Fast Voronoi Diagrams on Triangulated Surfaces

R. Kimmel[||] **J.A. Sethian**[**]

We apply the Fast Marching Method on triangulated domains to efficiently compute Voronoi diagrams on triangulated manifolds. The computational complexity of the proposed algorithm is optimal, $O(n \lg n)$, where n is the number of vertices. The algorithm also applies to weighted domains in which a different cost is assigned to each surface point.

Voronoi diagrams play important roles in many research fields such as robotic navigation and control, image processing, computer graphics, computational geometry, pattern recognition, and computer vision. Its Euclidean version, for which there are efficient implementations, is a building block in many applications.

The Voronoi diagram sets boundaries between a given set of source points and splits the domain into regions such that each region corresponds to the closest neighborhood of a source point from the given set. Let our domain be D, let the set of given n points be $\{p_j \in D, j \in 0, .., n-1\}$, and the distance between two points $p, q \in D$ be $d(p, q)$, then the Voronoi region G_i corresponds to the set of points $p \in D$ such that $d(p, p_i) < d(p, p_j), \forall j \neq i$.

Efficient construction of distance maps, minimal geodesics, Voronoi diagrams, and offset curves for non-flat and weighted domains is a challenging problem, see e.g. [11, 10, 5, 9, 4, 7]. The core of our approach is Sethian's Fast Marching Method, [15, 12, 13] which solves the Eikonal equation on a rectangular orthogonal mesh in $O(n \lg n)$ steps, where n is the total number of grid points. Contingent upon the triangulated upwind and monotonic update schemes given by Barth and Sethian [1], this technique was extended to triangulated surfaces by Kimmel and Sethian in [8]. The triangulated version of the Fast Marching Method has the same computational complexity, and solves the Eikonal equation on triangulated domains in $O(n \lg n)$ steps, where n is the number of vertices. Using this technique, one can compute distances on curved manifolds with local weights.

We show how to apply our method to compute Voronoi diagrams of a given set of points (or regions) on triangulated manifolds. The proposed algorithm computational complexity is optimal $O(n \lg n)$, its implementation is simple, and it also applies to weighted domains in which a different cost is assigned to each surface point.

The key idea is based on upwind finite difference operators as numerically consistent approximation to the differential operators in the Eikonal equation. Such an approximation selects the correct viscosity solution. The upwind operators allow to construct a solution to the Eikonal equation by optimally sorting the updated points using a heap structure.

[||]Computer Science Department, Technion, Israel Institute of Technology, Haifa 32000

[**]Department of Mathematics University of California, Berkeley, CA 94720

Fast Marching Method and the Eikonal Equation: Given a 2D weighted flat domain, or in other words isotropic non-homogeneous domain, the distance may be defined via the arclength definition. E.g. the arclength may be written as a function of the x and y Cartesian coordinates of the planar domain $ds^2 = \mathcal{F}(x,y)^2(dx^2 + dy^2)$, where $\mathcal{F}(x,y) : R^2 \to R^+$ is a function that defines a weight for each point in the domain.

The distance map $T(x,y)$ from a given point p_0 assigns a scalar value to each point in the domain that corresponds to its distance from p_0. It is easy to show, see e.g. [2], that the gradient magnitude of the distance map is proportional to the weight function at each point $|\nabla T(x,y)| = \mathcal{F}(x,y)$, where $|\nabla T| \equiv \sqrt{T_x^2 + T_y^2}$. This equation is known as the *Eikonal equation*. The 'viscosity' solution to the Eikonal equation coupled with the boundary condition $T(p_0) = 0$ results in the desired distance map.

Our first goal is to solve the Eikonal equation. The key is to construct a numerical approximation to the gradient magnitude that selects an appropriate 'weak solution'. Consider the following upwind approximation to the gradient, given by

$$\left(\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2\right)^{1/2} = \mathcal{F}_{ij}, \tag{1}$$

where $D_{ij}^{-x}T \equiv \frac{T_{ij} - T_{i-1,j}}{h}$ is the standard backwards derivative approximation, and $T_{ij} \equiv T(i\Delta x, j\Delta y)$.

The solution T can be systematically constructed in an upwind fashion. The upwind difference approximation of Eq. (1) means that information propagates one way from smaller values of T to larger values. The Fast Marching Method exploits this order of events. A point gets updated only by points with smaller values. This 'monotone property' allows us to keep a front of candidate points, that tracks the flow of information, ordered in a heap tree structure in which the root is always the smallest value. An update of an element in the heap tree is done in $O(\lg n)$. Thereby, the total computational complexity is $O(n \lg n)$. We refer to [15, 12, 13] for further details on the Fast Marching Method.

One could recognize similarity to Dijkstra's method [3] that computes minimum costs of paths on networks. Dijkstra algorithm would obviously fail to consistently solve our geometric problems. Actually, any graph search based algorithm induces the artificial metric imposed by the graph network and would be inconsistent with the continuous case, and thus fail to converge as the graph resolution is refined.

The Fast Marching Method that works for orthogonal grids may be viewed as a selection for the update of one of the four right angle triangles that share the same vertex. The extension to triangulated domains is motivated by this observation, by the geometric interpretation of the update step, and by an additional special treatment of obtuse angles. We refer to [8] for details on the extension of the fast marching method to triangulated domains. It is also based on a finite difference approximation to the Eikonal equation, this time on the surface, monotone by construction, consistent, upwind, and converges to the viscosity solution.

Examples: Figure 1 presents Voronoi diagrams on several beads. We first compute the distance from each of the initial given source points simultaneously using a single heap structure, and allow one vertex overlap between distance maps form different sources. The complexity for the distance computation is still $O(n \lg n)$. Next, we 'march' along the triangles, and for each triangle linearly interpolate the intersection curve between the two different distance maps, again an $O(n)$ operation.

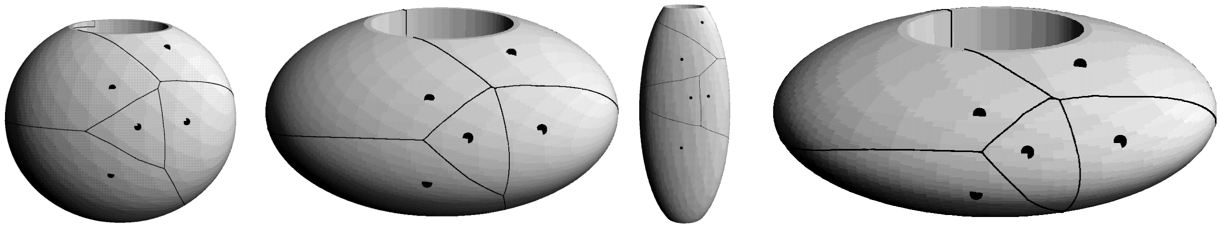

Figure 1: Voronoi diagrams of five points on four beads.

The algorithm complexity remains the same as we add weights to the surface, see Figure 2. In Figure 3 a different cost is assigned to each vertex. The cost, or weight function, is textured mapped onto the triangulated surface. Weighted geodesic Voronoi diagrams for several surfaces are presented in Figures 2,3. In both examples dark intensity mapped onto the surface indicates a low cost and the brighter the intensity the higher the cost.

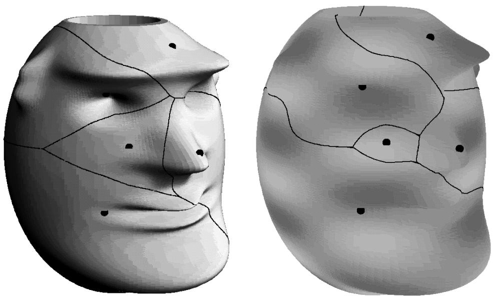

Figure 2: Voronoi diagrams of five points on a Synthetic Head with and without weights.

References

[1] T Barth and J A Sethian. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. in press, J. Comp. Phys., to appear, 1998.

[2] R Bellman and R Kalaba. *Dynamic Programming and modern control theory.* London mathematical society monographs, London, 1965.

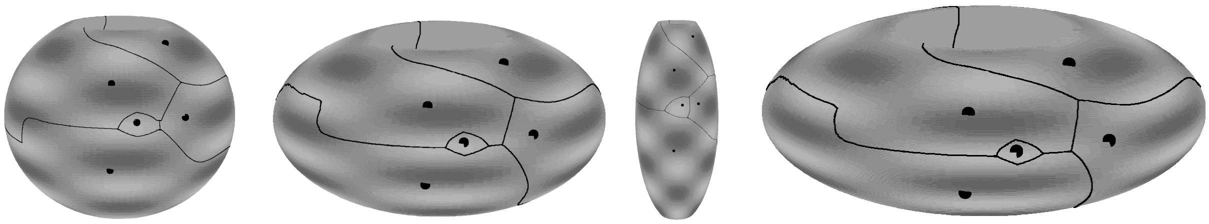

Figure 3: Weighted Voronoi diagrams of five points on four beads.

[3] E W Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematic*, 1:269–271, 1959.

[4] G Elber and E Cohen. Error bounded variable distance offset operator for free form curves and surfaces. *Int. J. Comput. Geom. & Applic.*, 1(1):67–78, 1991.

[5] R Kimmel, A Amir, and A M Bruckstein. Finding shortest paths on surfaces using level sets propagation. *IEEE Trans. on PAMI*, 17(6):635–640, June 1995.

[6] R Kimmel and A M Bruckstein. Shape offsets via level sets. *CAD*, 25(5):154–162, 1993.

[7] R Kimmel, N Kiryati, and A M Bruckstein. Sub-pixel distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision, Special Issue on Topology and Geometry in Computer Vision*, 6:223–233, June 1996.

[8] R Kimmel and J A Sethian. Computing geodesic paths on manifolds. *Proceedings of National Academy of Sciences, USA*, 95(15): 8431-8435, 1998.

[9] R Kunze, F E Wolter, and T Rausch. Geodesic Voronoi diagrams on parametric surfaces. Welfen lab. report, Welfen Lab., Univ. Hannover, Germany, July 1997.

[10] T Maekawa. Computation of shortest paths on free-form parametric surfaces. *Journal of Mechanical Design*, 118:499–508, 1996.

[11] J S B Mitchell, D M Mount, and C H Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, 1987.

[12] J A Sethian. A review of the theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, Cambridge University Press, 1996.

[13] J A Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge Univ. Press, 1996.

[14] J A Sethian. Curvature and the evolution of fronts *Comm. in Math. Phys.*, 101:487–499, 1985.

[15] J A Sethian. A marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.*, 93(4), 1996.