Contextual Line Generalization – Extending ArcGIS Generalization Toolset

Titus Tienaah [1,]*, Emmanuel Stefanakis [2], David Coleman [3]

Geodesy and Geomatics Engineering,
University of New Brunswick, Fredericton, New Brunswick, E3B 5A3.
{[1] ttienaah, [2] estef, [3] dcoleman}@unb.ca

Abstract

Simplification is an important operator in cartographic generalization. Simplification of geometries in isolation or out of spatial context often leads to topological and other spatial relation inconsistencies. The cartography toolbox of ArcGIS provides some functionality for topological line simplification. There is however, limited support for contextual simplification of static and mobility data using other layers as contextual constraints. It is important to extend the ArcGIS generalization toolset to support spatiotemporal data: these datasets have become commonplace with the advent of GPS and location based mobile devices.

This paper implements and extends the line generalization toolset in ArcGIS. The extension supports two-dimensional spatial and temporal simplification of lines using other layers as contextual constraints. The paper highlights results of test on real world spatial and spatiotemporal data.

Keywords: Line Generalization, Trajectory Simplification.

1 Introduction

Spatial data collection for topographic mapping is an expensive process, it is cost effective to collect data once, at the highest resolution, and then derive spatial and temporal representations at different scales. A solution to a multi-scale derived product through automated generalization is an important component for most national mapping and cadastral agencies. Source data collected at 1: 5000 can serve as a base data for derive products at 1:10,000, 1:25,000 and 1:50,000 without empirical data collection at the smaller scale. Multi-scale automated generalization is still an unsolved problem. In this paper we focus on contextual line simplification as a contribution towards consistent automated generalization.

Our main contribution is a contextual Douglas-Peucker [Douglas and Peucker, 1973] algorithm for static and spatiotemporal data. Line simplification is fundamental and well covered topic in cartographic generalization [e.g., Ramer, 1972; Douglas and Peucker, 1973; Reumann and Witkam, 1974; Opheim, 1982; Li and Openshaw, 1992; Normant and Tricot, 1993; Visvalingam and Whyatt, 1993; Fritsch and Lagrange, 1995; Balboa and López, 2000; Doihara et al., 2002; Qingsheng et al., 2002; Wu and Deng, 2003; Saux, 2003; Gribov and Bodansky, 2004; Kulik et al., 2005; Guilbert and Saux, 2008; Abam et al., 2010; Daneshpajouh and Ghodsi, 2011; Lei et al., 2012; Liu et al., 2012; Raposo, 2013; Li, 2013, Abam et al., 2014]. The Douglas-Peucker algorithm is one of the most popular algorithms in line simplification. We focus on line simplification as the as a useful abstraction between point data and area/regional data (e.g. polygon). Algorithmic solutions for line simplification can be generalized for higher forms of geometry composed of lines.

2 Background

In this paper we extend ArcGIS generalization toolset to provide consistent simplification of vessel trajectories using islands as a spatial context. We explore our implementation by extending ESRI ArcGIS generalization toolset. ArcGIS is one of the most widely used geospatial software in education and enterprise GIS. The Douglas-Peucker algorithm as implemented in the cartographic toolset has an implicit scope to reduce the set of vertices used to represent a polyline based on distance as an error offset. Simplification of geometric properties without context may change the meaning of such properties (semantic representation).

The *Simplify_line* (ArcGIS version10.2.2, 2015) tool provides an option to correct topological errors, such as line crossing, line overlaps, and zero-length lines. The error correction mechanism provided by ArcGIS is important and serves as a means to remove errors introduced as a result of simplification. We classify these errors as internal topological errors. Our contribution will focus on external topological inconsistencies that are violated by not observing contextual relationship between the line to be generalized and the space constraining its shape. In Figure 1, (a) and (b) violates a disjoint and direction (east side) relation respectively.

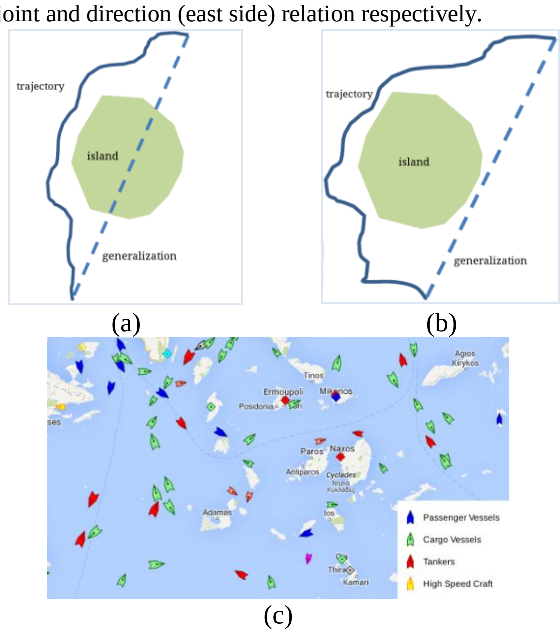


Figure 1: Source (solid line) and generalized (dashed line) trajectory against one polygon (island) in the same planar space (a, b); vessel trajectories in the Aegean, a snapshot from

Marine Traffic AIS (www.marinetraffic.com) (c). After Stefanakis [2012]

2.1 Contextual Simplification

The *point remove* option of *line_simplify* is efficient for data compression and for eliminating redundant details. As of ArcGIS version 10.2.2, there no support for constraining the shape of geometries with other layers serving as semantic constraints during line simplification. There is also no support for spatiotemporal line simplification. To demonstrate the importance of contextual simplification, we will strip the time component of trajectory data to represent it as two dimensional data and use islands as contextual information in the Aegean. Figure 2 uses *line_simplify* of ArcGIS and show a semantic violation of representing the original 2d trajectory. The trajectory does not intersect the island but both the constrained and unconstrained simplified lines violate the semantic representation of the trajectory.

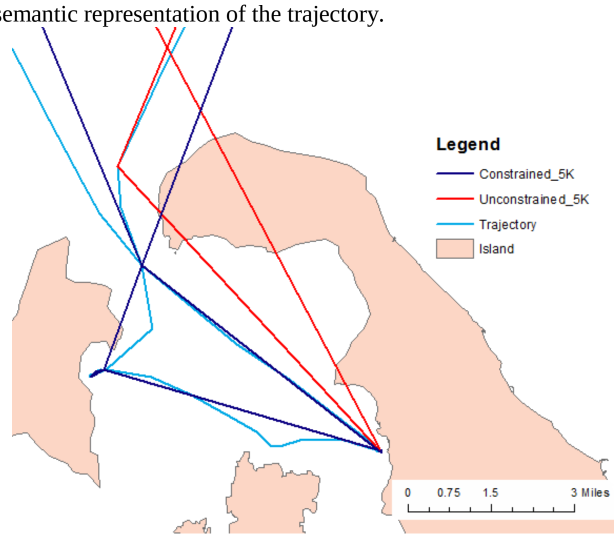


Figure 2: 2D Simplification - ArcGIS *Simplify_Line* at 5kilometres

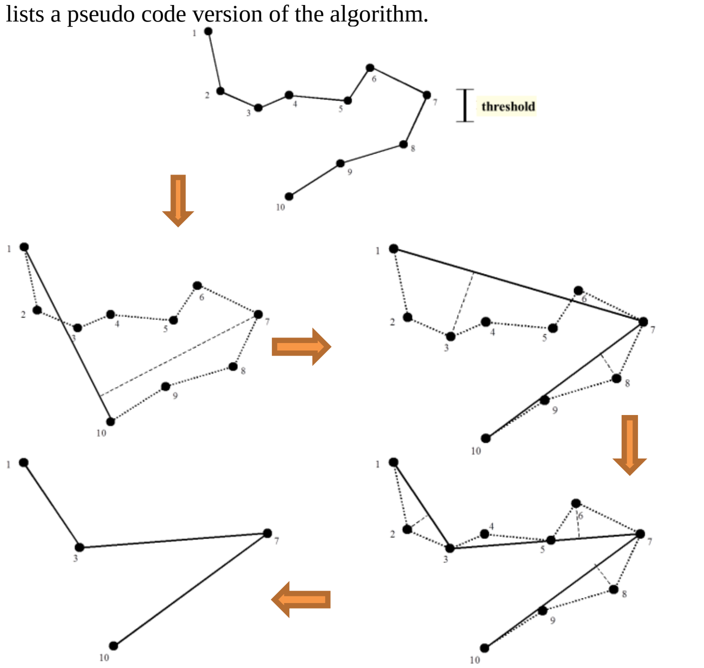
By extending the DP algorithm to handle spatiotemporal data, we contribute to cartographic toolbox with a new add-on to the generalization toolset with contextual simplification. Context geometries are geometries that give some spatial or thematic meaning to a polyline on a map. We define neighbourhood geometry as geometries intersecting the convex hull of vertices forming the polyline. The convex hull forms a polygon with a minimum set of vertices that envelope all the vertices of the polyline.

A simplified line is a subset of all vertices of the original polyline, such a line is completely within or shares boundary with convex hull. Contextual geometries that intersect the convex hull may have disjoint, intersect, or side relationship with the original polyline. Our implementation establishes a geometric intersection signature, direction and distance between lines and context geometries. A simplified line reinstates prioritized list of vertices from the original line to resolve constraint violation.

3. Synchronous Euclidean Distance

The algorithms for the generalization of trajectories are usually straight extensions of the algorithms for generalizing linear features in cartography (Robinson et al. 2005). In this paper, the Douglas-Peucker algorithm (1973), as extended by the notion of the Synchronous Euclidean Distance (Maratnia and de By 2004), is considered.

The input data in the original Douglas-Peucker (DP) algorithm is an ordered set of points (shaping a curve or trajectory) and a threshold distance $\varepsilon > 0$. The algorithm recursively divides the line. Initially, it marks the first and last point to be kept. The two points define a straight line segment as an approximation of the ordered set of points. It then finds the point in the set that is furthest from the line segment. If the point is closer than ε to the line segment, then any points not currently marked to keep can be discarded. Else, if the point is further than ε, that point must be kept and the new approximation of the set of points consists of two straight line segments. The algorithm recursively calls itself for each line segment. When the recursion is completed, a new line can be generated consisting of all (and only) those points that have been marked as kept. Figure 3(a) presents an example of the DP algorithm in generalizing a polyline, while Figure 3(b) lists a pseudo code version of the algorithm.



(a)

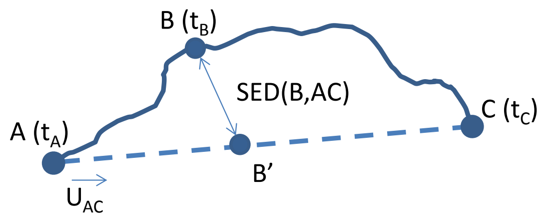
```
function DouglasPeucker(PointList[], threshold)
 report { PointList[1], PointList[end] }
 //Find the point (index) with the maximum distance (dmax)
 dmax = 0
 index = 0
 for i = 2 to (length(PointList) - 1)
  d = MinimumDistance(PointList[i], Line(PointList[1],PointList[end]))
  if d > dmax
   index = i
   dmax = d
  end
 end
  //If max distance is greater than the threshold, recursively simplify
 if dmax >= threshold
  report PointList[index]
  //Line division and recursive call
  DouglasPeucker(PointList[1...index], threshold)
  DouglasPeucker(PointList[index...end], threshold)
end
```

(b)

Figure 3: The Douglas-Peucker (DP) algorithm.

3.1 Synchronous Euclidean Distance

DP algorithm by itself is not able to consider the temporal dimension of a trajectory. This is achieved by introducing the notion of the Synchronous Euclidean Distance (SED; (Meratnia and de By 2004). In a trajectory, each point *i* is assigned a temporal stamp (t_i), which determines the time that the moving object crossed the point. Let A, B and C be three spatiotemporal locations recorded for a trajectory T, with $t_A < t_B < t_C$ (Figure 4a). Then, the SED for the point B is equal to the Euclidean distance BB', where the location B' is calculated with respect to the velocity vector U_{AC} (Figure 4b). In other words, the distance of point B from the straight line approximation AC is equal to BB', where B' is the spatiotemporal trace of B on that approximation at time t_B.



(a)

$$U_{AC} = \frac{AC}{t_C - t_A}$$

$$AB' = U_{AC}(t_B - t_A)$$

(b)

Figure 4: The Synchronous Euclidean Distance.

The notion of SED is introduced in the pseudo code of Figure 3b by replacing line 6 with the following line:

```
6'    d = SED(PointList[i], Line(PointList[1],PointList[end]))
```

In general, the SED (BB') is not perpendicular to the straight line approximation (AC) and hence its length is greater or equal to the perpendicular distance applied in DP algorithm. Figure 7 shows an SED simplification of trajectory with islands as constraint. We preserve geometric and direction relationship of the original line with its surrounding space by reinstating prioritized vertices based on SED distance to resolve inconsistent semantic simplification.

At each level of the SED decomposition of the trajectory, we establish a direction signature for the original line. This signature describes a side relation of a trajectory to each of its contextual neighbours. For example, Figure 6 shows a nine cell quadrant formed by the bounding box of a context neighbour with respect to a polyline. The signature formed is a *True* or *False* indicating intersection with the quadrant, starting from north-west, north, north-east… and ends at south-east: TTF-TTF-FTF. The simplified trajectory must preserve this signature to maintain the direction relation of the original trajectory.

Using *ArcPy* - a python site-package for ArcGIS, constrained DP and SED algorithms are added as an application add-in toolbar (Figure 5). These provide constrained spatio-temporal simplification and constrained Douglas-Peucker simplification using a second spatial layer as context.

Figure 5: Simplification Toolbar

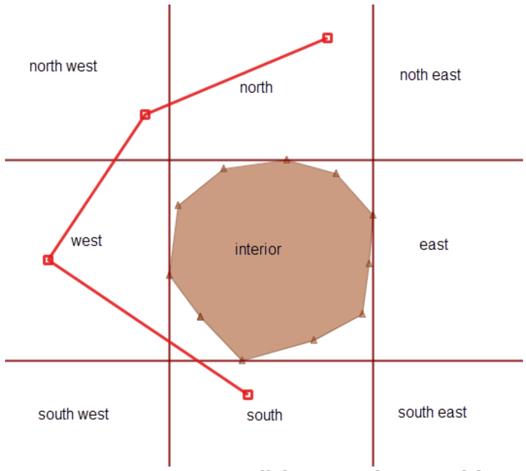


Figure 6: Nine cell direction relation model

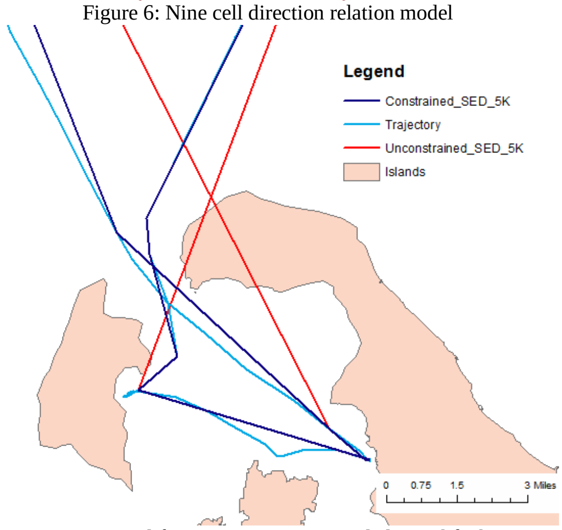


Figure 7: SED simplification at 5Km - Extended *Simplify_line*

4. Conclusion

In this paper, we have presented a technique of consistent simplification by extending the cartographic toolbox of ArcGIS. This contribution is an important step towards multi-scale automated generalization. Future work will focus on experimental evaluation and application to other forms of mobility data.

5 References

- Abam, M. A., Daneshpajouh, S., Deleuran, L., Ehsani, S., and Ghodsi, M. (2014). Computing homotopic line simplification. Computational Geometry, 47(7), 728-739.

- Abam, M. A., Berg, M., Hachenberger, P., and Zarei, A. (2010). Streaming Algorithms for Line Simplification. Discrete and Computational Geometry, 43(3), 497-515.
- ArcGIS,(2015).Cartography toolbox - *SimplifyLine _cartography* [online]. Available from: http://bit.ly/1ycbS52 [Accessed 8th Feb. 2015].
- Balboa, J. L. G. and López, F. J. A. (2000). Frequency Filtering of Linear Features by Means of Wavelets. A Method and an Example. Cartographic Journal, 37(1), 39-49.
- Ballard, D. H. (1981). Strip trees: a hierarchical representation for curves. Commun. ACM, 24(5), 310-321.
- Daneshpajouh, S. and Ghodsi, M., (2011). A Heuristic Homotopic Path Simplification Algorithm. In: MURGANTE, B., et al. eds. Computational Science and its Applications - ICCSA 2011. Springer Berlin Heidelberg, 132-140.
- Doihara, T., Wang, P., and Lu, W. (2002). An Adaptive Lattice Model and its Application to Map Simplification. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(4), 259-262.
- Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2), 112-122.
- Fritsch, E. and Lagrange, J., (1995). Spectral representations of linear features for generalisation. In: Frank, A. and Kuhn, W. eds. Spatial Information Theory A Theoretical Basis for GIS. Springer Berlin Heidelberg, 157-171.
- Gribov, A., and Bodansky, E. (2004). A New Method of Polyline Approximation. In A. Fred, T. Caelli, R. W. Duin, A. Campilho and D. De Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition (Vol. 3138, pp. 504-511): Springer Berlin Heidelberg.
- Guilbert, E., and Saux, E. (2008). Cartographic generalisation of lines based on a B-spline snake model. Int. J. Geogr. Inf. Sci., 22(8), 847-870. doi: 10.1080/13658810701689846
- Kulik, L., Duckham, M., and Egenhofer, M. (2005). Ontology-driven map generalization. Journal of Visual Languages and Computing, 16(3), 245-267.
- Lei, W., Yiming, G., Chen, C., and Yaowei, Y. (2012, 2-4 Nov. 2012). A Spatio-temporal Data Compression Algorithm. Paper presented at the Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference.
- Li, Z., and Openshaw, S. (1992). Algorithms for automated line generalization1 based on a natural principle of objective generalization. International Journal of Geographical Information Systems, 6(5), 373-389.
- Li, L., et al. (2013). An Algorithm for Fast Topological Consistent Simplification of Face Features⋆. Journal of Computational Information Systems, 9(2), 791-803.
- Liu, G., Iwai, M., and Sezaki, K. (2012). A method for online trajectory simplification by enclosed area metric. Paper presented at the Proc. of the Sixth International Conference on Mobile Computing and Ubiquitous Networking.
- Meratnia, N. and By, R., (2004). Spatiotemporal Compression Techniques for Moving Point Objects. In: BERTINO, E., et al. eds. Advances in Database Technology - EDBT 2004. Springer Berlin Heidelberg, 765-782.
- Normant, F., and Tricot, C. (1993). Fractal simplification of lines using convex hulls. Geographical Analysis, 25(2), 118-129.
- Opheim, H. (1982). Fast data reduction of a digitized curve. Geo-Processing, 2, 33-40.
- Qingsheng, G., Brandenberger, C., and Hurni, L. (2002). A progressive line simplification algorithm. Geo-spatial Information Science, 5(3), 41-45. doi: 10.1007/BF02826387
- Raposo, P. (2013). Scale-specific automated line simplification by vertex clustering on a hexagonal tessellation. Cartography and Geographic Information Science, 40(5), 427-443.
- Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing, 1(3), 244-256.
- Reumann, K., and Witkam, A. (1974). Optimizing curve segmentation in computer graphics. Paper presented at the Proceedings of the International Computing Symposium.
- Robinson, A.H., Morrison, J.L., Muehrcke, P.C.,Kimerling, A.J., Guptill, S.C., 2005. Elements of Cartography. 6th Edition. Wiley.
- Stefanakis, E., Trajectory generalization under space constraints. ed. the Proceedings of the 7th International Conference on Geographic Information Science, 2012, 18-21.
- Saux, E. (2003). B-spline Functions and Wavelets for Cartographic Line Generalization. Cartography and Geographic Information Science, 30(1), 33-50.
- Visvalingam, M. and Whyatt, J. D. (1993). Line generalisation by repeated elimination of points. Cartographic Journal, The, 30(1), 46-51.
- Wu, F., and Deng, H. (2003). Using genetic algorithms for solving problems in automated line simplification. Acta Geodaetica Et Cartographic Sinica, 4, 013.