

A importância da correta interpretação do dado na visualização volumétrica através da história, evolução e aplicações do algoritmo Marching Cubes

Alexandre M. M. Filho¹, Lis Ingrid R. L. Custódio¹

¹Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro (UERJ)
Campus Maracanã – 20.550-900 – Rio de Janeiro – RJ – Brasil

Alexmaimartfilho@gmail.com, liscustodio@ime.uerj.br

Resumo: *A Visualização Volumétrica está presente em diversas áreas de ciência. Através da criação de imagens é possível explorar a estrutura de dados volumétricos e extrair as informações neles contidas. Neste artigo discutimos a importância da correta interpretação e representação de dados volumétricos. Nesta direção, apresentamos as etapas que constituem a geração de imagens que representam as informações contidas no dado através do algoritmo Marching Cubes, o mais popular algoritmo de extração de isosuperfícies. Através do Marching Cubes e suas aplicações, este estudo explora os principais desafios na interpretação de dados volumétricos e as causas e consequências de sua representação incorreta.*

Palavras-chave: *Visualização Volumétrica. Marching Cubes. Coerência Topológica.*

Abstract: *Volumetric Visualization is present in several areas of science. By creating images, it is possible to explore the structure of volumetric data and extract the information contained therein. In this article we discuss the importance of correct interpretation and representation of volumetric data. In this sense, we present the steps that constitute the generation of images that represent the information contained in the data using the Marching Cubes algorithm, the most popular isosurface extraction algorithm. Through Marching Cubes and its applications, this study explores the main challenges in interpreting volumetric data and the causes and consequences of its incorrect representation.*

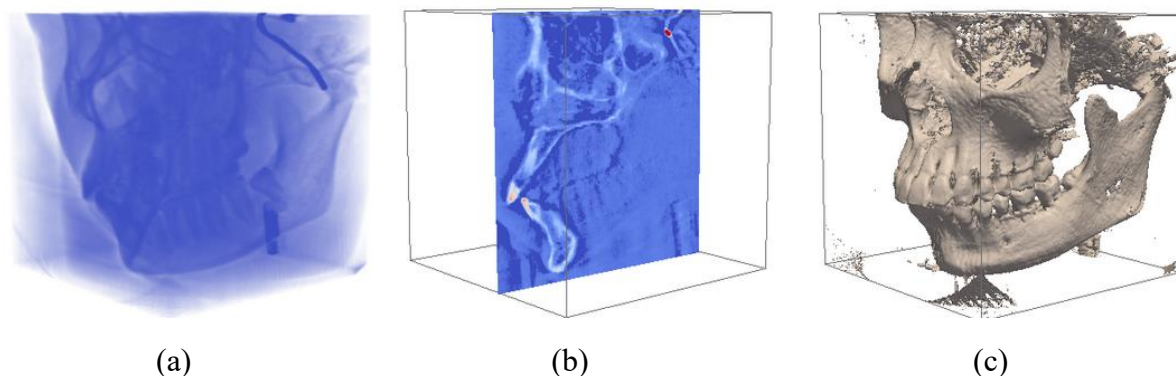
Keywords: *Volumetric Visualization. Marching Cubes. Topological Coherence.*

1. Introdução

Visualização Volumétrica é o nome dado ao conjunto de técnicas utilizadas na interpretação e representação das informações contidas em um dado volumétrico, através da qual é possível compreender e explorar as informações contidas neste dado. Como veremos neste artigo, as diferentes origens de um dado volumétrico faz com que este campo da Computação Gráfica tenha aplicação nas mais diversas áreas de ciência, como a medicina, geologia e física, sendo também amplamente utilizada na indústria dos jogos eletrônicos.

Uma vez obtidos, os dados são amostrados nos vértices de uma grade retangular tridimensional e a partir dessa amostragem são aplicadas técnicas capazes de gerar imagens representando as informações neles contidas.

Figura 1. Dado Skull. (a) Mapeamento de cor do dado amostrado em uma grade 512x512x512; (b) Uma das fatias que constituem o dado; (c) Representação de um conjunto de informações contidas no dado.



Fonte: Os autores.

Na Figura 1(a) apresentamos, através do mapeamento de cor, os valores de um dado volumétrico obtido através de tomografia computadorizada, amostrados em uma grade tridimensional de tamanho 512x512x512. Observe que, apesar de o dado estar amostrado em um domínio discretizado, o elevado número de vértices na grade e o pouco espaço entre eles, nos passa a impressão de existirem superfícies contínuas nesta imagem, estruturas que, como definiremos a seguir, são denominadas de isosuperfícies de um dado volumétrico. Na Figura 1(b), apresentamos uma das fatias que constituem o dado, forma como o dado é extraído. Através da união das informações contidas em fatias bidimensionais, como a ilustrada na Figura 1(b) é construído o dado volumétrico. Na Figura 1(c), apresentamos um modelo geométrico que representa uma das superfícies contidas no dado.

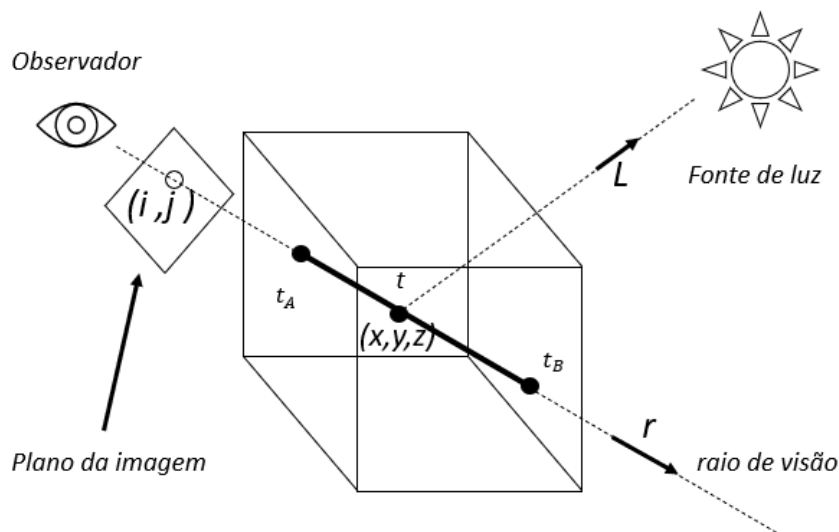
As técnicas de visualização volumétrica são classificadas em duas áreas: a visualização direta do volume e a visualização através de isosuperfícies. Essas áreas se diferem basicamente pela utilização ou não de representações intermediárias das informações contidas no dado para a geração da visualização final.

Na visualização direta do volume, uma vez determinada as regiões de interesse (informações as quais desejamos visualizar), os valores do dado volumétrico são mapeados em propriedades óticas como cor e opacidade, através de funções de transferência e, em seguida, é definida a direção de observação do dado (posição do observador). Dessa forma, a composição de um *pixel* (i, j) da imagem a ser gerada é dada pela composição dos elementos que um raio, partindo do observador e passando pelo ponto (i, j) encontra ao atravessar o volume, como ilustrado na Figura 2. Observe que, desta forma, para se obter uma imagem da região de interesse sob outro ponto de vista, será necessário processar todo o volume novamente e assim obter a imagem correspondente.

Na visualização através de isosuperfícies, a visualização das informações contidas no dado é gerada a partir da criação de modelos geométricos (malhas geralmente triangulares) que aproximam as isosuperfícies deste dado (como ilustrado na Figura 3). Uma vez gerados os modelos geométricos, são usados os métodos tradicionais de *rendering* de polígonos para a geração da visualização [Paiva et al., 1999]. Os objetos geométricos gerados são constituídos de malhas poligonais, uma coleção de faces que definem um objeto tridimensional. Observe

que, desta forma, a visualização da informação contida no dado se reduz a visualização do objeto geométrico gerado, tornando o processo de visualização computacionalmente menos custoso. Os avanços e desafios envolvidos nas técnicas utilizadas para a geração de modelos geométricos, relacionados a correta representação das informações contidas no dado volumétrico, são o foco deste trabalho.

Figura 2. Representação da visualização direta do volume.



Fonte: Adaptado do trabalho Introdução à Visualização Volumétrica [Paiva et al., 1999].

Neste artigo discutiremos a importância da correta interpretação de um dado volumétrico e as consequências da má interpretação das informações nele contidas através de um dos mais populares algoritmos de extração de isosuperfícies, o algoritmo Marching Cubes, proposto por Lorensen e Cline (1987). O problema da coerência topológica na geração da isosuperfície é particularmente crítico em aplicações médicas e geológicas, onde a integridade estrutural das informações extraídas pode impactar diretamente análises e decisões.

Por sua simplicidade e robustez o Marching Cubes foi amplamente estudado, sendo implementado em diversas ferramentas de visualização volumétrica e aprimorado ao longo dos anos até chegarmos a sua versão atual. Dessa forma, este artigo se concentra nos desafios e avanços relacionados à extração de isosuperfícies pelo Marching Cubes, com ênfase na correção de problemas topológicos que podem comprometer a qualidade da representação volumétrica.

Nas seções seguintes, apresentaremos o funcionamento, aplicações e evolução do algoritmo Marching Cubes, proposto por Lorensen e Cline (1987), e discutiremos os principais problemas que podem afetar topologicamente a representação das informações contidas no dado, abordando assim os desafios envolvidos na correta representação de um dado volumétrico. Na próxima seção, apresentaremos o algoritmo Marching Cubes, a base para as discussões que apresentaremos neste trabalho. Em seguida, apresentaremos algumas aplicações do algoritmo Marching Cubes e ferramentas de visualização que possuem o algoritmo em sua implementação. Finalmente, discutiremos os problemas topológicos presentes nas representações (das informações contidas no dado volumétrico) geradas pelo algoritmo e a sua evolução ao longo dos anos.

2. O algoritmo Marching Cubes

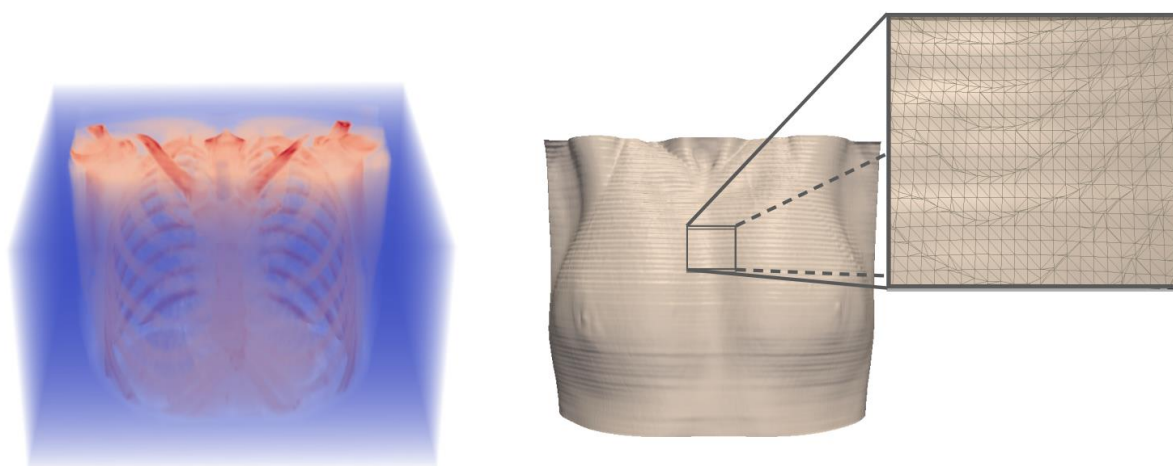
Por ser simples e robusto, o Marching Cubes é o algoritmo de visualização volumétrica mais utilizado dentre os algoritmos de extração de isosuperfícies e está implementado nas principais ferramentas de visualização. O algoritmo tem como dado de entrada um campo escalar, amostrado nos vértices de uma grade tridimensional, e um valor escalar, associado a isosuperfície de interesse. Como resultado, o algoritmo gera uma representação da isosuperfície na forma de uma malha poligonal triangular (Figura 3).

Dado um campo escalar $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ e um valor escalar α , definimos a isosuperfície associada a α o conjunto de pontos $x \in \mathbb{R}^3$ tais que $f(x) = \alpha$, e denotamos:

$$S_\alpha = \{x \in \mathbb{R}^3 : f(x) = \alpha\} \quad (1)$$

Dada uma grade tridimensional G onde está amostrado um campo escalar f , o Marching Cubes gera modelos geométricos que aproximam as isosuperfícies contidas no dado.

Figura 3. À esquerda, dado CT-Chest, obtido através de tomografia computadorizada, representado através de mapeamento de cor relacionada à densidade do tecido. À direita, malha triangular que aproxima uma das isosuperfícies contidas no dado.



Fonte: Os autores.

Se estivermos lidando com um dado médico, como o ilustrado na Figura 3, os valores amostrados sobre a grade G podem representar, por exemplo, a densidade de um tecido. Assim, ao fixarmos uma densidade específica (um valor escalar α), extraímos desse dado a informação desejada. Ao variarmos esse valor, podemos explorar o conjunto de dados de forma mais ampla, revelando diferentes aspectos de sua estrutura. Esse processo é exemplificado na Figura 4, que apresenta representações das informações contidas no dado CT-Chest por meio de isosuperfícies.

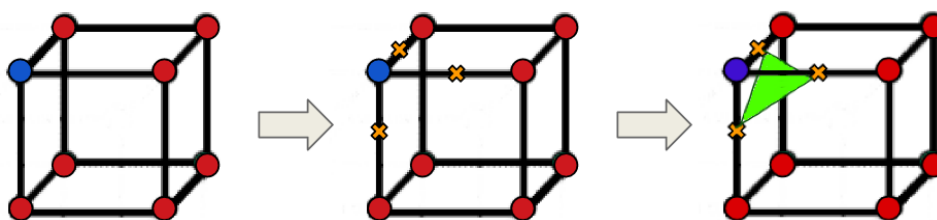
Figura 4. Representações obtidas através do algoritmo Marching Cubes de isosuperfícies contidas no dado CT-Chest (Figura 3).



Fonte: Os autores.

Para gerar a representação de uma isosuperfície S_α , utilizando a estratégia de divisão e conquista, para cada cubo da grade G , o Marching Cubes verifica de que modo S_α intersecta suas arestas. Para isto, classifica como negativo os vértices do cubo com valor escalar menor que α e positivo os vértices com valor escalar maior que α . Na Figura 5, ilustramos em azul o vértice (classificado como negativo) sobre o qual o valor do campo escalar é menor do que um dado valor de interesse α e, em vermelho, os vértices (classificados como positivos) sobre os quais o valor do campo escalar é maior que α . Desta forma, as arestas intersectadas serão as arestas com sinais opostos em seus vértices e uma aproximação do ponto de interseção é obtida através da interpolação linear dos valores do campo escalar sobre esses vértices.

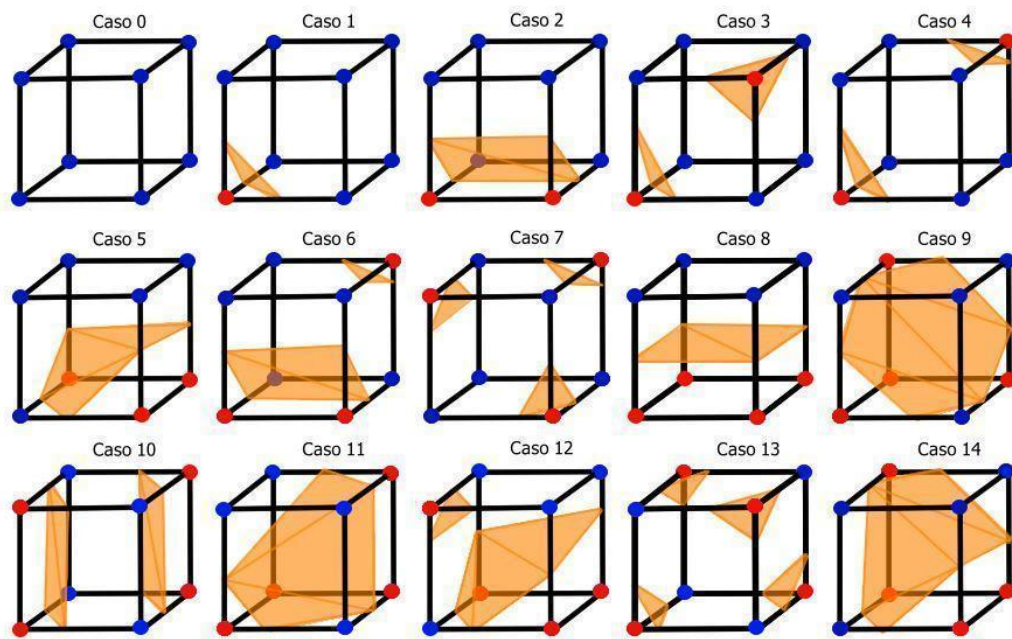
Figura 5: Exemplo da aproximação do comportamento da isosuperfície no interior de um cubo da grade.



Fonte: Os autores.

Uma vez obtidos os pontos de interseção, uma aproximação do comportamento da isosuperfície no interior do cubo é construída através de um ou mais triângulos. A triangulação gerada busca aproximar o comportamento da função linear que interpola os valores do campo escalar amostrados nos vértices do cubo. Como cada um dos 8 vértices de um cubo podem assumir os valores positivos ou negativos, existem então $2^8 = 256$ formas da isosuperfície intersectar essa célula.

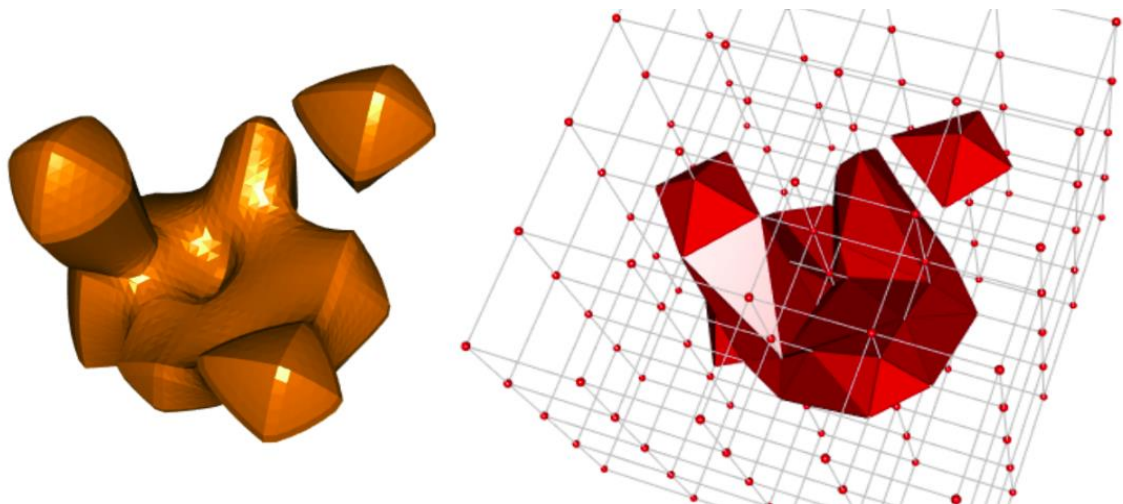
Figura 6. Tabela de triangulação do algoritmo Marching Cubes com configurações para diferentes interseções de isosuperfícies no cubo.



Fonte: Os autores.

Dessas 256 configurações notou-se que grande parte delas, na verdade, eram a rotação ou o caso complementar de uma mesma configuração. Com isso foi possível reduzir as 256 configurações para apenas 15 configurações, ilustradas na Figura 6, que irão compor a representação gerada da isosuperfície de interesse.

Figura 7. Esquerda: isosuperfície randomicamente gerada. Direita: modelo geométrico que aproxima a isosuperfície, gerado através do algoritmo Marching Cubes.



Fonte: Os autores.

Após percorrer toda a grade, as aproximações para o comportamento da isosuperfície em cada cubo são agrupadas em uma única malha triangular, como é possível observar na Figura 7.

3. Aplicações do Algoritmo Marching Cubes

Como mencionado na Seção 1, as diferentes origens de um dado volumétrico fazem com que a visualização volumétrica tenha aplicação em diferentes áreas da ciência. Nesta seção abordaremos algumas dessas aplicações.

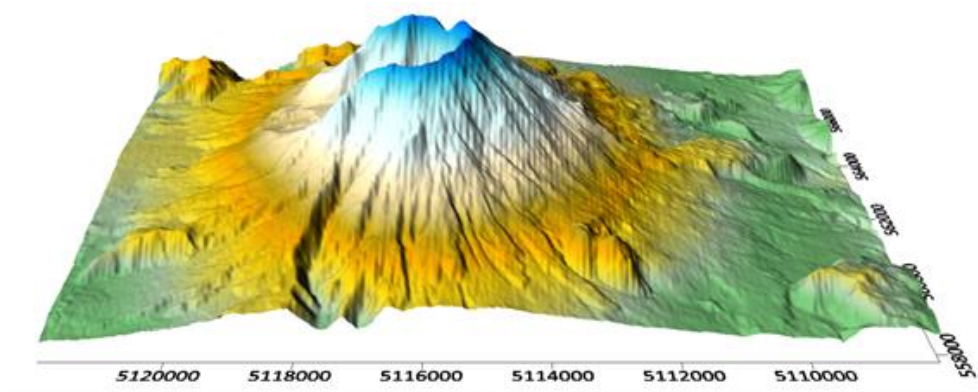
Na medicina, tendo como dado de entrada, dados obtidos através de Ressonância Magnética ou Tomografia Computadorizada, o Marching Cubes possibilita a criação de visualizações tridimensionais de órgãos e tecidos, o que permite uma melhor análise de exames e simulação de cirurgias.

As informações contidas no dado podem representar, por exemplo, a densidade do tecido em uma dada região examinada. Na Figura 1 e na Figura 3, apresentamos ilustrações de dados volumétricos (obtidos através de tomografia computadorizada) e modelos geométricos, construídos a partir desses dados, gerados pela ferramenta computacional de visualização volumétrica ParaView (2023). Ferramenta esta que utiliza algoritmo Marching cubes como extrator de isosuperfícies.

Na geologia, entre outras aplicações, o Marching Cubes é utilizado para modelar e reconstruir depósitos minerais e representações de terrenos. Como uma grande quantidade de dados volumétricos é criada nas atividades de mineração, o processamento visual desses dados auxilia no entendimento e análise das informações neles contidas (Seng et al. 2005).

Na Figura 8, podemos observar a malha gerada a partir do dado geológico, gerado pela ferramenta computacional *Surfer* (2023), desenvolvida pela empresa *GoldenSoftware* (2023). A ferramenta, que possui o Marching Cubes implementado com uma de suas funcionalidades, integra o modelo gerado pelo algoritmo a um mapeamento de cores que mapeia a altitude do relevo representado.

Figura 8. Superfície gerada em uma simulação com o software Surfer GoldenSoftware. Foi utilizado um mapa de cores para representar o relevo da superfície de nome Helens.

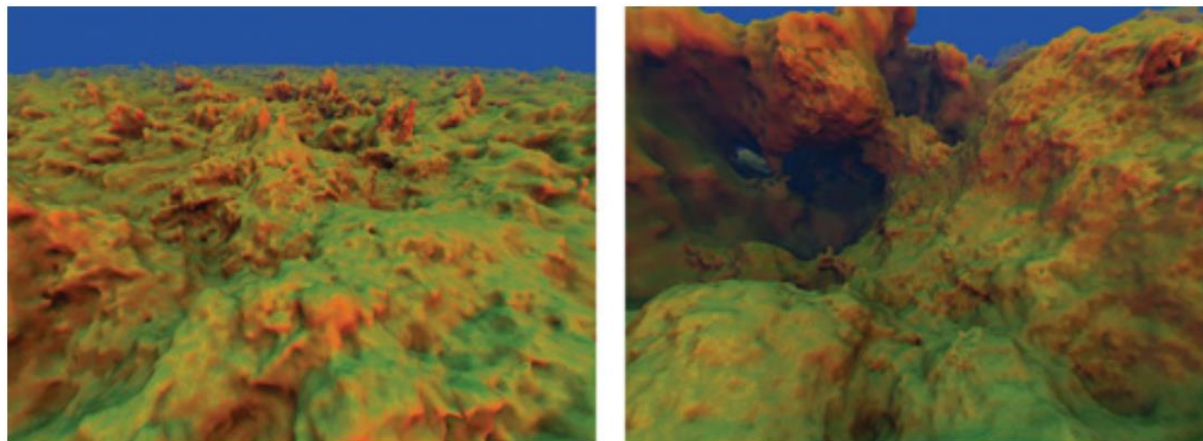


Fonte: Obtido através da ferramenta Surfer GoldenSoftware.

Na indústria de jogos, o algoritmo Marching Cubes é utilizado, por exemplo, para a geração de terrenos, criados em tempo real, que interagem com as ações do jogador principal. No decorrer do jogo, dados volumétricos são gerados randomicamente e o algoritmo é utilizado para gerar os modelos geométricos (malhas triangulares) capazes de representar a estrutura e relevo do terreno desejado. Essa técnica dá uma maior dinâmica ao jogo, fazendo com que a cada jogo, um novo cenário seja criado. Na Figura 9, apresentamos a imagem utilizada como exemplo em um tutorial para geração de terreno procedural desenvolvido pela NVIDIA (2023),

empresa referência no desenvolvimento de placas gráficas voltadas para jogos, onde a malha utilizada para representar o terreno (objeto geométrico ilustrado) foi gerada através do algoritmo Marching Cubes.

Figura 9. Geração de terreno procedural com a utilização do algoritmo Marching Cubes.

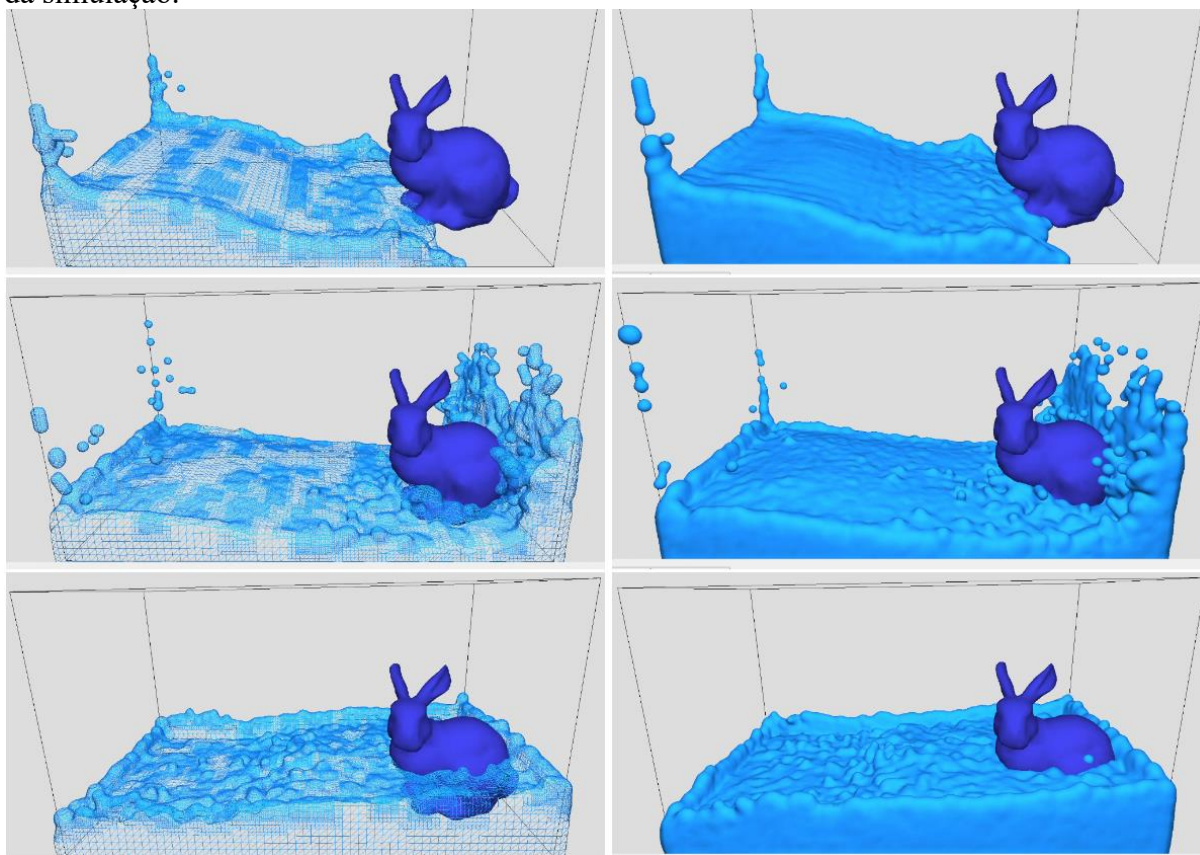


Fonte: Tutorial “*Generating Complex Procedural Terrains Using the GPU*”. NVIDIA Corporation.

O Marching Cubes é também integrado a técnicas de simulação e visualização de fluidos, frequentemente utilizada, tanto na engenharia como na área de efeitos especiais cinematográficos, para simular fenômenos naturais, tais como fogo, explosões, fumaça, água e ar. Durante a simulação, em cada instante de tempo, é gerado um dado volumétrico cuja isosuperfície de interesse representa a atual estrutura do fluido que está sendo simulado. Uma sequência de isosuperfícies pode representar um líquido sendo derramado em um recipiente (como ilustrado na Figura 10) ou fluindo ao redor de uma asa de um avião em uma simulação de voo. O Marching Cubes processa os dados volumétricos gerados durante a simulação, criando representações das isosuperfícies correspondentes [Zhu e Bridson, 2005; Solenthaler, 2007; Yu e Turk, 2010; Akinci, 2012].

Ao gerarmos a visualização dessas superfícies, uma após a outra, obtemos o equivalente ao movimento do fluido, permitindo assim a simulação do mesmo. Observe que tal aplicação, exige que as representações das superfícies sejam geradas e visualizadas rapidamente, o que se torna possível devido a eficiência do Marching Cubes. Em adicional, como veremos nas próximas seções, a possibilidade de integrar o Marching Cubes à estrutura de dados Octree acrescenta ao algoritmo eficiência e velocidade, características fundamentais para aplicações que exigem a geração das isosuperfícies em tempo real.

Figura 10. Resultado da utilização do Marching Cubes na geração das isosuperfícies resultantes da simulação.



Fonte: Artigo “*GPU-based Adaptive Surface Reconstruction for Real-time SPH Fluids*” (Shuchen Du e Takashi Kanai, 2014).

4. O Problema na Coerência Topológica do algoritmo Marching Cubes

O Marching Cubes, criado em 1987, passou por diversas alterações até ser o algoritmo que conhecemos hoje. A adoção generalizada do algoritmo fez com que a investigação e interesse no algoritmo fossem mantidos nesses mais de trinta anos desde a sua criação. Ao longo dos anos, melhorias foram propostas à versão original do algoritmo Marching Cubes para garantir a continuidade e coerência topológica da representação gerada.

Como veremos a seguir, os problemas de descontinuidade e incoerências topológicas na representação gerada pela Marching Cubes são causados por ambiguidades no comportamento do interpolante linear sobre as faces e interior do cubo. As ambiguidades ocorrem, pois, como vimos anteriormente, o algoritmo possui como dado de entrada o valor do campo escalar apenas sobre os vértices de uma grade tridimensional. Dessa forma, para algumas configurações de valores do campo escalar sobre os vértices de um cubo da grade existem duas ou mais maneiras de o interpolante se comportar sobre as suas faces e seu interior.

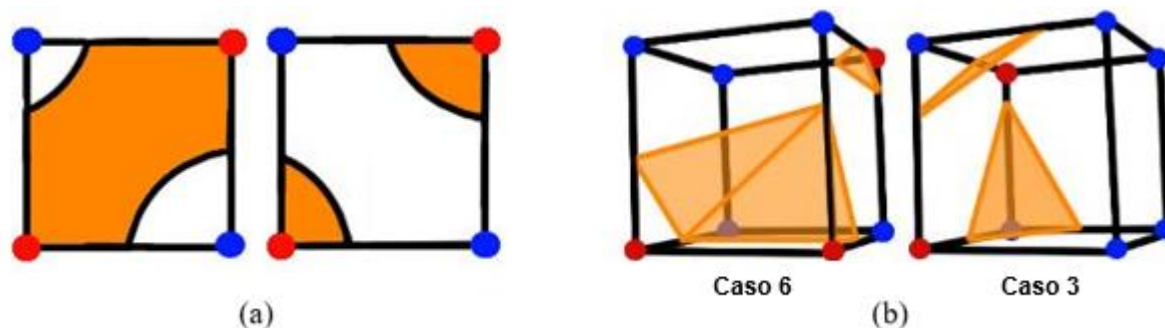
No Marching Cubes, o objetivo original de construir uma aproximação para a isosuperfície de interesse em toda a grade é reduzido a encontrar tal representação no interior de cada um dos cubos dessa grade, sendo posteriormente agrupadas para gerar a solução do problema inicial, estratégia de implementação conhecida como divisão e conquista. Dessa forma, as diferentes interpretações locais do dado, causadas pela existência de ambiguidades resultam em falhas na correta representação global do mesmo.

Nas próximas seções abordaremos os problemas relacionados às ambiguidades do interpolante linear sobre as faces e no interior do cubo, suas consequências e possíveis soluções para esses problemas.

4.1. Ambiguidade sobre as faces do cubo

Em uma face ambígua todas as arestas são intersectadas. Observe que esta configuração permite dois possíveis comportamentos do interpolante sobre a face, as regiões positivas (analogamente, as negativas) podem estar unidas ou separadas sobre a face, como ilustrado na Figura 11(a).

Figura 11. (a) Configurações possíveis para o comportamento do interpolante sobre uma face ambígua. (b) Consequência da ambiguidade sobre as faces do cubo na representação gerada.



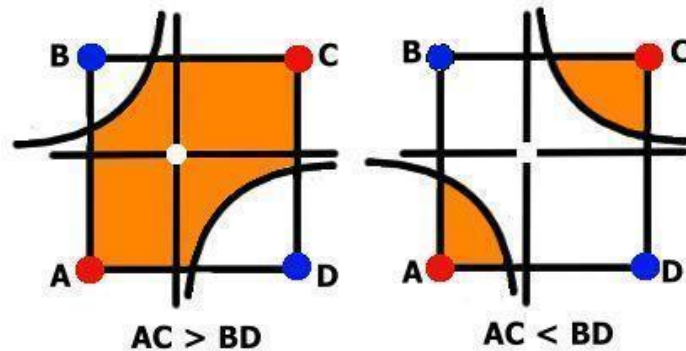
Fonte: Os autores.

Dessa forma, quando dois cubos vizinhos compartilham uma face ambígua, como o algoritmo interpreta o dado e gera a representação isoladamente em cada cubo, interpretações diferentes das informações contidas nessa face irão resultar em uma descontinuidade na malha gerada pelo algoritmo, como ilustrado na Figura 11(b). Esse problema foi observado por Nielson e Hamann (1991) que, no trabalho *The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes* propuseram um teste para resolver o problema de ambiguidade sobre as faces do cubo. O teste proposto consiste em um algoritmo capaz de determinar a forma correta de conectar os pontos de interseção sobre as arestas da face ambígua, evitando assim que o Marching Cubes tome decisões distintas ao interpretar a mesma face durante a análise de cubos vizinhos. A possibilidade (não observada no trabalho original) de diferentes comportamentos do interpolante para uma mesma configuração de sinais sobre os vértices de uma face implica na necessidade de inclusão de novos casos à tabela de triangulação. Em seu trabalho, Nielson e Hamann (1991), adicionaram 11 novas configurações à tabela proposta por Lorensen e Cline (1987).

4.1.1. O *Asymptotic Decider*

O *Asymptotic Decider* [Nielson e Hamann, 1991] detecta o correto comportamento do interpolante sobre a face ambígua e assim consegue determinar a forma correta de conectar os pontos de interseção sobre as arestas. O teste consiste em descobrir o valor do interpolante no ponto crítico sobre a face ambígua e através da comparação do valor obtido com os valores sobre os vértices da face determinar como unir os pontos de interseção.

Figura 12: Interpolação no ponto crítico sobre a face do cubo.



Fonte: Os autores.

Seja X o ponto crítico sobre a face ambígua, sua localização pode ser escrita em função dos valores amostrados sobre os vértices da face. Assim temos:

$$X = (S\alpha, T\alpha) \quad (2)$$

onde

$$S\alpha = A - B / A + C - B - D \quad (3)$$

$$T\alpha = A - D / A + C - B - D \quad (4)$$

Então:

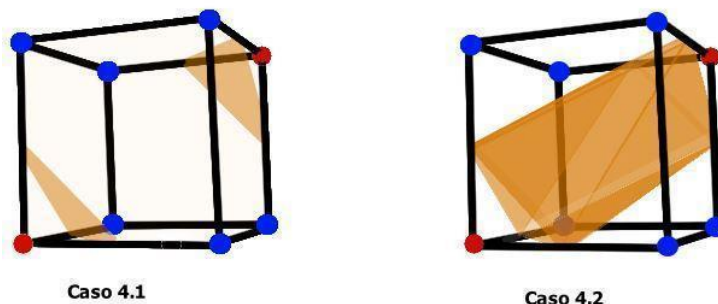
$$F(X) = F(S\alpha, T\alpha) = (A * C - D * B) / (A + C - B - D) \quad (5)$$

Sendo F a função que interpola bilinearmente os valores do campo escalar sobre os vértices da face.

Se $F(X) > \alpha$, ou seja, se o ponto crítico sobre a face for positivo, então existirá um caminho positivo conectando os vértices A e C. Caso contrário, as regiões positivas que contém os vértices A e C estão separadas sobre a face (Figura 12). Esse processo é feito em cada uma das faces ambíguas, certificando que cada cubo terá a correta interpretação, até que toda grade tenha sido lida e processada.

4.2. Ambiguidade no Interior do Cubo

Figura 13: Exemplo de ambiguidade interior (Caso 4).



Fonte: Os autores.

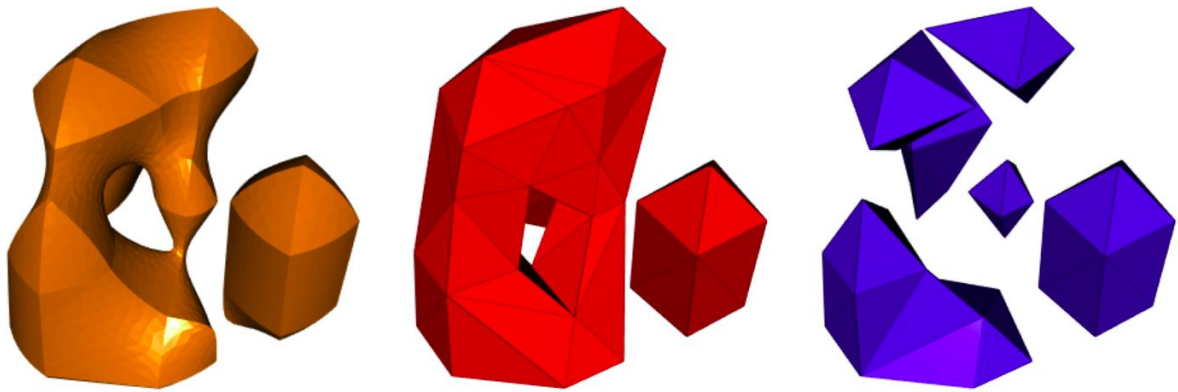
Em seu trabalho *On generating topologically consistent isosurfaces from uniform sample*, Natarajan (1994) observou que ainda era necessário adicionar novas configurações à tabela de triangulação devido a ambiguidade sobre as faces do cubo e foi o primeiro a observar a existência de ambiguidade no comportamento do interpolante no interior do cubo. Natarajan apontou a possibilidade de o interpolante formar túneis no interior do cubo e a necessidade de se incluir triangulações capazes de representá-los.

Posteriormente, Chernyaev (1995) propôs o Marching Cubes 33, neste trabalho, além de propor o teste para detectar corretamente o comportamento do interpolante no interior do cubo, complementou a tabela de triangulação com configurações coerentes com as novas possibilidades apontadas. Passando a apresentar o algoritmo Marching Cubes com 33 casos em sua tabela de triangulação. Essas 33 configurações foram posteriormente comprovadas eficientes para representar todos os comportamentos do interpolante dentro do cubo [Nielson, 2003]. Custódio et. al (2019), no trabalho, *An extended triangulation to the Marching Cubes 33 algorithm*, apresentou visualização interativa da tabela de triangulação proposta por Chernyaev (EMC33).

Assim como na ambiguidade sobre as faces do cubo, o problema da ambiguidade no interior do cubo ocorre quando uma configuração de sinais sobre os vértices permite diferentes comportamentos do interpolante, caracterizada pela existência de vértices diagonalmente opostos com mesmo sinal.

Na Figura 13, apresentamos as duas possíveis representações do caso 4, que originalmente admitia apenas a possibilidade de as regiões positivas estarem separadas no interior do cubo (caso 4.1). Como descrito por Chernyaev (1995), os dois vértices positivos, diagonalmente opostos, podem estar separados ou conectados por um caminho (região) positivo no interior do cubo e ambas configurações são possíveis, mas apenas uma delas estará correta topologicamente para uma dada configuração de valores sobre os vértices do cubo.

Figura 14. Esquerda: Representação de uma isosuperfície extraída de um dado volumétrico randomicamente gerado, amostrado em uma grade de tamanho 128 x 128 x 128. Centro e Direita: Representações (correta e incorreta) da mesma isosuperfície, agora considerando o mesmo dado amostrado em uma grade de tamanho 5 x 5 x 5, onde existem casos de ambiguidade no interior dos cubos.

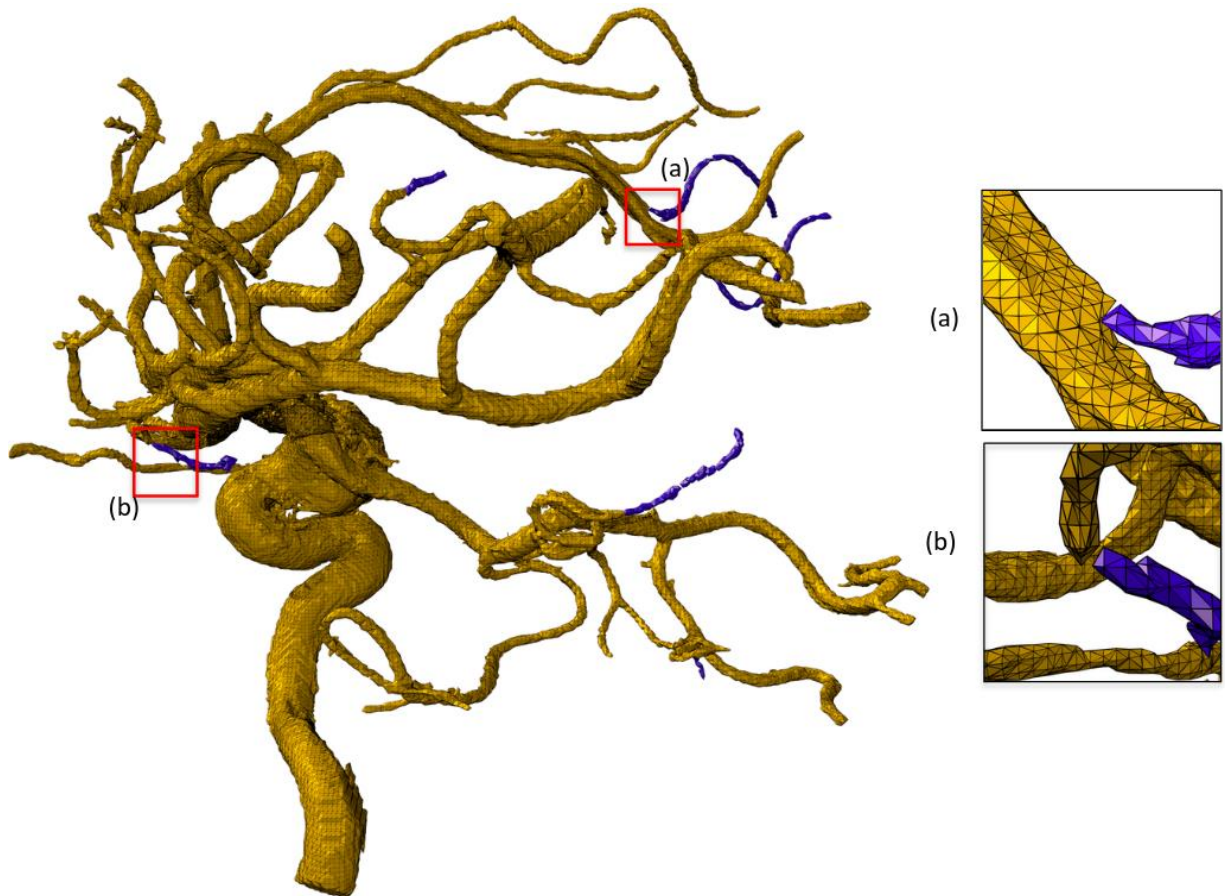


Fonte: Os autores.

A determinação incorreta dessas configurações pode acarretar em incoerências topológicas na representação da isosuperfície gerada pelo algoritmo, como podemos ver na Figura 14, que apresenta as consequências na representação do dado quando os casos ambíguos não são corretamente interpretados. Observe que as diferentes aplicações da visualização volumétrica fazem com que esse problema se estenda para as mais diversas áreas, cujas principais ferramentas de visualização ainda possuem em sua implementação as versões iniciais do algoritmo Marching Cubes, incapazes de interpretar corretamente o dado.

Na Figura 15, apresentamos a representação gerada pela ferramenta de visualização VTK [Schroeder et al., 2006] do dado médico Aneurisma. A principal componente conexa do dado está representada em dourado. Observe que, pelas características do dado, as componentes conexas em roxo deveriam, junto com a componente dourada, formar uma única componente conexa. No entanto, devido a problemas na interpretação da isosuperfície, estão separados da componente conexa principal.

Figura 15. Consequência da incorreta interpretação do dado volumétrico.

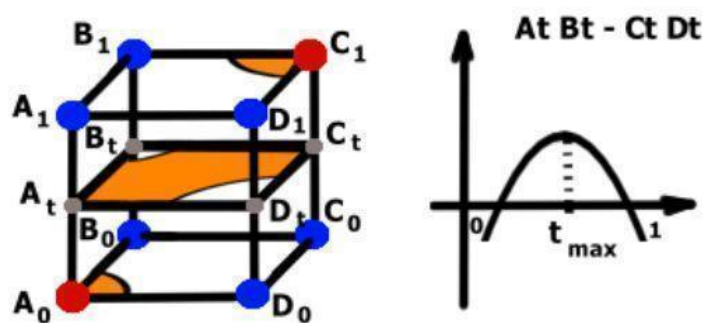


Fonte: Custódio et al., Practical Considerations on Marching Cubes 33 Topological Correctness.

4.2.1. Teste de ambiguidade interior

O teste proposto por Chernyaev (1995), uma extensão do teste de ambiguidade sobre as faces, consiste em verificar se existe (ou não) uma região positiva atravessando o cubo e conectando os vértices diagonalmente opostos, através da análise do comportamento do interpolante sobre planos que cortam o cubo paralelamente às suas bases, Figura 16.

Figura 16. Teste para verificar a ambiguidade interior.



Fonte: Os autores.

Sendo A_0, B_0, C_0 e D_0 os valores nos vértices tais que $t = 0$ e A_1, B_1, C_1 e D_1 os valores nos vértices tal que $t = 1$.

Observamos se:

$$A_0 * C_0 - B_0 * D_0 < 0 \quad (6)$$

$$A_1 * C_1 - B_1 * D_1 < 0 \quad (7)$$

Para que as regiões positivas estejam unidas através do interior do cubo, deve existir um valor real $t \in (0,1)$, tal que:

$$A_t * C_t - B_t * D_t > 0 \quad (8)$$

Onde,

$$A_t = A_0 + (A_1 - A_0)t \quad (9)$$

$$B_t = B_0 + (B_1 - B_0)t \quad (10)$$

$$C_t = C_0 + (C_1 - C_0)t \quad (11)$$

$$D_t = D_0 + (D_1 - D_0)t \quad (12)$$

Substituindo as Equações (9), (10), (11) e (12) na Equação (8), temos:

$$at^2 + bt + c > 0 \quad (13)$$

onde:

$$a = (A_1 - A_0)(C_1 - C_0) - (B_1 - B_0)(D_1 - D_0) \quad (14)$$

$$b = C_0(A_1 - A_0) + A_0(C_1 - C_0) \quad (15)$$

$$c = A_0C_0 - B_0D_0 \quad (16)$$

Se o valor da Equação (13) for positivo para algum $t \in (0,1)$, então existe uma região positiva atravessando o cubo, portanto as duas áreas positivas são conexas. Caso contrário, as regiões estarão separadas.

5. A versão atual do algoritmo Marching Cubes

Como vimos nas seções anteriores, o algoritmo Marching Cubes, desde sua criação, sofreu modificações que foram vitais para sua notoriedade. Modificações estas que corrigiram erros relacionados à qualidade de malha gerada e à coerência topológica da representação gerada.

Após algumas revisões, o algoritmo Marching Cubes chegou a configuração atual, com testes capazes de resolver as ambiguidades sobre as faces e interior do cubo (presentes no interpolante trilinear) e 33 triangulações capazes de representar qualquer comportamento do interpolante trilinear no interior do cubo, chegando a versão atual do algoritmo, conhecida como Marching Cubes 33, proposta por Chernyaev (1995).

Nielson (2003) realizou um estudo comprovando que as 33 configurações propostas por Chernyaev, são capazes de reproduzir qualquer comportamento do interpolante. No entanto, conforme observado por Etienne et al (2012), em seu trabalho “*Topology verification for isosurface extraction*”, as malhas geradas pelo Marching Cubes 33 ainda apresentavam problemas relacionados à coerência topológica da representação gerada. Problemas esses que

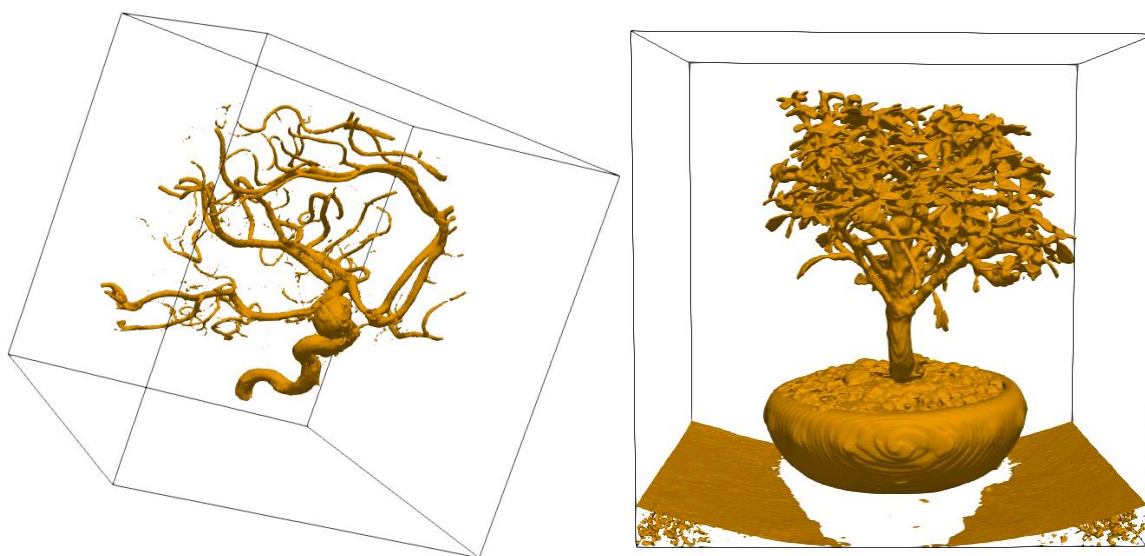
foram apresentados no trabalho *Practical considerations on marching cubes 33 topological correctness* [Custodio et. al, 2013], onde foram destacados erros presentes no algoritmo proposto por Chernyaev. Neste trabalho os autores propõem modificações no algoritmo Marching Cubes 33 capazes de garantir a coerência topológica da representação gerada.

6. Integrando o algoritmo Marching Cubes à estrutura de dados Octree

Juntamente com a sua simplicidade e robustez, a capacidade de obter resultados de alta resolução foi umas das características que contribuíram para tornar o Marching Cubes o principal algoritmo de extração de isosuperfícies. No entanto, o algoritmo não possui estratégias para uma localização eficiente dos cubos intersectados pela isosuperfície de interesse.

Durante a computação do dado de entrada, o algoritmo percorre todos os cubos da grade, um cubo de cada vez, para verificar se há ou não isosuperfícies a ser representada no interior de cada um deles. Tal abordagem representa um desperdício de tempo computacional, visto que, muitos dos cubos acessados estão vazios, ou seja, não são intersectados pela isosuperfície a ser representada [Cirne e Pedrini, 2011].

Figura 17. Modelos geométricos gerados através do algoritmo Marching Cubes. À direita, malha extraída do dado Aneurysm para o isovalor 100.1. À esquerda, malha extraída do dado Bonsai para o isovalor 39.5.



Fonte: Os autores.

Em nossos experimentos, como forma de exemplificar o elevado número de acessos à cubos vazios durante a computação do algoritmo Marching Cubes, fizemos uma pequena modificação no algoritmo proposto por Custódio et. al (2013) no trabalho *Practical considerations on marching cubes 33 topological correctness*. Foi acrescentado ao código a possibilidade de contabilizar o número de cubos acessados durante a computação que de fato eram intersectados pela isosuperfície de interesse, sendo possível assim contabilizar os cubos vazios. Utilizamos os dados Aneurysm e Bonsai, com os isovalores 100.1 e 39.5, respectivamente.

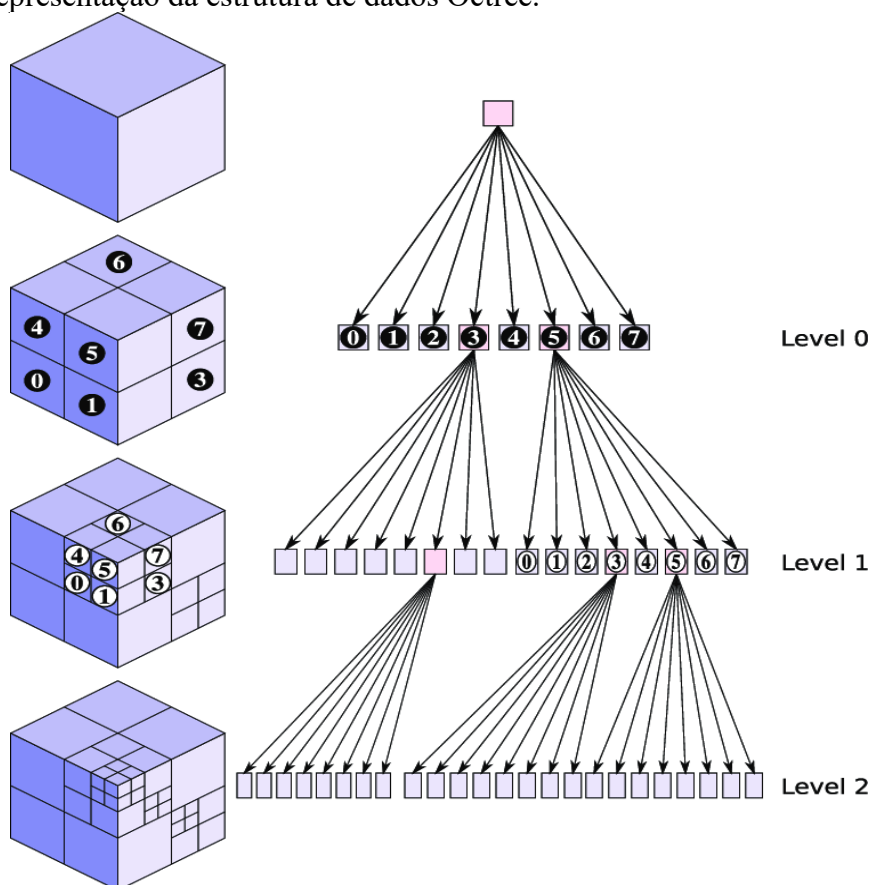
Na Tabela 1, apresentamos os resultados obtidos, onde podemos observar que, em média, para os exemplos utilizados, 97.01% dos cubos acessados pelo algoritmo durante a sua computação estavam vazios.

Tabela 1. Proporção dos cubos vazios encontrados pelo algoritmo Marching Cubes na geração das representações ilustradas da Figura 17.

Dado Volumétrico	n. de cubos vazios	n. total de tubos	% de Cubos vazios
Aneurysm(100.1)	16493577	16777216	98,31%
Bonsai(39.5)	16057648	16777216	95,71%

Fonte: Os autores

Figura 18. Representação da estrutura de dados Octree.



Fonte: Artigo “An octree-based adaptive semi-Lagrangian VOF approach for simulating the displacement of free surfaces”, Laurmaa et al., 2016.

Nesta direção, Wilhelms e Gelder (1992) propõem a integração do Marching Cubes à estrutura de dados Octree de forma a permitir que, através de um pré-processamento do dado de entrada, durante a sua computação, o algoritmo acessasse apenas as células da grade intersectadas pela isosuperfície de interesse, reduzindo assim o seu custo computacional.

Uma octree (representada na Figura 18) é uma estrutura de dados hierárquica em árvore, na qual cada nó interno possui exatamente oito filhos. Devido à sua alta capacidade de representar dados volumétricos, a octree permite que a navegação por suas ramificações

corresponda diretamente à exploração do dado volumétrico, proporcionando uma organização eficiente e adaptativa das informações.

Através de um pré-processamento do dado, assumimos o dado volumétrico como um nó raiz (com pequenas adaptações, nos casos onde as dimensões do dado não são potências de dois), então temos um grande cubo, cujos vértices são os 8 vértices das bordas do dado volumétrico. Subdividindo recursivamente um nó em oito octantes, é possível mapear na estrutura criada o dado volumétrico, sendo as folhas do último nível criado, os cubos contidos na grade que representa o dado original.

Dessa forma, através do dado volumétrico, obtemos uma árvore (estrutura de dados), onde cada um de seus nós é marcado com os valores mínimo e máximo dos escalares amostrados nos vértices dos cubos nele contido. Esses marcadores e a natureza hierárquica da octree permitem que o Marching Cubes navegue na estrutura criada (ao invés de marchar por todos os cubos da grade de entrada), percorrendo a árvore e acessando apenas as folhas (cubos da grade) para as quais o isovalor de interesse esteja entre os valores mínimo e máximo armazenados.

7. Conclusão

Este artigo destacou a relevância da interpretação precisa de dados volumétricos, enfatizando como os avanços no algoritmo Marching Cubes contribuem para aplicações práticas em áreas como medicina, geologia e simulações interativas. Em especial, a capacidade de reconstrução detalhada de superfícies a partir de dados amostrados tem um papel crucial na visualização e análise de estruturas complexas, como órgãos internos em exames de imagem médica e formações geológicas em modelagens tridimensionais. Nessa direção, apresentamos as melhorias propostas ao algoritmo Marching Cubes na busca de se obter um algoritmo capaz de representar corretamente as informações contidas no dado. Em adicional, apresentamos a integração do Marching Cubes à estrutura de dados Octree como meio de aumentar a eficiência do algoritmo através de um pré-processamento do dado de entrada. Tal integração permite que a marcha do algoritmo por todos os cubos da grade de entrada seja substituída pela navegação através da Octree acessando apenas os cubos intersectados pela isosuperfície de interesse.

Referências

Akinci G, Ihmsen M, Akinci N, Teschner M, Parallel Surface Reconstruction for Particle-Based Fluids. **Comput. Graph.** Forum 31(6): 1797-1809 (2012).

Ayachit, U. : The ParaView Guide: A Parallel Visualization Application, **Kitware Inc**, 2015, ISBN 9781930934306

Chernyaev, E.V.: Marching Cubes 33: Construction of Topologically Correct Isosurfaces, **Institute for High Energy Physics**, Moscow, Russia, Report CN/95-17, 1995.

Cirne M, Pedrini H, Acceleration of the Marching Cubes Technique for Volumetric Visualization on Graphics Processing Unit Using Spatial Data Structures, **Proceedings of the 17th Brazilian Symposium on Multimedia and the Web**, 2011, ISSN 2175-9642.

Custodio, L.; Etienne, T.; Pesco, S. ; Silva, C. Practical considerations on marching cubes 33 topological correctness, **Computer & Graphics**, v.37, n.7, p. 840 – 850, 2013.

Custodio, L.; Pesco, S.; Silva, C. An extended triangulation to the Marching Cubes 33 algorithm. **Journal of the Brazilian Computer Society**. 2019, v. 25, n. 6.

Du, S., Kanai, T.: GPU-based Adaptive Surface Reconstruction for Real-time SPH Fluids. In: **WSCG proceedings**, pp. 141–150 (2014).

Dürst, M.J.: Re: additional reference to Marching Cubes. **SIG-GRAPH Comput. Graph.** 22(5), 243 (1988).

Etienne T, Nonato L, Scheidegger C, Tienry J, Peter T, Pascucci V., "Topology verification for isosurface extraction", **IEEE Transactions on Visualization and Computer Graphics**, vol. 18, no. 6, pp. 952-965, June 2012.

EMC33. Extended Marching Cubes 33. Disponível em:

<https://liscustodio.github.io/ExtendedMC33/Figure7.html>. Acessado em: 14 de janeiro de 2025.

Grosso, R. Construction of topologically correct and manifold isosurfaces. **Computer Graphics Forum**, 2016, 35, 187–196.

Laine S. e Karras T. **Efficient Sparse Voxel Octrees - Analysis, Extensions, and Implementation**. Technical report, NVIDIA, 2010.

Lewiner, T., Lopes, H., Vieira, A.W., e Tavares, G., 2003. Efficient implementation of Marching Cubes' cases with topological guarantees, **Journal of Graphics Tools**, 8 (2) (2003), pp. 1–15.

Lopes, Adriano e Brodle, Ken., 2003. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. **IEEE Transactions on Visualization and Computer Graphics** 9, 1 (2003), 16–29.

Lorensen, William E. e Cline, Harvey E. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In **Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)**. Association for Computing Machinery, New York, NY, USA, 163–169.

Matveyev, S.V.: Marching cubes: surface complexity measure. In: **Proceedings of SPIE—The International Society for Optical Engineering**, vol. 3643, pp. 220–225 (1999).

Moreira, Marco Antônio e Nardi, Roberto. O mestrado profissional na área de Ensino de Ciências e Matemática: alguns esclarecimentos. **Revista Brasileira de Ensino de Ciências e Tecnologia**, v. 2, n. 3, p. 1-9, 2009.

Natarajan, B.K.: On generating topologically consistent isosurfaces from uniform samples. **The Visual Computer**, 11(1), 52–62 (1994).

Nielson, G. M. e Hamann, B.: The asymptotic decider: Resolving the ambiguity in marching cubes. **eScholarship**, University of California, 1991.

Nielson, G. M. : On marching cubes, **IEEE Transactions on Visualization and Computer Graphics**, vol. 9, no. 3, pp. 283-297, July-Sept. 2003, doi:10.1109/TVCG.2003.1207437.

NVIDIA. NVIDIA Corporation, 2023. Disponível em: <https://developer.nvidia.com/gpugems/gpugems3/part-i-geometry/chapter-1-generating-complex-procedural-terrains-using-gpu> . Acessado em: 26 out. 2023.

Paiva, A. C. ; Seixas, R.B. ; Gattass, M . **Introdução a Visualização Volumétrica**. Rio de Janeiro- RJ: Departamento de Informática - PUC/Rio, 1999 (Relatório Técnico).

Schroeder, Will; Martin, Ken; Lorensen , Bill (2006), The Visualization Toolkit (4th ed.), **Kitware**, ISBN 978-1-930934-19-1.

Seng, Dewen; Li, Zhongxue; Li, Cuiping; Li, Chunmin: Application of marching cubes algorithm in visualization of mineral deposits, **Journal of University of Science and Technology Beijing**, March 2005, pp. 203-207.

Solenthaler, Barbara; Schläfli, Jürg; Pajarola, Renato: A unified particle model for fluid-solid interactions. **Journal of Visualization and Computer Animation** 18(1): 69-82 (2007)

Surfer, versão 26.1.216 **Golden Software**, 2023. (informações em <http://www.goldensoftware.com>).

Tiago Etienne, Luis Gustavo Nonato, Carlos Scheidegger, Julien Tienry, Thomas J. Peters, Valerio Pascucci, Robert M. Kirby, and Claudio T. Silva. 2012. Topology Verification for Isosurface Extraction. **IEEE Transactions on Visualization and Computer Graphics** 18, 6 (June 2012), 952–965. <https://doi.org/10.1109/TVCG.2011.109>

Wang W, Jiang Z, Qiu H, And Li W. Real-time simulation of fluid scenes by smoothed particle hydrodynamics and marching cubes, **Mathematical Problems in Engineering**, vol. 2012, Article ID 635631, 9 pages, 2012.

WILHELMS J, GELDER A. Octrees for faster isosurface generation. **ACM Transactions on Graphics**, 11(3):201–227, July 1992.

Yu, Jihun; Turk, Greg : Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernels. **Symposium on Computer Animation** (2010):217-225

Zhu, Yongning; Bridson, Robert: Animating sand as a fluid. **ACM Trans. Graph.** 24(3): 965-972 (2005).