Centerline Extraction with Fast Marching Methods

Maksuda Lillah and Jeff Boisvert

Determining the orientation of elongated geological bodies can be important. A fast marching technique for centerline determination, and subsequent orientation generation, is explored. The immediate application is for channel centerline extraction for use in geostatistical property modeling accounting for locally varying anisotropy. Typically, such channels are generated with multiple point statistical techniques and channel centerlines are not known. When generating property models with these facies, properties such as porosity should conform to the channel as well. The method explored can be used to determine centerlines of geological bodies. These centerlines can then be used as input point data for the generation of exhaustive maps of anisotropy necessary for many geostatistical techniques that explicitly incorporate locally varying anisotropy.

Introduction

The geometry of a deposit often aligns with the continuity of variables of interest within modeling domains. Modeling complex deposits with locally varying anisotropies, like fluvial channels, is important for accurate flow response prediction.

Here, skeletonization algorithms are used to represent the general shape of a formation. We use Sethian's fast marching method and its deviant form with multistencils in calculating the centerline of 3D ore bodies. Skeletonization of 3D objects is meant for compact representation of discrete objects, aiding in visualization, feature tracking and extraction etc. The skeleton of an object can have several definitions. Blum first put forth a precise definition as the locus of the centers of maximal disks contained in the original object. For our application, the skeleton is defined as a subset of the original dataset where (1) the skeleton has *centered* geometry within the bounds of the dataset and (2) it is largely connected. At the core of the skeletonization algorithm is the Fast Marching Method (FMM). Several variations will be explore and compared for centerline extraction.

FMMs track the evolution of a closed surface in 3D (closed curve in 2D) as it arrives at the points of a discretized lattice (wave propagation). A closed surface or interface propagation is identified by its motion, with two types of motion (1) a uniformly expanding or contracting motion normal to the current plane and (2) where the interface can be expressed as the boundary value formulation, whereas an interface advancing erratically is an initial value problem.

Boundary Value Formulation

Consider Γ *to* be the stated interface that moves in one direction with a strictly advancing or receding front with velocity *F*. The path of the interface boundary overlies a grid. In order to track the progress of the surface we denote *T(x)* as the arrival time of the front as it reaches node x. Time values can be used to infer the distance from the boundary from current location of the front. In higher dimensions the equation of motion of a propagating front is:

$$\left|\nabla T\right| F = 1 \qquad (1)$$

$$\textit{where the arrival time} T \textit{ of initial positio is set } 0;$$

$$T(\Gamma_0) = 0$$

The arrival time at the initial position is 0. The boundary value problem is formulated as in (1), where the speed depends only on the position x and is a nonlinear first-order PDE known as the Eikonal equation. The FMM approximates the solution to (1) by calculating a scalar field T.

Fast Marching Method

Skeletal points are formed by the collapse of compact interface segments as the FMM algorithm tracks the front's movement. The FMM algorithm begins from an initial set of conditions (one or several voxels) and calculates the distance field from the voxel(s). The voxels in the initial condition are frozen as the distances to the neighbors are calculated. Neighbors with computed distances are in the narrow band voxels. This narrow band propagates in the iteration of the central loop. As the voxel with the smallest distance is frozen, further distances are computed from the surrounding voxels. Thus we see the algorithm maintaining a narrow band of grid points, freezing voxels in its march. An interface is expected to traverse the grid points once, so the frozen voxels are only used to calculate distance values of others and is never recalculated.

The time field in the FMM algorithm can be stated more precisely as:

1. Central loop: search among all the narrow band points for the point with the smallest distance, extract and change its label to frozen.
2. For each neighbor that is not frozen and not in the narrow band, compute distance and insert in the narrow band. If the neighbor is in the narrow band recomputed its value.
3. Update the narrow band order as the distances are recomputed.
4. Loop back to extract the voxel with smallest distance value.

Distances are calculated from the solutions to Eikonal equation. The distance values of voxels within the narrow band are estimated so that the gradient of the arrival time is equal to reciprocal of speed of the front. Sethian (1996) proposes the following formula the square of the gradient:

$$\|\Delta T\|^2 = \begin{cases} \max(V_A - V_B, V_A - V_C, 0)^2 + \\ \max(V_A - V_D, V_A - V_E, 0)^2 + \\ \max(V_A - V_F, V_A - V_G, 0)^2 \end{cases} \qquad (2)$$

For a point with unknown distance values, the squared gradient in (2) is calculated from a six-connected neighborhood scheme. Solution to (2) is based on the use of one sided derivatives computed using forward and backward differences:

$$V_A - V_B = G[x,y,z] - G[x-1,y,z] = D^{-x}G$$

$$V_C - V_A = G[x+1,y,z] - G[x,y,z] = D^{+x}G$$

where, D^{-x} and D^{+x} are the backward and forward differences, G is the voxel lattice (voxel distance is implicitly assumed a unit distance).

The squared partial derivative in the x-direction can be approximated as:

$$T^2_x \approx \max(D^{-x}G, -D^{+x}G, 0)^2 = \max(V_A - V_B, V_A - V_C, 0)^2$$

The approximated squared gradient length can be calculated, after the partial derivatives of *y* and *z* as:

$$\|\Delta T\|^2 = T_x^{\,2} + T_y^{\,2} + T_z^{\,2}$$

FMM assumes a linear front which could create problems in computing distance from highly curvilinear fronts. This has motivated several derivatives of FMM. Often termed the higher accuracy version of FMM (FMMHA), considers the second order partial derivatives of the forward and backward differences. In addition to FMM, the FMMHA imposes the following conditions:

1. Points two voxels away in any direction from current (i,j,k) are frozen (i.e. distance values are known)
2. Points stated as in (1) must have less travel time than those voxels one point away from (i,j,k)

Multistencils Fast Marching Method

The FMM and FMMHA are highly consistent and accurate in solving the Eikonal equation but suffer from numerical errors along diagonal neighborhoods. Hassouna and Farag incorporate exhaustive diagonal information in FMM by using several stencils (Figure 1) centered at each grid point covering all possible neighbors. So the method solves the Eikonal equations at several stencils and picks the solutions satisfying the FMM causality condition, i.e. the arrival time *T(x)* depends on immediate neighbors that have smaller values. For details on its derivation interested readers are directed to Hassouna and Farag (2007).

3D Centerlines

The use of FMM or FMMHA results in a set of coordinates that define the centerline. Centerline extraction is a matter of connecting these points, where adjoining pixels give us the desired 3D surface. Figure 2 provides an example for a 3D object. In literature FMM is reported to track the evolution of closed 3D surfaces (curves in 2D) but areas where the centerline loops in on itself are possible. This is a problem because the morphology in certain parts is not captured. Hassouna et al (2007) derivate of the FMM can contour the medial axis of an curvilinear object very well, however their model does not have features where an 'arm' of the object loops into itself. To overcome this we introduce breaks where the major 'loop' or closed formations lie and then apply the methods separately (Figure 3).

The aim is to obtain a measure of continuity at every point in the modeling region. Therefore, as the skeleton represents the general shape of the body, we can use it to generate an exhaustive LVA field for the deposit of interest by kriging. However, we can also populate an LVA map by considering the directional derivative along the centerline and interpolating it locally. The derivative in each axis is representative of the surrounding features, and is assumed to be smooth.

Each voxel now stores its respective derivative, which around a small neighborhood relays the change in maximum direction of continuity of the centerline. We calculate the structure tensor from the directional vectors following the methodology outlined in paper 107 of this report. The choice of window size is important as too large a neighborhood would undermine the geometry in our narrow region of interest.

It should be noted that the model is necessarily binary. Furthermore, these methodologies show centerlines that assume continuity of the formation outside the modeling domain. To reduce the effect we add a buffer zone such that a band of zero values outline the model boundary.

The final issue is interpolating the LVA from the centerlines. While here we have generated LVA from structure tensors (Figures 4 and 6), the centerline could be digitized to be point axial data and the method from paper 123 could be used (Figure 5 and 7). In general the FMM and FMMHA both generate reasonable LVA field.

Results and Future Work

The MFMM is termed the higher accuracy variant of the original FMM. At each location the former solves the Eikonal equation along several stencils (introduces several stencils not aligned with the natural coordinate system). LVA maps from both the FMM and MFMM are similar. These methodologies provide a robust tool in extracting the general shape/features of interest for use in LVA field inference.

References

Baerentzen J. Andreas. On the implementation of Fast Marching Methods for 3D Lattices. Technical Report, Technical University of Denmark. 2001,IMM(13).

Boisvert, J B. Geostatistics with Locally Varying Anisotropy, PhD (2010). University of Alberta, Canada.

Deutsch, C V, Journel, A G. GSLIB: Geostatistical Software Library and User's Guide, Oxford University Press, New York, 2nd Ed.

Hassouna M, Farag Aly. Robust Centerline Extraction Framework Using Level Sets. IEEE Computer Society Conference CVPR. 2005.

Hassouna M, Farag Aly. Multistencils Fast Marching Methods: A Highly Accurate Solution the Eikonal Equation on Cartesian Domain. IEEE Pattern Analysis and Machine Intelligence. 2007, 29(9).

Sethian, J A. A Fast Marching Level Set Method for Monotonically Advancing Fronts, Proc. Nat. Academic Sci. 1996,93(4).p1591-1595.

Telea, Alexandru, Wijk, Jarke J. van. Joint Eurographics – IEEE TCVG Symposium on Visualization. 2002.

YanShu Yin, ShengHe Wu, Shang S. et al. A reservoir skeleton-based multiple point geostatistics method. Since in China Series D: Earth Sciences. 2009,52 (Supp,I),p171-178.

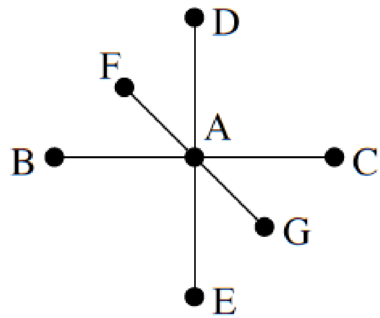

Figure 1 Stencil for a six-connected FMM neighborhood

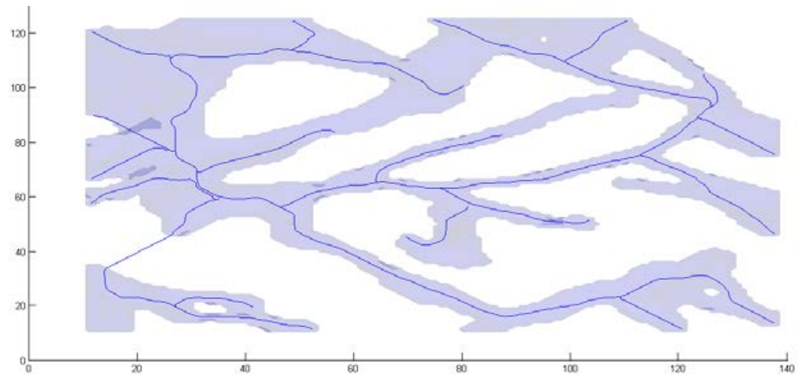

Figure 2 Centerline created using FMM

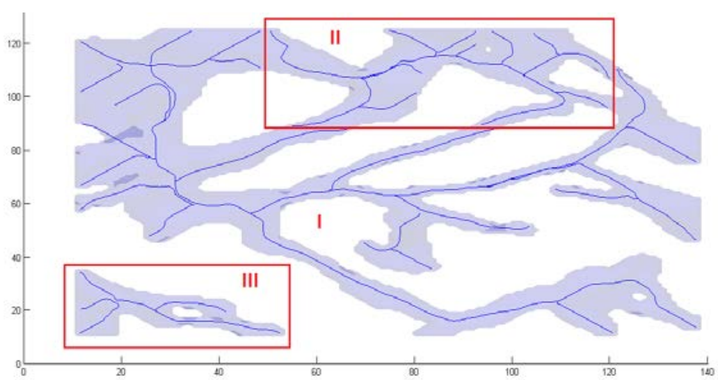

Figure 3 The presence of several 'looped' areas are cut to make different regions, and FMM applied separately. The resulting centerline traces a larger part of the model. This convention of regionalized centerline modeling is used in rest of the results.

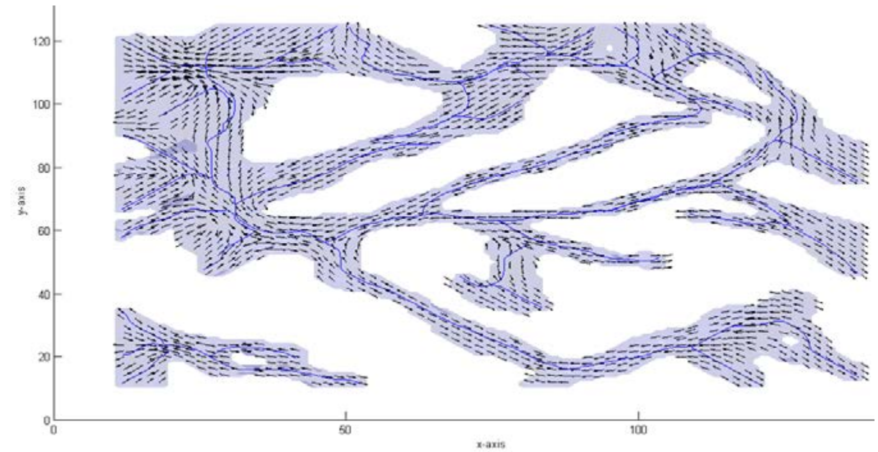

Figure 4 Plan view of LVA map populated using tensorial interpolation

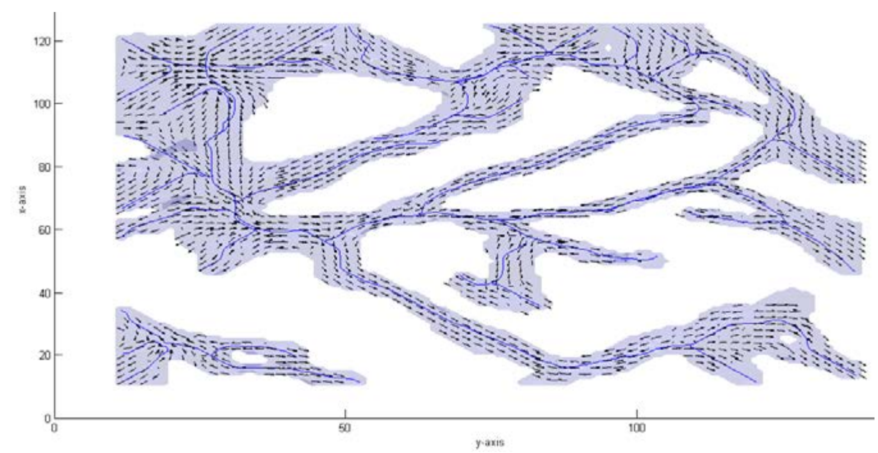

Figure 5 LVA map generated using kriging

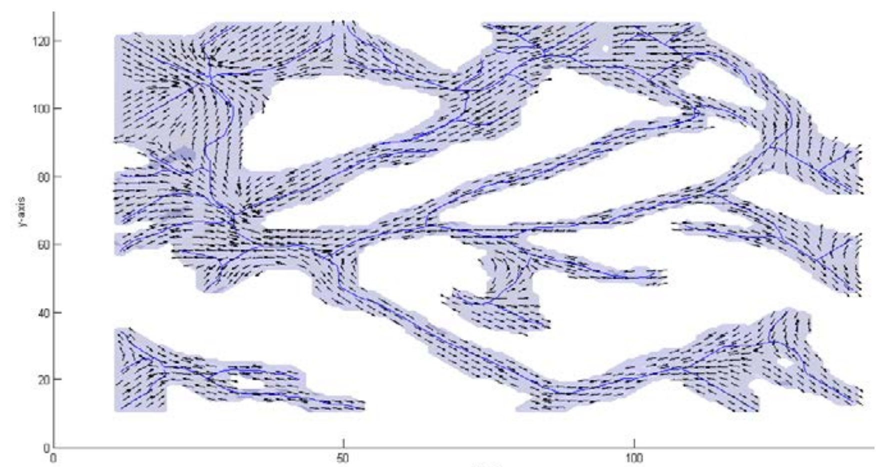

Figure 6 Centerlines generated using MFMM; LVA shown are from tensorial interpolation

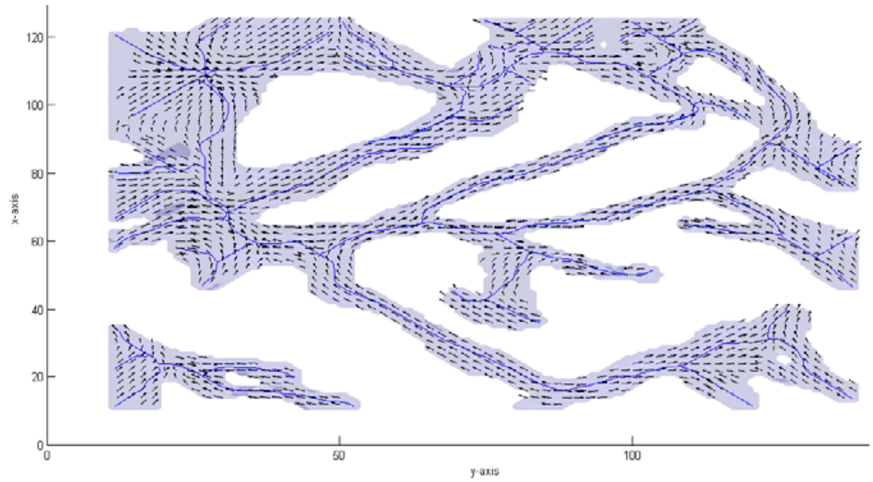

Figure 7 Kriged LVA results of the centerline values; the latter generated with MFMM