

## 4 Marching Cubes

### 4.1 Introdução

Uma iso-superfície é uma superfície representando todas as posições no espaço tridimensional em que o campo escalar  $f(x, y, z)$  assume um certo valor  $v$ , ou seja,  $f = v$ . O volume 3D logo se divide em duas regiões distintas: posições "dentro" da iso-superfície, assumindo valores maiores ou iguais a  $v$  e posições "fora" da iso-superfície, com valores menores que  $v$  (Figura 9).

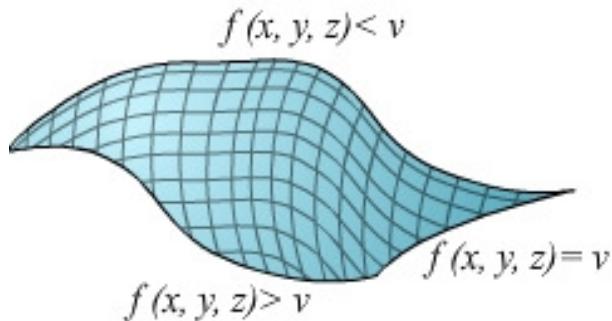


Figura 9 - Representação de uma iso-superfície

O algoritmo de Marching Cubes (Lorensen e Cline, 1987) tem se tornado uma referência quando o campo escalar amostrado é escalado numa matriz tridimensional. O método utiliza uma aproximação trilinear para localizar a superfície correspondente ao valor especificado, gerar a malha de triângulos e as normais dos vértices de cada triângulo.

### 4.2 O Algoritmo

O algoritmo pode ser explicado analisando a seção seguinte sobre o método *Marching Squares*, que é o análogo em duas dimensões.

#### 4.2.1 Marching Squares

Dado um grid bidimensional, cada vértice possui um valor peso  $W$ . O método traça linhas conectando arestas dos quadrados deste grid de acordo com um valor de referência  $N$ , também chamado de "iso-valor". A "iso-linha" é formada utilizando uma interpolação bilinear entre as linhas traçadas no passo anterior (Figura 10).

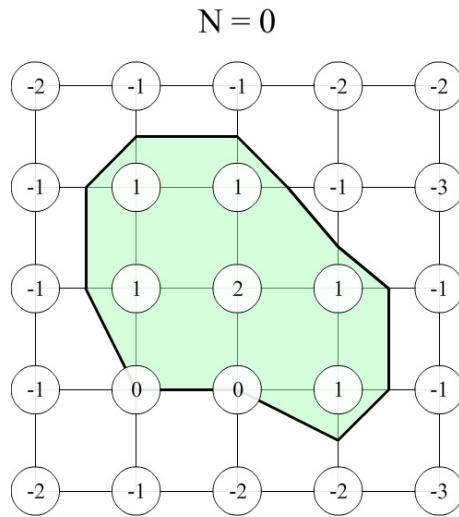


Figura 10 - Exemplo da iso-linha gerada com o algoritmo Marching Squares com  $N=0$ . Os números em cada vértice representam o seu peso  $W$

As iso-linhas são traçadas de acordo com uma tabela, denominada *lookup table* (Figura 11). Ela representa todas as configurações possíveis para representar as linhas que irão fazer parte do contorno.

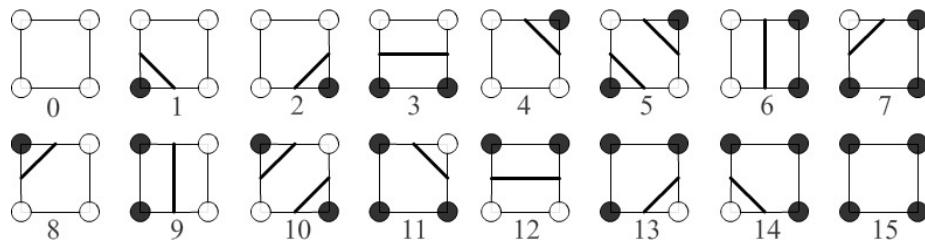


Figura 11 - Lookup table do algoritmo Marching Squares. Pontos pretos representam valores positivos e brancos valores negativos

A tabela da Figura 10 apresenta ambiguidades. Dentre elas, os casos 5 e 10 podem representar duas configurações diferentes (Figura 12). As ambiguidades foram abordadas no método Marching Cubes por Nielson e Hamann (1991) e aprimoradas em trabalhos posteriores (Chernyaev, 1995 e Lewiner *et al*, 2003).

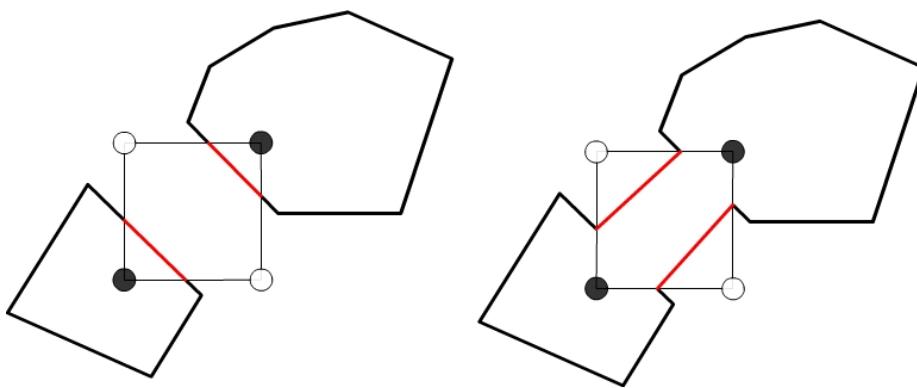


Figura 12 - Ambiguidade apresentada pela lookup table do algoritmo Marching Squares

Na seção seguinte, será mostrado o algoritmo de Marching Cubes, que é a expansão do método Marching Squares em três dimensões.

#### 4.2.2 Marching Cubes

O método aproxima a superfície dividindo o espaço de domínio em uma série de cubos lógicos, sendo que cada 4 vértices fazendo parte de um plano de corte (Figura 13).

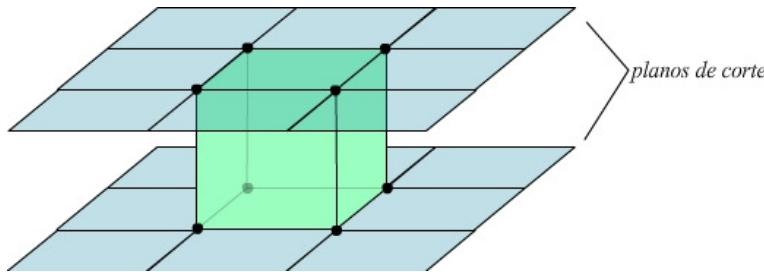


Figura 13 - Planos de corte de um cubo lógico utilizado no algoritmo de Marching Cubes

Segundo o trabalho de Lorensen (1987), o algoritmo realiza a verificação da interseção da superfície com todos os cubos do domínio. Para encontrar esta interseção é dado um valor "1" para um vértice caso seu valor seja maior ou igual ao valor da função da superfície a ser construída. Estes vértices são considerados como estando "dentro" da superfície". Já vértices cujos valores são menores que o valor da função da superfície recebem "0" e são considerados como parte externa desta superfície. A superfície intercepta as arestas do cubo que possuem uma aresta com o valor "1" e outra com valor "0" (Figura 14).

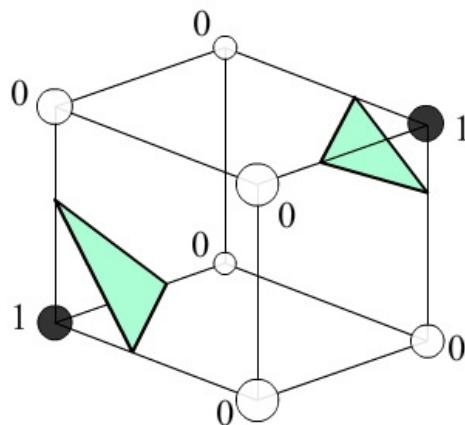


Figura 14 - Construção da superfície através de vértices do cubo

Como há 8 vértices em cada cubo, sendo que cada vértice pode estar dentro ou fora da superfície esperada, existem  $2^8 = 256$  maneiras diferentes dessa superfície interceptar as arestas deste cubo. Logo, foi construída uma *lookup table* para realizar a identificação de cada caso. Em seu trabalho, Lorensen (1987) identificou 15 padrões dos 256 casos possíveis devido à simetria do cubo, facilitando, assim, a codificação (Figura 15).

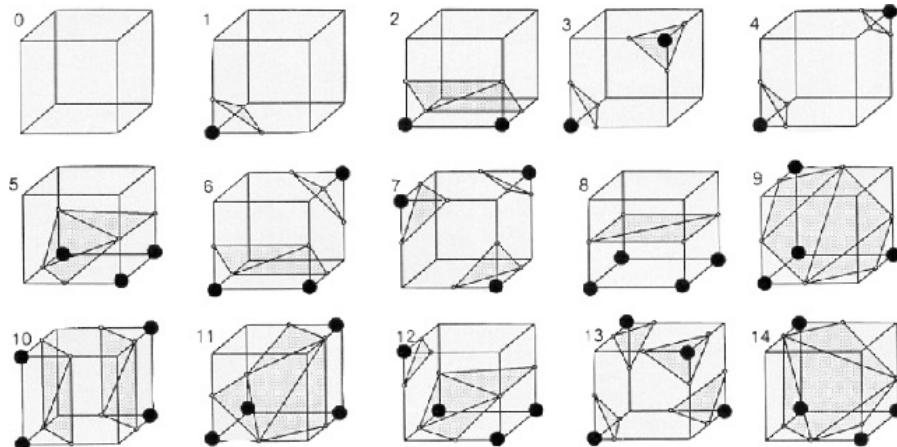


Figura 15 - Padrões estabelecidos por Lorensen (1987) para formar a lookup table

Apesar dos 256 índices possíveis, a mesma configuração da *lookup table* anterior pode gerar ambiguidades. Logo, imperfeições topológicas, tanto nas interfaces quanto dentro do próprio cubo lógico, podem ocorrer em alguns casos.

Para evitar a ambiguidade na interface dos cubos, Nielson e Hamann (1991) aplicaram testes na face de um cubo para checar a transição coerente entre um cubo e outro. Apesar da resolução dessa ambiguidade, não há garantias de uma topologia correta da malha a ser gerada. Isso ocorre porque

ainda há maneiras diferentes de triangularizar o cubo utilizando a mesma configuração dependendo da interpolação trilinear utilizada.

Chernyaev (1995), estendendo a lookup table de Lorensen (1987), resolveu as ambiguidades internas. Porém, apenas com o trabalho de Lewiner *et al* (2003) é que foi possível realizar uma consistência topológica, completando os trabalhos anteriores. Esta implementação foi utilizada nessa dissertação.

A última parte do algoritmo diz respeito ao cálculo do vetor unitário da normal de cada vértice dos triângulos gerados. Lorensen (1987) concluiu que o vetor normal em cada ponto do *grid* é exatamente o gradiente do campo escalar  $f$  neste mesmo ponto. Logo, após este cálculo, é feita uma interpolação trilinear para definir o vetor normal para cada vértice de triângulo dentro do cubo.

Dado os vértices de cada triângulo e o vetor normal de cada vértice, é possível renderizar a malha de triângulos que representa a iso-superfície estimada (Figura 16).

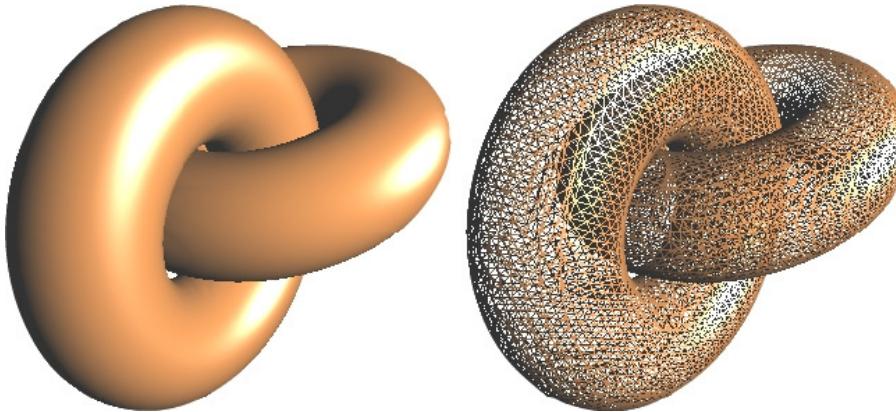


Figura 16 - Imagens de dois torus geradas a partir do algoritmo de Lewiner (2003). Na esquerda: a superfície construída. Na direita: a representação da malha