando que a face compartilhada é **BCD**. Para isso, basta considerar $\mathbf{P}=\mathbf{A}$. Assim, as coordenadas no sistema associado a **ABCD** serão: $a_p=a_A=1$ e $b_p=b_A=c_p=c_A=d_p=d_A=0$ e as equações (I) se tornam:

$$a_E e_A = 1 \quad f_A + b_E e_A = 0 \quad g_A + c_E e_A = 0 \quad h_A + d_E e_A = 0$$

de onde vem que

$$e_A = \frac{1}{a_E} \quad f_A = -\frac{b_E}{a_E} \quad g_A = -\frac{c_E}{a_E} \quad h_A = -\frac{d_E}{a_E}.$$

Recordando as relações entre os vértices da face comum, $\mathbf{F} = \mathbf{B}$, $\mathbf{G} = \mathbf{C}$ e $\mathbf{H} = \mathbf{D}$, as coordenadas de **A** em relação a **BCDE** serão:

$$e_A = \frac{1}{a_E} \quad b_A = -\frac{b_E}{a_E} \quad c_A = -\frac{c_E}{a_E} \quad d_A = -\frac{d_E}{a_E}.$$

Ainda sobre o método de lançamento de raios, é necessário converter para o novo sistema a representação da direção do raio. Sejam (a_L, b_L, c_L, d_L) e (e_L, f_L, g_L, h_L) as coordenadas da direção **L** nos dois sistemas:

$$\mathbf{L} = a_L \mathbf{A} + b_L \mathbf{B} + c_L \mathbf{C} + d_L \mathbf{D} = e_L \mathbf{E} + f_L \mathbf{F} + g_L \mathbf{G} + h_L \mathbf{H}$$

Considerando que uma direção é dada por um vetor, que por sua vez pode ser definido a partir de dois pontos (subtração), as equações (I) são válidas também para conversão de coordenadas baricêntricas de direções. Assim

$$e_L = \frac{a_L}{a_E} \quad f_L = b_L - e_L b_E \quad g_L = c_L - e_L c_E \quad h_L = d_L - e_L d_E.$$

Recorrendo novamente às relações supostas entre os vértices da face comum, $\mathbf{F} = \mathbf{B}$, $\mathbf{G} = \mathbf{C}$ e $\mathbf{H} = \mathbf{D}$, e considerando $\bar{b}, \bar{c}, \bar{d}, \bar{e}$ como coordenadas relativas ao sistema associado a **BCDE**, pode-se escrever

$$\bar{e}_L = \frac{a_L}{a_E} \quad \bar{b}_L = b_L - \bar{e}_L b_E \quad \bar{c}_L = c_L - \bar{e}_L c_E \quad \bar{d}_L = d_L - \bar{e}_L d_E.$$

Referências Bibliográficas

[Alfeld89]

P. Alfeld, "Scattered Data Interpolation in Three or More Variables", in T. Lyche e L.L. Schumaker (eds) "Mathematical Methods in CAGD", Academic Press, NY, pp. 1-33, 1989.

[Allgower87]

E. Allgower e S. Gnutzmann, "An Algorithm for Piecewise-linear Approximation for Implicitly Defined Two-dimensional Manifolds", SIAM Journal on Numerical Analysis, vol. 24, nº 2, 1987.

[Apolinário95]

A.L. Apolinário Jr., "Poligonização Adaptativa da Fronteira de Sólidos CSG", Tese de Mestrado, COPPE/UFRJ, 1995.

[Barbalho93]

V. Barbalho, "Geração de Malhas de Elementos Finitos Adaptativas de Sólidos CSG", Tese de Mestrado, COPPE/UFRJ, 1993.

[Bertolotto94]

M. Bertolotto, E. Bruzzone, L. De Floriani e E. Puppo, "Multiresolution Representation of Volume Data through Hierarchical Simplicial Complexes", Aspects of Visual Form Processing, pp. 73-82, World Scientific, Singapore, 1994.

[Blinn82]

J.F. Blinn, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces", Proceedings of Siggraph'82, ACM Computer Graphics, vol. 16, nº 3, pp. 21-29, 1982.

[Bowyer81]

A. Bowyer, "Computing Dirichlet Tessellations", The Computer Journal, vol. 24, nº 2, pp. 162-166, 1981.

[Bueno96]

L.P. Bueno, "Minimização da Energia de Deformação na Modelagem de Terreno", Tese de Doutorado, COPPE/UFRJ, março de 1996.

[Castelo90]

A. Castelo, S. Freitas e G. Tavares, "Approximation to Manifolds and its Application to Implicit ODES", Lecture in Applied Mathematics, nº 26, p. 29, AMS, 1990.

[Cendes87]

Z.J. Cendes e S.H. Wong, "C¹ Quadratic Interpolation Over Arbitrary Point Sets", IEEE Computer Graphics & Applications, pp. 8-16, novembro de 1987.

- [Chernyaev95]
E.V. Chernyaev, "Marching Cubes 33: Construction of Topologically Correct Isosurfaces", Proceedings of the Graphicon'95, Saint-Petersburg, Russia, julho de 1995.
- [Chiang98]
Y. Chiang e C.T. Silva, "External Memory Techniques for Isosurface Extraction for Scientific Visualization", 1998.
- [Cignoni94]
P. Cignoni, C. Montani e R. Scopigno, "MagicSphere: an Insight Tool for 3D Data Visualization", Computer Graphics Forum, vol. 13, nº 3, pp. 317-328, 1994.
- [Cignoni95]
P. Cignoni, C. Montani, D. Sarti e R. Scopigno, "On the Optimization of Projective Volume Rendering", in P. Zanarini, R. Scateni, J.J. van Wijk (eds.), "Visualization in Scientific Computing", Springer KG, Wien, pp. 58-71, 1995.
- [Cignoni98]
P. Cignoni, "Scientific Visualization Based on Simplicial Complexes", Tese de Doutorado, Università Degli Studi di Pisa, março de 1998.
- [De Floriani95]
L. De Floriani e E. Puppo, "Hierarchical Triangulation for Multiresolution Surface Description", ACM Transactions on Graphics, vol. 14, nº 4, pp. 363-411, outubro de 1995.
- [Drebin88]
R.A. Drebin, L. Carpenter e P. Hanrahan, "Volume Rendering", Computer Graphics, vol. 22, nº 4, pp. 65-74, 1988.
- [Edelsbruner89]
H. Edelsbrunner, "An Acyclicity Theorem for Cell Complexes in d Dimension", ACM Symposium on Computational Geometry, pp. 145-151, 1989.
- [Elvins92]
T.T. Elvins, "A Survey of Algorithms for Volume Visualization", Computer Graphics, vol. 26, nº 3, pp. 194-201, 1992.
- [Farin86]
G. Farin, "Triangular Bernstein-Bézier Patches", Computer Aided Geometric Design, nº 3, pp. 83-127, 1986.
- [Farin90]
G. Farin, "Curves and Surfaces for Computer Aided Geometric Design", seg. ed., Academic Press, 1990.
- [Foley93]
T.A. Foley, H. Hagen e G.M. Nielson, "Visualizing and Modeling Unstructured Data", The Visual Computer, vol. 9, pp. 439-449, 1993.
- [Garrity90]
M.P. Garrity, "Raytracing Irregular Volume Data", Computer Graphics, vol. 24, nº 5, pp. 35-40, 1990.

- [George91]
P.L. George, "Automatic Mesh Generation. Application to Finite Element Methods", Willey & Sons, 1991.
- [Giertsen92]
C. Giertsen, "Volume Visualization of Sparse Irregular Meshes", IEEE Computer Graphics & Applications, pp. 40-48, março de 1992.
- [Giertsen93]
C. Giertsen e J. Petersen, "Parallel Volume Rendering on a Network of Workstations", IEEE Computer Graphics & Applications, pp. 16-23, novembro de 1993.
- [Globus94] A. Globus e E. Raible, "Fourteen Ways to Say Nothing with Scientific Visualization", IEEE Computer, vol. 27, nº 7, pp. 86-88, julho de 1994.
- [Gomes89]
J.M. Gomes e G. Tavares, "Métodos Simpliciais em Computação Gráfica", 17º Colóquio Brasileiro de Matemática, SBM, 1989.
- [Gomes94]
J. Gomes e L. Velho, "Computação Gráfica: Imagem", Série de Computação e Matemática (IMPA/SBM), 1994.
- [Haber91]
R.B. Haber, B. Lucas e N. Collins, "A Data Model for Scientific Visualization with Provision for Regular and Irregular Grids", Proceedings of Visualization'91, pp. 298-305, outubro de 1991.
- [Hoppe96]
H. Hoppe, "Progressive Meshes", Proceedings of Siggraph'96, ACM Computer Graphics, pp. 99-108, 1996.
- [Itoh95]
T. Itoh e K. Koyamada, "Automatic Isosurface Propagation Using na Extrema Graph and Sorted Boundary Cell Lists", IEEE Transactions on Visualization and Computer Graphics, vol. 1, nº 4, pp. 319-327, dezembro de 1995.
- [Joe91]
B. Joe, "Construction of Three-dimensional Delaunay Triangulations Using Local Transformations", Computer Aided Geometric Design, vol. 8, pp. 123-142, 1991.
- [Kajiya84]
J.T. Kajiya e B.P.Von Herzen, "Ray Tracing Volume Densities", Proceedings of Siggraph'84, ACM Computer Graphics, vol. 18, nº 4, pp. 165-174, 1984.
- [Kaufman94]
A. Kaufman, K.H. Höhne, W. Krüger, L. Rosenblum e P. Schröder, "Research Issues in Volume Visualization", IEEE Computer Graphics & Applications, pp. 63-67, março de 1994.
- [Lacroute94]
P. Lacroute e M. Levoy, "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation", Proceedings of Siggraph'94, pp. 451-457, 1994.

[Laur91]

D. Laur e P. Hanrahan, "Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering", ACM Computer Graphics, vol. 25, nº 4, julho de 1991.

[Lawson86]

C.L. Lawson, "Properties of n-Dimensional Triangulations", Computer-Aided Geometric Design, nº 3, pp. 231-246, 1986.

[Levoy88]

M. Levoy, "Display of Surfaces from Volume Data", IEEE Computer Graphics & Applications, pp. 29-37, maio de 1988.

[Levoy90a]

M. Levoy, "A Hybrid Ray Tracer for Rendering Polygon and Volume Data", IEEE Computer Graphics & Applications, pp. 33-40, março de 1990.

[Levoy90b]

M. Levoy, "Efficient Ray Tracing of Volume Data", ACM Transactions on Graphics, vol. 9, nº 3, pp. 245-261, 1990.

[Livnat96]

Y. Livnat, H-W. Chen e C.R. Johnson, "A Near Optimal Isosurface Extraction Algorithm Using the Span Space", IEEE Transactions on Visualization and Computer Graphics, vol. 2, nº 1, pp. 73-84, março de 1996.

[Lorensen87]

W.E. Lorensen e H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", ACM Computer Graphics, vol. 21, nº 4, pp. 163-169, julho de 1987.

[Ma95]

K.L. Ma, "Parallel Volume Ray-Casting for Unstructured-Grid Data on Distributed-Memory Architectures", IEEE/ACM Parallel Rendering Symposium, pp. 23-30, 1995.

[Max86a]

N. Max, "Atmospheric Illumination and Shadows", Proceedings of Siggraph'86, ACM Computer Graphics, vol. 20, nº 4, pp. 117-124, 1986.

[Max86b]

N. Max, "Light Diffusion Through Clouds and Haze", Computer Vision, Graphics and Image Processing, vol. 33, pp. 280-292, 1986.

[Max90]

N. Max, P. Hanrahan e R. Crawfis, "Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions", Computer Graphics, vol. 24, nº 5, pp. 27-33, 1990.

[Max93]

N. Max, B. Becker e R. Crawfis, "Flow Volumes for Interactive Vector Field Visualization", Proceedings of Visualization'93, pp. 19-25, outubro de 1993.

- [Max95a]
N. Max, "Optical Models for Direct Volume Rendering", IEEE Transactions on Visualization and Computer Graphics, vol. 1, nº 2, pp. 97-108, 1995.
- [Max95b]
N. Max e R. Crawfis, "Advances in Scientific Visualization", apresentado no IS&T/SPIE Symposium on Electronic Imaging: Science and Technology, San Jose, 1995.
- [McCormick87]
B.H. McCormick, T.A. DeFanti e M.D. Brown, eds. "Visualization in Scientific Computing", Computer Graphics, vol. 21, nº 6, novembro de 1987.
- [Mesh91]
S. Meshkat, J. Ruppert e H. Li, "Three-dimensional Unstructured Grid Generation Based on Delaunay Tetrahedrization", Proceedings of The Third International Conference on Numerical Grid Generation, pp. 841-851, junho de 1991.
- [Neider93]
J. Neider, T. Davis e M. Woo, OpenGL Programming Guide, Addison Wesley, 1993.
- [Neves97a]
J.M.R. Neves, "Interpolação Polinomial com Retalhos Simpliciais de Bézier", Relatório Técnico 001-97, Departamento de Computação / UFOP, janeiro de 1997.
- [Neves97b]
J.M.R. Neves e R.C.M. Persiano, "Visualizing Scalar Fields Represented by Adaptive Square Triangulations", X Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, pp. 95-102, outubro de 1997.
- [Nielson93]
G.M. Nielson, "Scattered Data Modeling", IEEE Computer Graphics & Applications, pp. 60-70, janeiro de 1993.
- [Nielson94]
G.M. Nielson, P. Brunet, M. Gross, H. Hagen e S.V. Klimenko, "Research Issues in Data Modeling for Scientific Visualization", IEEE Computer Graphics & Applications, pp. 70-73, março de 1994.
- [O'Rourke94]
J. O'Rourke, "Computational Geometry in C", Cambridge University Press, 1994.
- [Persiano91]
R.M. Persiano, M. Salim e L.P. Bueno, "Boundary Evaluation of CSG Solids by Simplicial Subdivision", COMPUGRAPHICS'91, Portugal, 1991.
- [Persiano93]
R.M. Persiano, J.L.D. Comba e V. Barbalho, "An Adaptive Triangulation Refinement Scheme and Construction", VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, SBC, pp. 259-266, 1993.
- [Persiano94]
R.M. Persiano e A. Apolinário. "Boundary Evaluation of CSG Models by

Adaptative Triangulation”. Proceedings of CSG 94 Conference - Set-Theoretic Solid Modelling : Techniques and Applications.

[Persiano96]

R.M. Persiano, “Bases da Modelagem Geométrica”, 10ª Escola de Computação, UNICAMP, julho de 1996.

[Popovic97]

J. Popovic e H. Hoppe, “Progressive Simplicial Complexes”, Proceedings of Siggraph’97, ACM Computer Graphics, pp. 217-224, 1997.

[Port84]

T. Porter e T. Duff, “Compositing Digital Images”, Proceedings of Siggraph’84, ACM Computer Graphics, vol. 18, nº 4, pp. 253-259, 1984.

[Prep85]

F.P. Preparata e M.I. Shamos, “Computational Geometry: An Introduction”, Springer-Verlag, New York, NY, 1985.

[Roseman94]

L.J. Rosenblum, “Research Issues in Scientific Visualization”, IEEE Computer Graphics & Applications, pp. 61-63, março de 1994.

[Rush87]

H.E. Rushmeier e K.E. Torrance, “The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium”, Proceedings of Siggraph’87, ACM Computer Graphics, vol. 21, nº 4, pp. 293-302, 1987.

[Sabella88]

P. Sabella, “A Rendering Algorithm for Visualizing 3D Scalar Fields”, Proceedings of Siggraph’88, ACM Computer Graphics, vol. 22, nº 4, pp. 51-58, 1988.

[Salin91]

M. Salim, L.P. Bueno e R.C.M. Persiano, “Aproximação Linear por Partes de Variedades Implicitamente Definidas Usando Poda no Espaço”, IV Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, SBC, p. 27, 1991.

[Samet90]

H. Samet, “Applications of Spatial Data Structures”, Addison Wesley Publishing Co., 1990.

[Schroeder92]

W.J. Schroeder, J.A. Zarge e W.E. Lorensen, “Decimation of Triangle Meshes”, Proceedings of Siggraph’92, ACM Computer Graphics, vol. 26, pp. 65-70, julho de 1992.

[Schroeder98]

W. Schroeder, K. Martin e W. Lorensen, “The Visualization Toolkit”, segunda edição, Prentice Hall PTR, 1998.

[Shen95]

H.-W. Chen e C. R. Johnson, “Sweeping Simplicies: A Fast Isosurface Extraction

Algorithm for Unstructured Grids”, IEEE Visualization’95, pp. 143-150, outubro de 1995.

[Shen96]

H.-W. Shen, C.D. Hansen, Y. Livnat e C.R. Johnson, “Isosurfacing in Span Space with Utmost Efficiency (ISSUE)”, IEEE Visualization’96, outubro de 1996.

[Shirley90]

P. Shirley e A. Tuchman, “A Polygonal Approximation to Direct Scalar Volume Rendering”, Computer Graphics, vol. 24, nº 5, pp. 63-70, 1990.

[Silva95]

C. Silva, J. Mitchell e A. Kaufman, “Automatic Generation of Triangular Irregular Networks Using Greedy Cuts”, IEEE Visualization '95, pp. 201-208, 1995.

[Silva96a]

C.T. Silva, J.S.B. Mitchell e A.E. Kaufman, “Fast Rendering of Irregular Grids”, ACM/IEEE Volume Visualization’96, pp. 15-22, 1996.

[Silva96b]

C.T. Silva, “Parallel Volume Rendering of Irregular Grids”, PhD Thesis, State University of New York at Stony Brook, dezembro de 1996.

[Soares97]

I.P. Soares e J.G.V. Moreira, “Visualização de Dados Volumétricos Escalares, em Malhas Adaptativas, em Arquiteturas Paralelas”, Relatório Técnico 012-97, Departamento de Computação / UFOP, novembro de 1997.

[Speray90]

D. Speray e S. Kennon, “Volume Probes: Interactive Data Exploration on Arbitrary Grids”, Computer Graphics, vol. 24, nº 5, pp. 5-12, 1990.

[Stein94]

C. Stein, B. Becker e N. Max, “Sorting and Hardware Assisted Rendering for Volume Visualization”, ACM Volume Visualization Symposium’94, pp. 83-89, 1994.

[Upton88]

C. Upton e M. Keeler, “The V-Buffer: Visible Volume Rendering”, Proceedings of Siggraph’88, ACM Computer Graphics, vol. 22, nº 4, pp. 59-64, 1988.

[Watson81]

D.F. Watson, “Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes”, The Computer Journal, vol. 24, nº 2, pp. 167-172, 1981.

[Watt93]

A. Watt, “Three-Dimensional Computer Graphics”, segunda edição, Addison-Wesley, 1993.

[Weatherill94]

N.P. Weatherill e O. Hassan, “Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints”, International Journal for Numerical Methods in Engineering, vol. 37, pp. 2005-2039, 1994.

[Westover90]

L. Westover, "Footprint Evaluation for Volume Rendering", Proceedings of Siggraph'90, ACM Computer Graphics, vol. 24, nº 4, pp. 367-376, 1990.

[Wilhelms91]

J. Wilhelms e A.V. Gelder, "A Coherent Projection Approach for Direct Volume Rendering", Proceedings of Siggraph'91, ACM Computer Graphics, vol. 25, nº 4, pp. 275-284, 1991.

[Wilhelms92]

J. Wilhelms e A.V. Gelder, "Octrees for Faster Isosurface Generation", ACM Transactions on Graphics, vol. 11, nº 3, pp. 201-227, julho de 1992.

[Wilhelms93]

J. Wilhelms, "Pursuing Interactive Visualization of Irregular Grids", The Visual Computer, vol. 9, pp. 450-458, 1993.

[Wilhelms96]

J. Wilhelms, A.V. Gelder, P. Tarantino e J. Gibbs, "Hierarchical and Parallelizable Direct Volume Rendering for Irregular and Multiple Grids", Proceedings of Visualization'96, pp. 57-64, 1996.

[Williams92a]

P.L. Williams, "Interactive Direct Volume Rendering of Curvilinear and Unstructured Data", PhD thesis, University of Illinois at Urbana-Champaign, 1992.

[Williams92b]

P.L. Williams e N. Max, "A Volume Density Optical Model", ACM Workshop on Volume Visualization, pp. 61-68, 1992.

[Williams92c]

P.L. Williams, "Visibility Ordering Meshed Polyhedra", ACM Transactions on Graphics, vol. 11, nº 2, pp. 103-126, abril de 1992.

[Williams98]

P.L. Williams, N. Max e C.M. Stein, "A High Accuracy Volume Renderer for Unstructured Data", IEEE Transactions on Visualization & Computer Graphics, vol. 4, nº 1, março de 1998.

[Yagel93]

R. Yagel, "Volume Rendering Polyhedral Grids by Incremental Slicing", OSU-CISRC-10/93-TR35, Department of Computer and Information Science, The Ohio State University, outubro de 1993.
<http://www.cis.ohio-state.edu/~yagel/papers/1993/TR35.ps.gzip>

[Yagel96a]

R. Yagel, "Towards Real Time Volume Rendering", Proceedings of Graphicon'96, Vol. 1, Saint-Petersburg, Russia, pp. 230-241, 1996.

[Yagel96b]

R. Yagel, D. M. Reed, A. Law, P. Shih e N. Shareef, "Hardware Assisted Volume Rendering of Unstructured Grids by Incremental Slicing", Proceedings of Symposium on Volume Visualization, pp. 55-62, Nov. 1996.

[Zhou97]

Y. Zhou, B. Chen e A. Kaufman, "Multiresolution Tetrahedral Framework for Visualizing Volume Data", Proceedings of Visualization'97, IEEE Computer Society Press, pp. 135-142, Nov. 1997.