Exercises for

Chapter 8: Domain Modeling Techniques

1 EXERCISE 1

Selection is a simple visualization technique that, given some dataset D, keeps cells $c \in D$ whose data values respect a given criterion. Thereby, we can focus our exploration on specific zones of interest in the data. Consider now a scalar dataset D represented on a uniform 2D or 3D grid using quad or cubic cells respectively. Data is encoded at cell vertices. Consider now a user-given scalar value s. We next select all cells $S = \{c \in D\}$ that have at least a vertex whose data-value is lower than s and another vertex whose data-value is larger than s. What is the relationship between the selected cell-set S and the isocontour I (constructed with marching squares or marching cubes) for the same value s? Specifically, answer the following questions

- Does S contain I? Are there any parts of I not contained in S?
- Are there any cells in S through which I does not pass?

2 EXERCISE 2

Consider a simple 2D scalar dataset D, represented on a uniform grid with quad cells and cell-based data attributes. For this dataset, we next use the selection function that keeps all cells $S = \{c \in S\}$ where the cell value is larger or equal to a given user threshold s_0. The resulting cell-set S is also called a threshold-set of D. Clearly, for datasets D which do not encode a monotonic signal, S may contain several disconnected components. Now, imagine that we want to prevent this: For any given value s_0, we want that S has a single *connected component*. This means possibly adding to S some cells C that have values below s_0. Imagine an algorithm that constructs such an S, but ensures that the error created by adding the extra cells C is minimal. Outline the design of such an algorithm for the two cases below

- The error is defined as the size $|C|$of the set C. That is, we want to minimize the number of added cells $c \in C$.
- The error is defined as the sum of differences $\sum_{c \in C} |s(c) - s_0|$. We want to minimize this error.

Hints: First, construct the set S using standard thresholding. Next, consider how to incrementally add cells to S to connect its various disconnected components, while minimizing the target error.

3 EXERCISE 3

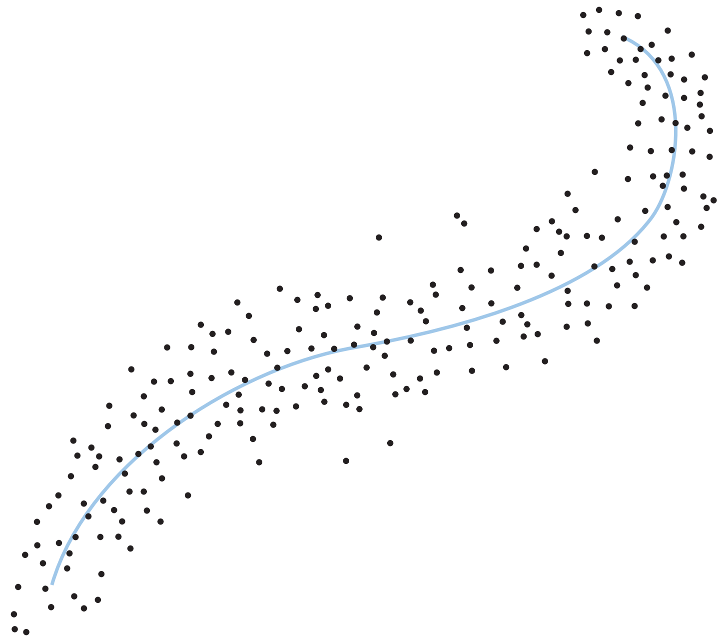
Constructing geometric primitives, such as curves and surfaces, from scattered point sets is an important building-block of many data visualization algorithms. In this context, consider a set P of N 2D points $\mathbf{p}_i$ with coordinates (x_i, y_i) being integers, such as pixels in an image. Propose an efficient algorithm that finds the straight line segment containing the most points $\mathbf{p}_i$ in P. The output of the algorithm should be the set of points $\mathbf{p}_i \in P$ that are located on this straight-line segment, ordered along the respective segment. Discuss your design and the complexity of the algorithm as function of the input point-set size N.

Hints: Since our points have integer coordinates, note that the vectors $\mathbf{p}_j - \mathbf{p}_i$ formed by two points $\mathbf{p}_i$ and $\mathbf{p}_j$ in the input set cannot assume any directions. Consider also that vectors determines by point-pairs that are located on the same line-segment will have *exactly* the same directions.

4 EXERCISE 4

Consider a 2D scattered point cloud $P = \{(x_i, y_i)\}$, like shown in the figure below. For this point cloud, we know that its points are spread relatively close to a 1D curve pattern. Imagine and describe an algorithm that constructs a 2D curve C, represented *e.g.* as a polyline, given P as input, so that C passes through the perceived *local center* of the point cloud P. For example, for the point cloud shown in the figure below, we would like to obtain a curve C like the light blue one shown in the figure. Describe the strengths and limitations of your algorithm.

Hints: There are several solutions to this problem. To begin with, you need to propose a definition of C based on the notion of 'local center' of the point cloud. Next, use this definition to construct the curve C. Possible solutions are: (a) construct a closed curve D that tightly surrounds P and next shrink D towards its center; (b) iteratively shift points in P towards their local neighborhoods' centers, and next connect the resulting points by a polyline.

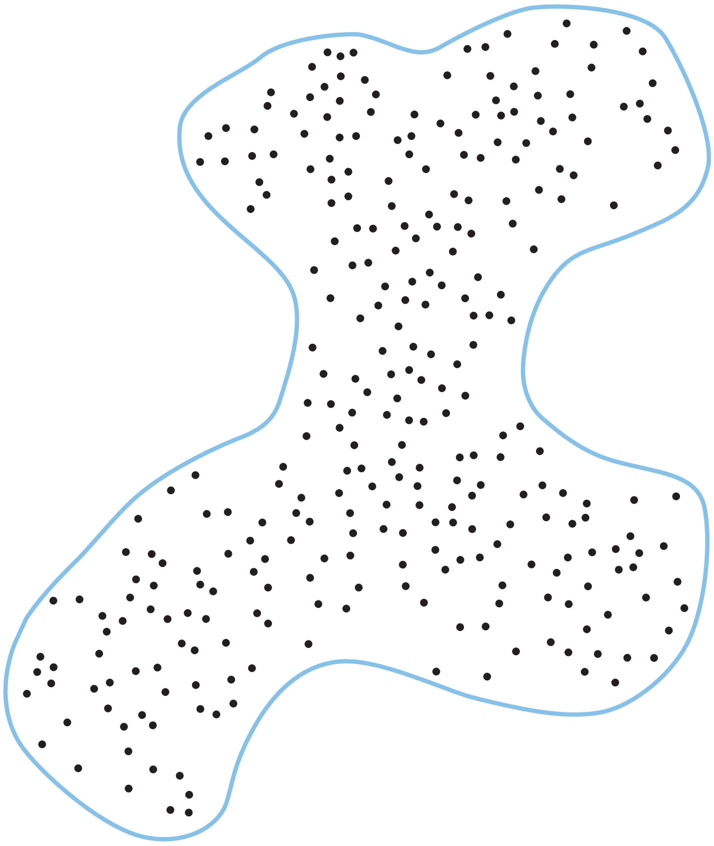


Curve extracted from a 2D point cloud.

5 EXERCISE 5

Consider a 2D scattered point cloud $P = \{(x_i, y_i)\}$, like shown in the figure below. Imagine and describe an algorithm that constructs a 2D closed curve, or contour, represented *e.g.* as a polyline, that tightly surrounds all points, and in the same time is locally smooth. For example, for the point cloud represented below, we would like to obtain the contour as the one shown in light blue in the figure.

Hints: To ensure that the surrounding contour is smooth and closed, think of generating it as an isoline of some scalar field constructed based on the points' locations. Take care, though, that isolines may have multiple disconnected components. An alternative approach is to start with a curve which surely encloses all points, and then adapt it so that it tightly surrounds the points.



Smooth curve (shown in blue) tightly surrounding a point cloud.

6 EXERCISE 6

Consider a closed 3D surface S, represented *e.g.* as an unstructured triangle mesh. We would like to segment, or partition, this surface S into several parts P_i, so that each part P_i is delimited by a sharp edge, or curvature extremum line, of the surface. For example, if S is a cube, we should obtain six parts, each being equal to a cube face; if S is a capped cylinder, we should obtain three parts, corresponding to the cylinder's two circular 'caps' and to its lateral surface, respectively. The union of all P_i should be equal to S, and no two different parts P_i and P_j should overlap. Imagine and describe a surface segmentation algorithm that constructs such a partitioning. Describe the algorithm in pseudocode, and discuss its perceived strengths and limitations.

Hints: First, describe how you would detect edges on the surface based on curvature. Note that, for a general surface, such edges may not partition the surface into disjoint segments, so you need to take care of 'gaps' between edge end-points.

7 EXERCISE 7

For a set of N 2D points, or sites, its *Voronoi diagram* consists of N planar convex polygonal cells, each cell containing the 2D-plane points closer to one of the sites. For a (closed) 2D planar curve C, its *medial axis* (or skeleton) is the set of 2D points which admit at least two different closest points in C. Given such a curve C, represented *e.g.* as a 2D pixel chain in a digital image, explain how we could use a tool or algorithm that computes the Voronoi diagram to compute a (subset of) C's medial axis.

Hints: To better understand the problem, use the applet at

`http://www.cs.cornell.edu/home/chew/Delaunay.html`

to interactively compute the Voronoi diagram of a set of 2D points. Think what happens when the point density increases.

8 EXERCISE 8

Surfaces are often represented by 3D triangular meshes. Consider such a mesh that should capture a closed (watertight), orientable, non-self-intersecting, surface. For such a mesh to be usable in many computational applications, it should meet several quality criteria concerning its vertices and triangles. List four such criteria, and argue why they are important.

Hints: To deduce the criteria, think of which vertex and/or triangle configurations should not occur if the mesh is an accurate representation of the surface type described above. Alternatively, think of geometric operations that make sense on the continuous surface and which would not be easily or robustly implementable on a mesh that does not have such properties.

9 EXERCISE 9

Surface reconstruction methods take as input a 3D point cloud C (with possibly normal information at each point), which samples a continuous 3D surface S_C, and output a 3D meshed surface S hat attempts to approximate the original S_C. Give three examples of such reconstruction methods. For each method, mention two advantages and two disadvantages as opposed to the other methods, or as you perceive them from an end-user perspective.

10 EXERCISE 10

Given a uniform 3D grid, *e.g.* consisting of cubic cells (voxels) that encode a 3D scalar field, an isosurface of this field extracted by the classical marching cubes method will contain triangles having roughly the same size. Using this observation, imagine and describe a method that, given an unstructured 3D triangle mesh representing a closed surface, would produce a refinement or simplification of this mesh where all triangles have roughly the same size, and this size is a user-controlled parameter.

Hints: Consider how to create a 3D implicit surface (using a scalar field) that is close to a given 3D triangle mesh.

11 EXERCISE 11

Consider two 2D shapes S_1 and S_2, represented *e.g.* as (the foreground pixels of) two 2D binary images on a uniform pixel grid. Propose and detail the computation of a distance function $d \to \mathbb{R}^+$ that should capture the difference, or dissimilarity between the two shapes. The value $d(S_1, S_2)$ should be zero when the shapes are identical, modulo uniform scaling and rotation in the plane, and increasingly larger than zero as the shapes are increasingly different.

Hints: One way to approach the problem is by divide and conquer. First, design d so that it recognizes shapes which are identical *and* have the same rotation in the plane and size. Next, extend d to cope with rotation and scaling.

End of Exercises for
Chapter 8: Domain Modeling Techniques