A
O Método Fast Marching e a Transformada de Distância

Este apêndice descreve um algoritmo eficiente, desenvolvido em [46], que pode ser utilizado para o cálculo da transformada de distância de objetos volumétricos.

A.1
Definição

A transformada de distância $\mathcal{T}$, aplicada a um objeto gráfico $\mathcal{O}$, calcula um campo escalar (ou vetorial) que representa distâncias mínimas entre o objeto e os pontos do espaço no qual ele está envolvido. A transformada $\mathcal{T}$ pode ser definida da seguinte maneira:

$$\mathcal{T}_{\mathcal{O}}(p) = min_{p_i \in \mathcal{O}} \quad dist(p, p_i), \tag{A-1}$$

onde p representa pontos arbitrários do espaço, e $dist$ representa uma função distância ou métrica utilizada. Assim, para cada ponto p do espaço, a transformada calcula a distância de p ao ponto $p_i \in \mathcal{O}$ que está mais próximo de p. Algumas métricas bastante utilizadas para o cálculo de transformadas de distância são a métrica euclidiana, a métrica *cityblock* e a métrica *chessboard* [53]. Vários algoritmos procuram calcular a transformada de distância de objetos volumétricos (ou matriciais em geral) a partir da *codificação de voxels*, onde um conjunto inicial de voxels que intercepta uma iso-superfície d é iniciado com um valor (em geral com zero). Em seguida, a partir deste conjunto inicial, a informação vai sendo propagada aos vértices vizinhos até que todos os voxels do volume tenham sido percorridos.

A.2
Propagação de Interfaces

A heurística da codificação de voxels pode ser comparada ao paradigma da evolução de uma frente que avança sobre um determinado

PUC-Rio - Certificação Digital Nº 9816117/CA

meio, como é o caso de uma frente em chamas que avança se propagando sobre uma região coberta de gramas. Esta comparação pode ser bastante útil, no sentido de que o cálculo da transformada de distância pode recorrer à Teoria da Evolução de Interfaces (a frente em chamas se propagando é uma interface), numa tentativa de encontrar novas soluções para suas aplicações. Sethian [35, 46] desenvolveu um algoritmo baseado na conservação das Leis Hiperbólicas [1] para calcular eficientemente a propagação de uma interface sobre outra. Este algoritmo pode ser utilizado para o cálculo da transformada de distância.

Uma interface pode ser geometricamente considerada uma curva ou uma superfície que separa dois meios que estão interagindo entre si. Ou seja, a interface diz respeito à borda ou fronteira que separa os dois meios. Suponha que a interface esteja se movendo em direção a sua normal, com uma dada velocidade F. A figura A.1 mostra uma interface separando dois meios.

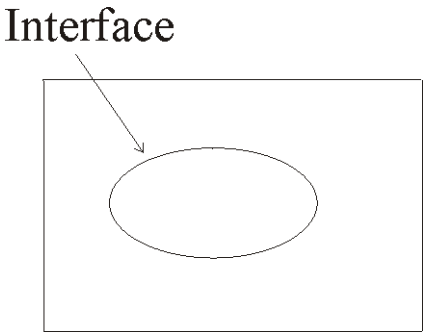

Figura A.1: Interface separando dois meios.

De um modo geral, a velocidade de propagação F pode depender de vários fatores:

- Propriedades Locais - são aquelas que dependem da geometria local à curva, como curvatura, vetor normal, etc.
- Propriedades Globais - são aquelas que dependem da forma, posição e características específicas de uma determinada interface.
- Propriedades Independentes - são aquelas que dependem do posicionamento da interface, como por exemplo, um fluido no qual a interface está sendo conduzida.

A solução para o problema de evolução de interface pode ser formulada de várias maneiras. Uma maneira utilizada em [46] é conhecida como *formulação do valor de borda*, que pode ser colocada da seguinte forma:

PUC-Rio - Certificação Digital Nº 9816117/CA

partindo de uma curva ou interface inicial (instante inicial zero), a cada instante T a interface vai evoluindo, ocupando uma nova posição no espaço (figura A.2 à esquerda), ou seja, há um tempo T associado a cada nova interface resultante da evolução. Com isto cada curva pode ser vista como uma curva de nível de uma função tempo T (figura A.2 à direita). A função T é associada ao tempo no sentido de que a interface em evolução passa por cada ponto apenas uma vez, ou seja, há apenas um único valor T associado a cada ponto. Em outras palavras, a velocidade de propagação é sempre positiva, $F > 0$ (ou sempre negativa).

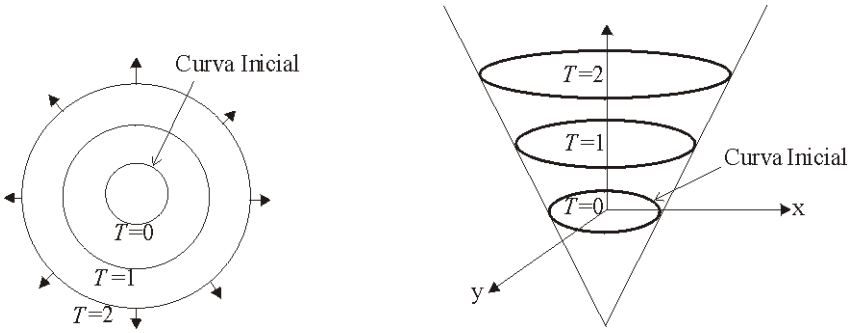

Figura A.2: Formulação do valor de borda.

A motivação do nome formulação do valor de borda surge do fato de que, a cada instante T que se deseja saber onde a interface se encontra, basta tomar a borda da superfície $T(x, y)$ na altura T, conforme mostra a figura A.2 da direita. Esta formulação é utilizada para definir um método de propagação chamado de *Fast Marching*.

A.3 Método Fast Marching

O Método Fast Marching propaga a interface baseando-se na formulação do valor de borda, onde uma função tempo T associa a cada ponto do espaço ao instante em que a interface atinge este ponto (x, y). Esta formulação é utilizada apenas no caso onde a velocidade é sempre positiva (ou sempre negativa). Quando a interface se propaga com velocidade constante esta formulação pode ser utilizada para calcular campos de distância. Neste caso, a função T calculada em cada ponto representa a distância daquele ponto à interface inicial.

A idéia do método Fast Marching é que, partindo de uma interface inicial, discretizada sobre uma grade, a função T vai sendo construída sobre os pontos da grade, à medida que a interface se propaga. A figura A.3 mostra

PUC-Rio - Certificação Digital Nº 9816117/CA

uma curva sobre uma grade 2D, onde nos pontos (i, j) que correspondem à interface inicial, $T_{ij} = 0$ (início da propagação). Nos demais pontos, externos à interface, o valor de T não é conhecido. O objetivo do método é justamente calcular estes valores de T, de maneira eficiente, à medida que a curva evolui. Em cada iteração da propagação é construída uma nova camada da superfície T, que corresponde ao avanço ou propagação da frente.

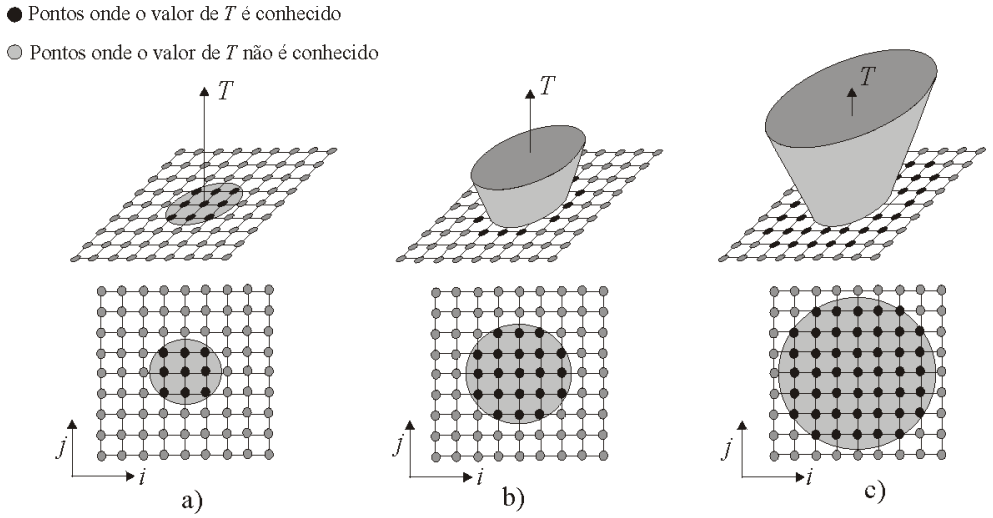

Figura A.3: Construção da função T no método Fast Marching.

O Método Fast Marching é formulado a partir da *equação Eikonal* $\|T\|F = 1$. A solução numérica desta equação é baseada nas Leis da Conservação Hiperbólica e pode ser resolvida através do esquema:

$$\begin{bmatrix} \max(D_{ijk}^{-x}, 0)^2 + \min(D_{ijk}^{+x}, 0)^2 + \\ \max(D_{ijk}^{-y}, 0)^2 + \min(D_{ijk}^{+y}, 0)^2 + \\ \max(D_{ijk}^{-z}, 0)^2 + \min(D_{ijk}^{+z}, 0)^2 \end{bmatrix} = \frac{1}{F_{ijk}}$$

onde $D_{ijk}^{-x} = (T_{ijk} - T_{i-1,j,k})/\Delta x$ e $D_{ijk}^{+x} = (T_{i+1,j,k} - T_{ijk})/\Delta x$. $D_{i,j,k}^{-y}, D_{ijk}^{+y}, D_{ijk}^{-z}$ e D_{ijk}^{+z} são definidos de maneira análoga.

A forma padrão de se resolver estas equações requer iterações. A cada iteração *iter* os valores T_{ijk}^{iter}, no ponto (i, j, k), vão sendo calculados, a partir dos valores T_{ijk} dos vizinhos da iteração anterior. O algoritmo abaixo descreve esta idéia:

Durante o cálculo dos valores de T, a informação vai sempre se propagando a partir dos pontos com menores valores de T. A figura A.4 explica como o processo de propagação de uma interface 2D se dá a cada iteração. Os pontos pretos representam as posições onde a função T é conhecida e os pontos brancos, posições onde T é desconhecida. Partindo de uma interface inicial, representada por um ponto preto, onde $T = 0$ (figura

PUC-Rio - Certificação Digital Nº 9816117/CA

Algorithm 25 FastMarching

Para $iter = 1, n$
  Para $i, j, k = 1, dim$
    Resolver a equação para T_{ijk}^{iter+1}, a partir de $T_{i-1,j,k}^{iter}, T_{i+1,j,k}^{iter}, T_{i,j-1,k}^{iter}, T_{i,j+1,k}^{iter}, T_{i,j,k-1}^{iter}, T_{i,j,k+1}^{iter}$
  FimPara
FimPara

A.4a), são calculados os valores de seus quatro vizinhos (representados pelos pontos cinzas A, B, C e D, na figura A.4b), através da equação Eikonal discreta. Dentre estes quatro pontos, a propagação deve seguir a partir daquele que tiver o menor valor de T. Ou seja o algoritmo para propagação requer que haja uma ordenação dos pontos cinzas. Supondo que o ponto A contém o menor T, a propagação deve prosseguir a partir dele (figura A.4c), ou seja, o ponto A é setado para preto, indicando que a propagação já é conhecida em A, e seus vizinhos são calculados (representados pelos pontos cinzas E, F e G, na figura A.4d). É importante observar que o ponto preto inicial, apesar de ser vizinho de A, não entrou neste processo, pois a propagação tem sentido único, não é retroativa. A propagação deve continuar a partir do ponto cinza (B, C, D, E, F ou G) que tiver o menor valor de T. Mais uma vez será necessária a rotina de ordenação para selecionar o ponto cinza de menor T. Supondo que o ponto D contém o menor valor, a propagação deve seguir a partir dele (figura A.4e). O valor de T é calculado nos vizinhos de D, que são setados para cinza. É importante observar que um dos vizinhos de D (o ponto E) já era cinza, pois também é vizinho de A, mas seu valor deve ser recalculado (apenas os vizinhos pretos são poupados). O processo se repete até que a função T seja determinada.

Utilizando a equação Eikonal, o cálculo de T nos vizinhos nunca fornece valores menores do que os pontos já conhecidos (pontos pretos), e portanto, a "marcha"segue sempre em um único sentido, se afastando da origem. Para a propagação nos pontos da grade, há três categorias de pontos: pontos com valores conhecidos (pretos), pontos candidatos a prosseguirem com a propagação (cinzas) e pontos desconhecidos (brancos), que seriam os pontos "distantes"da interface. Na busca da solução, a marcha se dá sempre transformando pontos brancos em pontos cinzas e cinzas em pretos, conforme mostra a figura A.5.

Uma das grandes vantagens deste método, além da eficiência, é que a mesma abordagem é utilizada tanto do ponto de vista discreto quanto contínuo. A discretização do objeto pode ser feita em qualquer resolução,

PUC-Rio - Certificação Digital Nº 9816117/CA

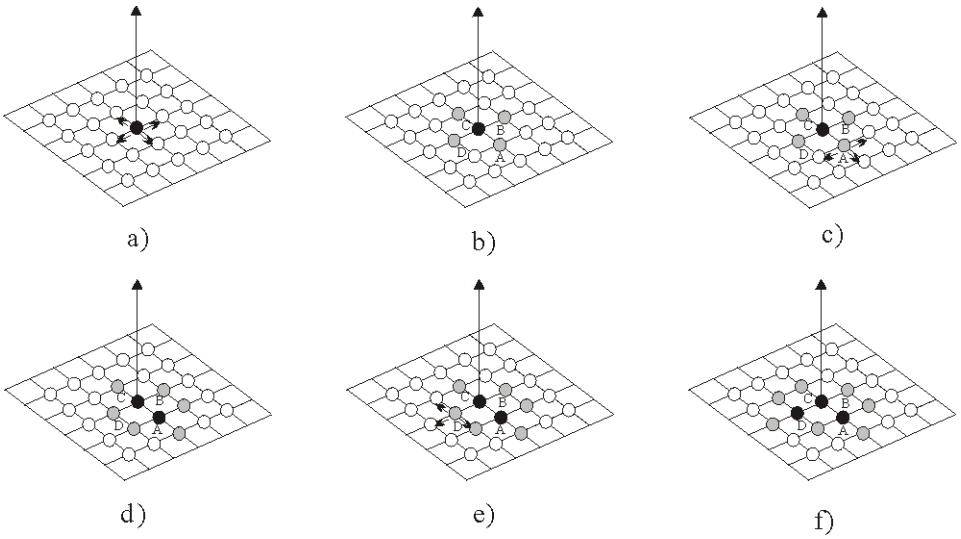

Figura A.4: Propagação dos valores de T nos pontos (i, j).

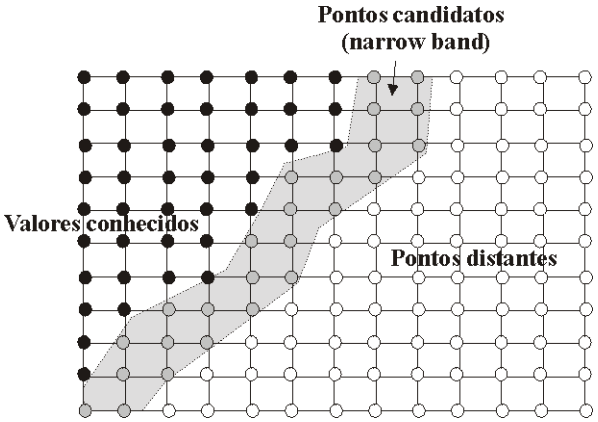

Figura A.5: Classificação dos pontos.

o que permite que possa ser controlada a margem de erro, nos casos que requerem reconstrução do objeto.

PUC-Rio - Certificação Digital Nº 9816117/CA