

## Bibliografia

- [1] LAX, P. D.. **Hyperbolic systems for conservation laws and mathematical theory of shock waves.** SIAM Reg. Conf. Series, Lectures in Applied Math, (11):1–47, 1970.
- [2] CARMO, M. P. D.. **Elementos de geometria diferencial.** Ao Livro Técnico, 1971.
- [3] KEPPEL, E.. **Approximating complex surfaces by triangulation of contour lines.** IBM Journal of Research and Development, 19:2–11, 1975.
- [4] CARMO, M. P. D.. **Differential geometry of curves and surfaces.** Englewood Cliffs, N.J. : Prentice-Hall, 1976.
- [5] RUDIN, W.. **Principles of Mathematical Analysis.** McGraw-Hill, New York, 1976.
- [6] FUCHS, H.; M., K. Z. ; USELTON, S. P.. **Optimal surface reconstruction from planar contours.** Communications of ACM, 20:693–702, 1977.
- [7] CHRISTIANSEN, H. N.; SEDERBERG, T. W.. **Conversion of complex contour line definitions into polygonal element mosaics.** Computer Graphics, 12:187–192, 1978.
- [8] BATNITZKY, S.; PRICE, H. I.; COOK, P. N.; COOK, L. T. ; DWYER, S. J.. **Three-dimensional computer reconstruction from surface contours for head ct examinations.** Journal of Computer Assisted Tomography, 5:60–67, 1981.
- [9] SOROKA, B. I.. **Generalized cones from serial sections.** Computer Graphics and Image Processing, 15(2):154–166, 1981.
- [10] WANG, Y. F.; AGGARWAL, J. K.. **Construction of surface from 3d volumetric scene description.** Pattern Recognition, 19(3):197–207, 1986.

- [11] LEVIN, D.. **Multidimensional reconstruction by set-valued approximation.** IMA Journal of Numerical Analysis, 6:173–184, 1986.
- [12] LORENSEN, W. E.; CLINE, H. E.. **Marching cubes: A high resolution 3d surface construction algorithm.** Computer Graphics (Proceedings of SIGGRAPH '87), 21(4):163–169, July 1987.
- [13] LIMA, E. L.. **Curso de Análise vol. 2.** Instituto de Matemática Pura e Aplicada, 1987.
- [14] BOISSONNAT, J. D.. **Shape reconstruction from planar cross sections.** Computer Vision, Graphics and Image Processing, 44:1–29, 1988.
- [15] LEVOY, M.. **Efficient ray tracing of volume data.** ACM Transactions on Graphics, 9(3):245–261, July 1990.
- [16] EKOULE, A. B.; PEYRIN, F. C. ; ODET, C. L.. **A triangulation algorithm from arbitrary shaped multiple planar contours.** ACM Trans. on Graph, 10(2):182–199, 1991.
- [17] SHINAGAWA, Y.; KUNNI, T. L.. **The homotopy model: a generalized model for smooth surface generation from cross sectional data.** The Visual Computer, 7:72–86, 1991.
- [18] GOMES, J.; VELHO, L.. **Implicit Objects in Computer Graphics.** Instituto de Matemática Pura e Aplicada, 1992.
- [19] WILHELMS, J.; GELDER, A. V.. **Octrees for faster isosurface generation.** ACM Trans. on Graphics,, 11(3):201–227, July 1992.
- [20] MEYERS, D.; SKINNER, S. ; CLINE, H. E.. **Surfaces from contours.** ACM Trans. on Graphics, 11(3):228–258, 1992.
- [21] TURK, G.. **Re-tiling polygonal surfaces.** Computer Graphics, SIGGRAPH, 26(2):55–64, July 1992.
- [22] GEIGER, B.. **Three dimensional modeling of human organs and its applications to diagnosis and surgical planning.** Technical Report 2105, INRIA, France, 1993.
- [23] ROSSIGNAC, J.; BORREL, P.. **Multiresolution 3d approximations for rendering complex scenes.** Modeling in Computer Graphics: Methods and Applications, 1993.

- [24] GOMES, J.; COSTA, B.; DARSA, L. ; VELHO, L.. **Graphical Objects**. Instituto de Matemática Pura e Aplicada, 1994.
- [25] CIGNONI, P.; FLORIANI, L. D.; MONTINI, C.; PUPP, E. ; SCOPIGNO, R.. **Multiresolution modeling and visualization of volume data based on simplicial complexes**. Symposium on Volume Visualization, ACM SIGGRAPH, p. 19–26, October 1994.
- [26] CORMEN, T. H.; LEISERSON, C. E. ; RIVEST, R. L.. **Introduction to Algorithms**. McGraw Hill, 1994.
- [27] ECK, M.; DEROSE, T.; DUCHAMP, T.; HOPPE, H.; LOUNSBERY, M. ; STUERTZLE, W.. **Multiresolution analysis of arbitrary meshes**. Proceedings of SIGGRAPH '95, p. 173–182, 1995.
- [28] GOMES, J.; VELHO, L.. **Abstraction paradigms for computer graphics**. The Visual Computer, 5(5):227–239, 1995.
- [29] BAJAJ, C.; COYLE, E. ; LINK, K.. **Arbitrary topology shape reconstruction from planar cross sections**. Graph. Models Image Process, 58(6):524–543, 1996.
- [30] BAREQUET, G.; SHARIR, M.. **Piecewise-linear interpolation between polygonal slices**. Computer Vision and Image Understanding, 6(2):251–272, 1996.
- [31] OLIVA, J. M.; PERRIN, M. ; COQUILLART, S.. **3d reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoi diagram**. Comp. Graph. Forum, 15(3):397–408, 1996.
- [32] GITLIN, C.; O´ROURKE, J. ; SUBRAMANIAN, V.. **On reconstructing polyhedral from parallel slices**. International Journal of Comp. Geometry and Applications, 6(1):103–122, 1996.
- [33] FLOATER, M.; WESTGAARD, G.. **Smooth surface reconstruction from cross-section using implicit methods**. Technical report, SINTEF, 1996.
- [34] VELHO, L.. **Simple and efficient polygonization of implicit surfaces**. Journal of Graphics Tools, 1(2):5–24, 1996.
- [35] SETHIAN, J. A.. **A fast marching level set method for monotonically advancing fronts**. National Academy of Sciences, 93(4):1591–1595, 1996.

- [36] ZHOU, Y.; CHEN, B. ; KAUFMAN, A.. **Multiresolution tetrahedral framework for visualizing regular volume data.** IEEE Visualization, 34:135–142, 1997.
- [37] GALIN, E.; AKKOUCHE, S.. **Fast surface reconstruction from contours using implicit surfaces.** Implicit Surface '98. Eurographics and ACM SIGGRAPH, p. 139–144, June 1998.
- [38] LEE, A. W. F.; SWELDENS, W.; SCHRÖDER, P.; COWSAR, L. ; DOBKIN, D.. **Maps: Multiresolution adaptive parameterization of surfaces.** Proceedings of SIGGRAPH '98, p. 95–104, 1998.
- [39] ARAÚJO, P. V.. **Geometria diferencial.** Coleção Matemática Universitária, 1998.
- [40] PUPPO, E.. **Variable resolution triangulations.** Computational Geometry Theory and Applications, 3-4(11):219–238, 1998.
- [41] NONATO, L. G.. **Reconstrução Volumétrica de Objetos Dados por Seções Bidimensionais.** PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 1998.
- [42] CONG, G.; PARVIN, B.. **An algebraic solution to surface recovery from cross-sectional contours.** Graphical Models and Image Processing, 61:222–243, 1999.
- [43] BAJAJ, C.; PASCUCCI, V.. **Progressive isocontouring.** Technical Report TR 99-36, University of Texas at Austin, 1999.
- [44] WESTERMANN, R.; KOBBELT, L. ; ERTL, T.. **Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces.** The Visual Computer, 15(2):100–111, 1999.
- [45] KOBBELT, L. P.; VORSATZ, J.; LABSIK, U. ; SEIDEL, H. P.. **A shrink wrapping approach to remeshing polygonal surfaces.** Computer Graphics Forum, 18(3):119–130, 1999.
- [46] SETHIAN, J. A.. **Level Set Methods and Fast Marching Methods.** Cambridge University Press, second edition, 1999.
- [47] WOOD, Z. J.. **Semi-regular mesh extraction from volumes.** Master's thesis, California Institute of Technology, 2000.
- [48] VELHO, L.; GOMES, J.. **Variable resolution 4-k meshes: Concepts and applications.** Computer Graphics Forum, 19(4):195–214, 2000.

- [49] PEIXOTO, A.; GATTASS, M.. **Reconstrução de superfícies a partir de seções bidimensionais**. Technical Report 28/00, Pontifícia Universidade Católica do Rio de Janeiro, julho 2000.
- [50] LEVOY, M.; OULLI, K.; CURLESS, B.; RUSINKIEWICZ, S.; KOLLER, D.; PEREIRA, L.; GINZTON, M.; ANDERSON, S.; DAVIS, J.; GINSBERG, J. ; SHADE, J. ANDFULK, D.. **Digital michelangelo project: 3d scanning of large statues**. SIGGRAPH, 2000.
- [51] KHODAKOVSKY, A.; SCHRÖDER, P. ; SWELDENS, W.. **Progressive geometry compression**. Proceedings of SIGGRAPH, p. 271–278, 2000.
- [52] VELHO, L.. **Semi-regular 4-8 refinement and box spline surfaces**. Proceedings of SIBGRAPI 2000, p. 131–138, October 2000.
- [53] PEIXOTO, A.; VELHO, L.. **Transformadas de distância**. Technical Report 35/00, Pontifícia Universidade Católica do Rio de Janeiro, setembro 2000.
- [54] GAVRILIU, M.; CARRANZA, J.; BREEN, D. ; BARR, A.. **Fast extraction of adaptive multiresolution meshes with guaranteed properties from volumetric data**. Proceedings of IEEE Visualization, p. 295–302, October 2001.
- [55] SHAFFER, E.; GARLAND, M.. **Efficient adaptive simplification of massive meshes**. IEEE Visualization, 2001.
- [56] LINSEN, P.. **Point cloud representation**. Technical report, Faculty of Computer Science, University of Karlsruhe, 2001.
- [57] OHTAKE, Y.; BELYAEV, A. ; PASKO, A.. **Dynamic meshes for accurate polygonization of implicit surfaces with sharp features**. Shape Modeling International, p. 74–81, 2001.
- [58] VELHO, L.. **4-8 subdivision**. Computer-Aided Geometric Design, 5(18):397–427, 2001.
- [59] PAULY, M.; GROSS, M. ; KOBELT, L. P.. **Efficient simplification of point-sampled surfaces**. IEEE Visualization 2002, 2002.
- [60] SÁ, A.; CARVALHO, P. C. P. ; VELHO, L.. **(b,s)-bcsI: Structured light color boundary coding for 3d photography**. 7th International Fall Workshop on Vision, Modeling and Visualization, 2002.

## 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:

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

onde  $p$  representa pontos arbitrários do espaço, e  $\text{dist}$  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

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.

### Interface

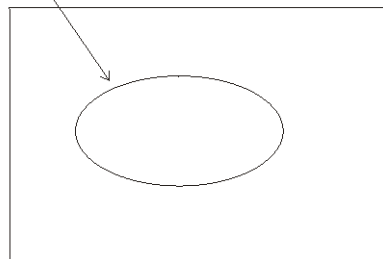


Figura A.1: Interface separando dois meios.

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:

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).

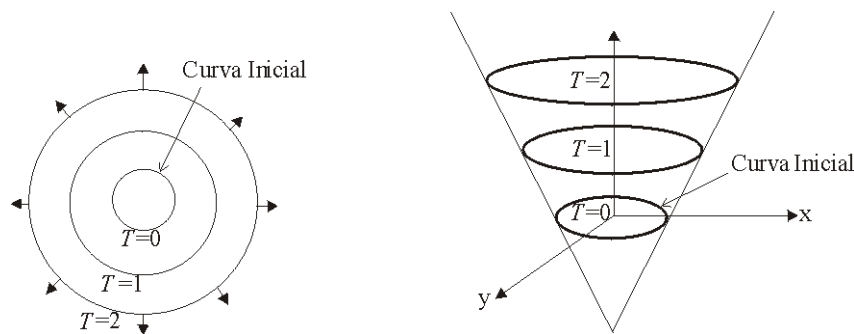


Figura A.2: Formulação do valor de borda.

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



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.

● Pontos onde o valor de  $T$  é conhecido

○ Pontos onde o valor de  $T$  não é conhecido

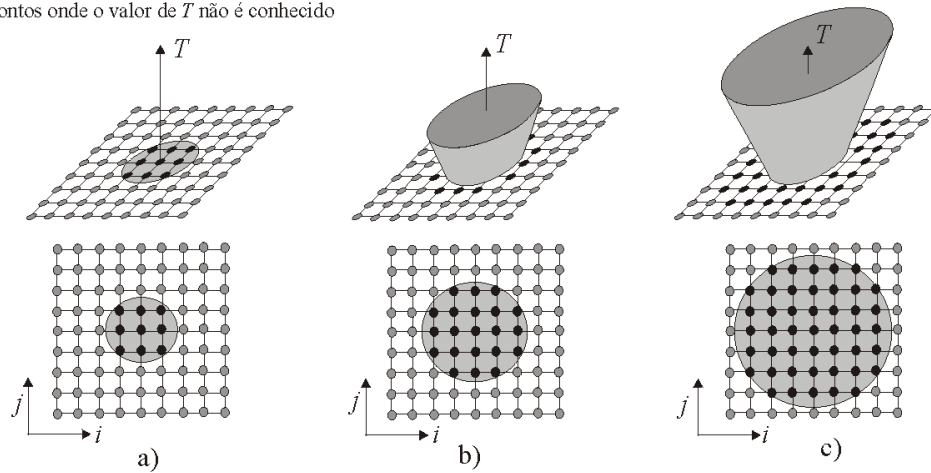


Figura A.3: Construção da função  $T$  no método Fast Marching.

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:

$$\left[ \begin{array}{l} \max(D_{ijk}^{-x}, 0)^2 + \min(D_{ijk}^{+x}, 0)^2 + \\ \max(D_{ijk}^{-y}, 0)^2 + \min(D_{ijk}^{+y}, 0)^2 + \\ \max(D_{ijk}^{-z}, 0)^2 + \min(D_{ijk}^{+z}, 0)^2 \end{array} \right] = \frac{1}{F_{ijk}}$$

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

---

**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,

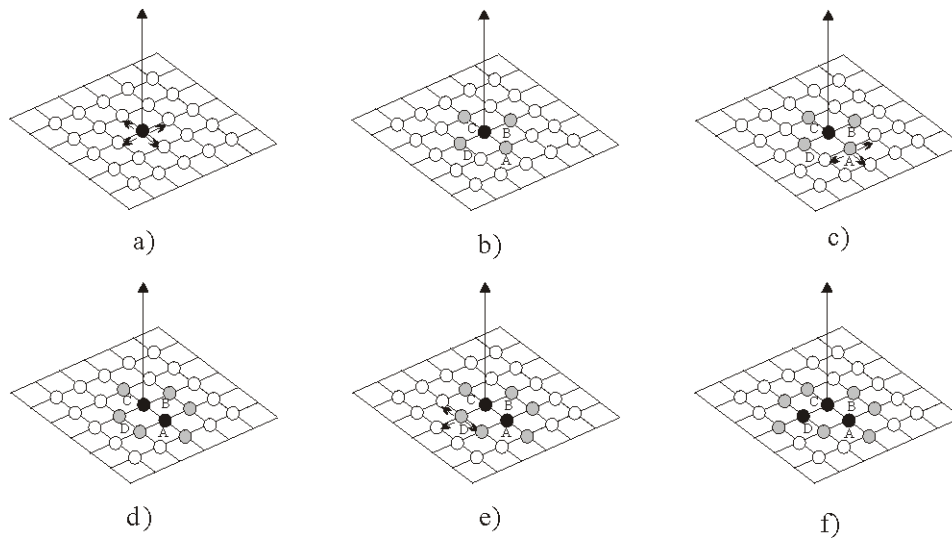


Figura A.4: Propagação dos valores de  $T$  nos pontos  $(i, j)$ .

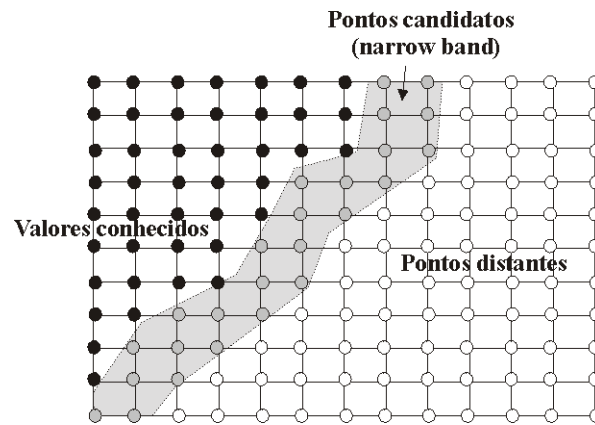


Figura A.5: Classificação dos pontos.

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