Creating Polygon Models for Spatial Clusters

Fatih Akdag, Christoph F. Eick, and Guoning Chen

University of Houston, Department of Computer Science, USA
{fatihak,ceick,chengu}@cs.uh.edu

Abstract. This paper proposes a novel methodology for creating efficient polygon models for spatial datasets. A comprehensive analysis framework is proposed that takes a spatial cluster as an input and generates a polygon model for the cluster as an output. The framework creates a visually appealing, simple, and smooth polygon for the cluster by minimizing a fitness function. We propose a novel polygon fitness function for this task. Moreover, a novel emptiness measure is introduced for quantifying the presence of empty spaces inside polygons.

Keywords: Spatial data mining, Polygon Models for Point Sets, Spatial Clustering, Polygon Fitness Function, Polygon Emptiness Measure.

1 Introduction

Polygons serve an important role in the analysis of spatial data. In particular, polygons can be used as a higher order representation for spatial clusters, such as for defining the habitat of a particular type of animal, for describing the location of a military convoy consisting of a set of vehicles, or for defining the boundaries between neighborhoods of a city consisting of sets of buildings. It is computationally much cheaper to perform certain calculations on polygons than on sets of objects. For example, polygons have been used to describe the functional regions of a city [1]. A given location can be assigned to one of those functional regions efficiently by checking in which polygon the location is included. Moreover, relationships and changes between spatial clusters can be studied more efficiently and quantitatively by representing each spatial cluster as a polygon. Polygon analysis is particularly useful to mine relationships between multiple related datasets, as it provides a useful tool to analyze discrepancies, progression, change, and emergent events [2].

However, there is not an established procedure in the literature on how to derive polygonal models from spatial clusters. The objective of the research described in this paper is to find an optimal set of polygons for two dimensional spatial clusters. The input of this process is a spatial cluster containing a set of points and its output is a polygon—the model of the cluster. As shown in Figure 1, many different polygon models (or a set of polygons as in Figure 1e) can be generated for the same set of points. Therefore, it is desirable to define application specific criteria for evaluating different polygon models. Coming up with such criteria and evaluation measures is the focus of this paper.

T. Andreasen et al. (Eds.): ISMIS 2014, LNAI 8502, pp. 493–499, 2014.

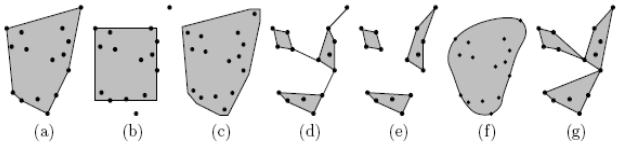

Fig. 1. Different shapes generated for the same set of points (taken from [3])

Main contributions of this paper include:

- A novel quantitative polygon fitness function is introduced to guide the generation of polygons from point clouds, alleviating the parameter selection problem when using existing polygon generation methods.
- A novel emptiness measure is introduced that quantifies the presence of empty areas in a polygon.

The rest of the paper is organized as follows. In Section 2, we compare the existing methods for creating polygon models. Section 3 provides a detailed discussion of our methodology. We present the experimental evaluation in Section 4, and Section 5 concludes the paper.

2 Related Work

Convex hulls are the simplest way to enclose a set of points in a polygon. However, convex hulls may contain large empty areas that are not desirable. Creating polygon models based on Voronoi diagrams or Delaunay triangulations is another commonly used approach. Matt Duckham et al. [4] propose a "*simple, flexible, and efficient algorithm for constructing a possibly non-convex, simple polygon that characterizes the shape of a set of input points in the plane, termed a Characteristic shape*". The algorithm firstly creates the Delaunay triangulation of the point set—which actually is the convex hull of the point set—and then reduces it to a non-convex hull by replacing the longest outside edges of the current polygons by inner edges of the Delaunay triangulation until a termination condition is met.

The Alpha shapes algorithm, introduced by Edelsbrunner et al. [5] also uses Delaunay triangulation as the starting step and generates a hull of polylines, enclosing the point set and this hull is not necessarily a closed polygon. Thus, the Alpha shapes algorithm requires post-processing for creating polygons out of the polylines. Besides, there is no easy way of determining the proper parameter for Alpha shapes algorithm.

Chaudhuri et al. [6] introduce s-shapes and r-shapes; the proposed algorithm firstly generates a staircase like shape called s-shape, which is determined using an s parameter and then reduces it to a smoother shape using the r parameter. Authors state that "*to get a perceptually acceptable shape, a suitable value of r should be chosen, and there is no closed form solution to this problem*".

A commercial algorithm, Concave Hull [7], generates polygons by using a method that is similar to the "gift-wrapping algorithm" used for generating convex hulls. It employs a k-nearest neighbors approach to find the next point in the polygon and creates a simple connected polygon unless the smoothness parameter k is too large and the points are not collinear. A density-based clustering algorithm, DContour [8] uses density contouring for generating polygonal boundaries. However, selecting the proper kernel width for the density estimation approach is non-trivial.

3 Methodology

In this section, we firstly discuss desirable polygon models and then propose a methodology that addresses the shortcomings of the existing algorithms.

3.1 Polygon Models

Figure 2 depicts three polygons that were created for the same cluster. The generated polygon in Figure 2a covers the largest area, and has the smallest perimeter, the least number of edges and the smoothest shape. However, it is obviously not a good model for the cluster because it includes large empty areas that are not relevant for the cluster. On the other hand, the polygon in Figure 2b has the largest perimeter, the most number of edges and covers the smallest area. Yet, it is also not a good representation for the cluster due to its ruggedness. Additionally, this polygon has a potential overfitting problem as it is quite complex and therefore more sensitive to noise. The polygon in Figure 2c balances the two objectives as it does not include large empty areas and has a low degree of ruggedness.

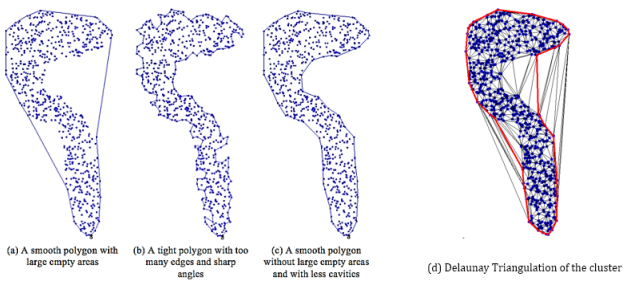

Fig. 2. Three polygons (a-c) generated for a cluster and its Delaunay Triangulation(d)

In the following, a polygon generation framework will be introduced which fits a polygon P to a set of spatial objects D minimizing the two objectives, we introduced earlier; namely, generating smooth polygons that have *a low emptiness with respect to D* and a low complexity. Additionally, we require that all objects in D are inside the polygon P. More formally, we define the problem of fitting a polygon P to a set of spatial objects as follows: Let D be a set of spatial objects in the cluster. Our goal is to find a polygon P that minimizes the following fitness function:

$$\phi(P,D) = Emptiness(P,D) + C * Complexity(P) \quad (1)$$

subject to the following constraint:

$$\forall o \in D: inside(o,P) \quad (2)$$

where *C* is a parameter which assesses the relative importance of polygon complexity with respect to polygon emptiness; e.g. if we assign a large value of *C*, smooth polygons will be preferred. *Emptiness(P,D)* is a quantitative emptiness measure that assesses the degree to which P contains empty regions with respect to D. *Complexity(P)* measures the complexity of polygon P.

3.2 Measuring the Emptiness and Complexity of P with Respect to D

In this paper we use Delaunay Triangulation DT(D) of a point set D, to define emptiness of polygon P which is supposed to model D. In general, as can be seen in Fig. 2d, areas with very low density can be identified as large triangles in the Delaunay triangulation; that is, triangles whose area is above a certain size θ. Let $P_{CONV}=(\cup_{t\in DT(D)} t)$ be the outer polygon of the DT(D) which is the convex hull of D. We define emptiness of a polygon P with respect to a point cloud D as follows:

$$\text{Emptiness}(P,D) := (\Sigma_{t\in DT(D)\wedge area(t)>\theta \wedge inside(t,P)} (area(t)-\theta) / area(P_{CONV}) \quad (3)$$

When assessing emptiness, we go through the triangles inside P and add the differences between θ and the area they cover, but only if the size of their area is above θ, and divide this sum by the area of the convex hull of D; be aware that p_{CONV} is not the area P covers, but a usually larger polygon which is the union of all triangles of Delaunay triangulation. It should be noted that when measuring emptiness, triangles that are not part of P do not contribute to emptiness.

We assess the complexity of polygons using the polygon complexity measure which was introduced by Brinkhoff et al. [9]. In this work, polygons with too many notches, having significantly smaller areas and larger perimeters compared to their convex hulls are considered complex polygons. Most importantly, it is a suitable measure to assess the ruggedness of a polygon model generated.

At the moment, we use Characteristic shapes to generate polygons in conjunction with the proposed fitness function as this produces decent polygon models. The algorithm itself has a normalized parameter *chi* which has to be set to an integer value between 1 and 100. In order to find the value of *chi* which minimizes the employed fitness function we exhaustively test all 100 *chi* values, and return the fittest polygon.

4 Experimental Evaluation

In this section, we present the experimental results using the fitness function ϕ defined in equation (1) for spatial clusters in a dataset called Complex8 [10]. Figures 3b-3d depict the polygons generated for the cluster in Fig. 3a using different C parameters and Table 1 reports the area, perimeter, emptiness, and complexity for polygons in

these figures along with the optimal *chi* parameter values selected by the fitness function to create these polygons. We observe that, setting C=0.35 gives quite reasonable results. We also observe that setting C to higher values generates larger polygons. For very low C values, the generated polygons are quite complex having many edges and cavities.

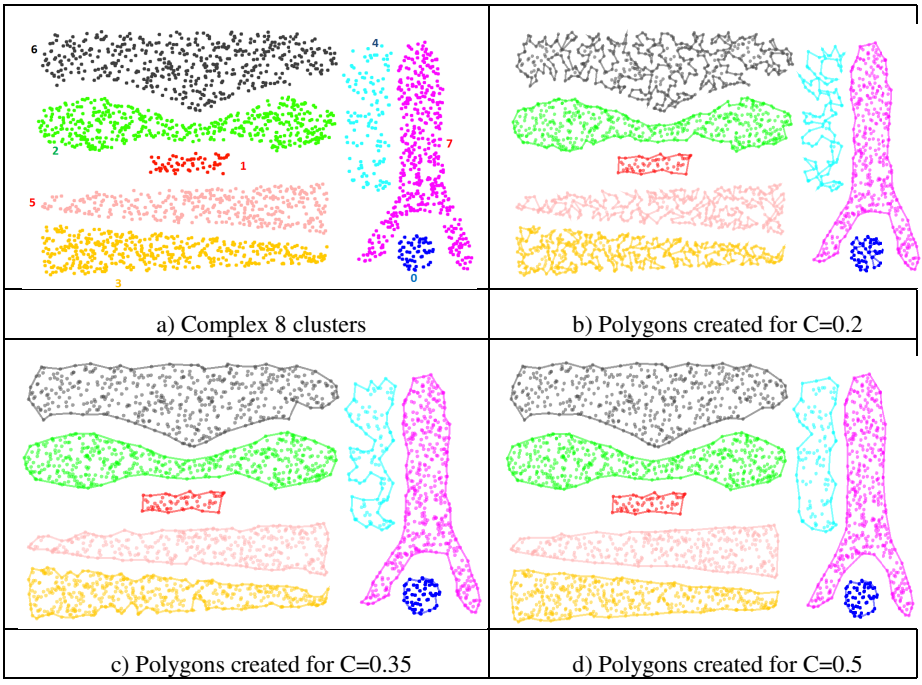

Fig. 3. Complex 8 Dataset and polygons generated using different chi parameters

Table 1. Statistics for polygons in Figures 3b-3d separated by comma in respective order. P0-P7 represent polygons for clusters 0-7 in the dataset and colored respectively.

	area	perimeter	emptiness	complexity	chi
P0	1088, 2030, 2030	328, 173, 173	0.077, 0.219, 0.219	0.49, 0.02, 0.02	37, 70, 70
P1	2697, 2697, 2741	287, 287, 286	0.144, 0.144, 0.148	0.052, 0.052, 0.046	34, 34, 37
P2	21492, 23107, 23107	1052, 997, 997	0.084, 0.096, 0.096	0.125, 0.072, 0.072	6, 13, 13
P3	9477, 18057, 20146	2465, 1058, 954	0.072, 0.192, 0.233	0.589, 0.118, 0.02	2, 5, 10
P4	4829, 8408, 11246	1171, 751, 561	0.057, 0.113, 0.197	0.562, 0.319, 0.089	5, 10, 16
P5	9007, 19122, 20413	2460, 968, 947	0.063, 0.19, 0.211	0.606, 0.043, 0.015	2, 8, 18
P6	17560, 34719, 35061	3019, 1018, 1003	0.044, 0.162, 0.168	0.632, 0.054, 0.04	4, 13, 14
P7	19759, 19759, 20807	1003, 1003, 984	0.042, 0.042, 0.049	0.188, 0.188, 0.17	8, 8, 14

It can be seen that quite different chi values are chosen for different spatial clusters by our approach. The polygon P4 (cyan-colored) best illustrates the effect of changing the C parameter. The generated polygon for P4 in Fig. 3b is very tight and rugged having a smaller area, larger perimeter, smaller emptiness and larger complexity values compared to polygons generated with larger C values. On the other hand, the generated polygon for P4 in Fig. 3d is smoother; it has fewer edges and empty areas producing a larger area and emptiness value, smaller perimeter and a smaller complexity value.

5 Conclusion

In this paper, we proposed a methodology for creating simple polygons for spatial clusters. As popular polygon model generation algorithms have input parameters that are difficult to select, we introduced a novel fitness function to automate parameter selection. We are not aware of any other work that uses this approach. The fitness function balances the complexity of the polygon generated and the degree the polygon contains empty areas with respect to a point set. The methodology uses the Characteristic shapes algorithm in conjunction with the fitness function. We also claim that the proposed fitness function can be used in conjunction with other polygon generating algorithms, such as the Concave Hull algorithm, and Alpha shapes.

We tested the methodology with Complex 8 dataset and our methodology proved to be effective at creating desired polygon models. When used with our polygon fitness function, the Characteristic shapes algorithm generated very accurate polygon models. As a future work, we plan to extend our methodology to allow for holes in polygons and for polylines in spatial cluster models.

References

1. Cao, Z., Wang, S., Forestier, G., Puissant, A., Eick, C.F.: Analyzing the Composition of Cities Using Spatial Clustering. In: Proc. 2nd ACM SIGKDD International Workshop on Urban Computing, Chicago, Illinois (2013)
2. Wang, S., Chen, C.S., Rinsurongkawong, V., Akdag, F., Eick, C.F.: A Polygon-based Methodology for Mining Related Spatial Datasets. In: Proc. of ACM SIGSPATIAL International Workshop on Data Mining for Geoinformatics (DMG), San Jose (2010)
3. Galton, A., Duckham, M.: What is the region occupied by a set of points? In: Raubal, M., Miller, H.J., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2006. LNCS, vol. 4197, pp. 81–98. Springer, Heidelberg (2006)
4. Duckham, M., Kulik, L., Worboys, M., Galton, A.: Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. Pattern Recognition 41, 3224–3236 (2008)
5. Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R.: On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29, 551–559 (1983)
6. Chaudhuri, A.R., Chaudhuri, B.B., Parui, S.K.: A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border. Computer Vision and Image Understanding 68, 57–275 (1997)

7. Moreira, A., Santos, M.Y.: Concave hull: a k-nearest neighbours approach for the computation of the region occupied by a set of points. In: International Conference on Computer Graphics Theory and Applications GRAPP (2007)
8. Chen, C.S., Rinsurongkawong, V., Eick, C.F., Twa, M.D.: Change Analysis in Spatial Data by Combining Contouring Algorithms with Supervised Density Functions. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 907–914. Springer, Heidelberg (2009)
9. Brinkhoff, T., Kriegel, H.-P., Schneider, R., Braun, A.: Measuring the Complexity of Polygonal Objects. In: Proc. of the Third ACM International Workshop on Advances in Geographical Information Systems, pp. 109–117 (1995)
10. Salvador, S., Chan, P.: Determining the Number of Clusters/Segments in Hierarchical clustering/Segmentation Algorithm. In: ICTAI, pp. 576–584 (2004)