Development of an Improved Spatial Metadata Simplification Algorithm

Sarah Hafer-Martin[1,2,3], Walter Baskin[1,2,3], Georgina Hayes-Crepps[1,2,4]

Atmospheric Science Data Center[1], NASA Langley Research Center[2], ADNET Systems, Inc[3] and Analytical Mechanics Associates[4]

Introduction

The National Aeronautics and Space Administration's (NASA) Atmospheric Science Data Center (ASDC) at NASA Langley Research Center in Hampton, VA provides atmospheric science data products and services to the science community, including enhanced search and subsetting capabilities for numerous Earth Science datasets.

The ASDC is the official Distributed Active Archive Center (DAAC) of record for the Tropospheric Emissions: Monitoring of Pollution (TEMPO) instrument. TEMPO is situated on a geostationary satellite positioned at a longitude near the center of the conterminous United States and focused on North America, making hourly swaths of its field of regard from east to west.

Spatial metadata is an essential component for the discovery and distribution of Earth Science data. The simplified polygonal boundaries representing the archived data files ensure that any granule can be identified quickly and accurately by a spatial query. Historically the Douglas-Peucker algorithm has been used for polygon simplification; however, due to the nature of the algorithm, a buffer must be added to the polygon before simplification to ensure pivotal points are not removed by the algorithm. This adds in additional error to the polygon simplification. ASDC's goal is to test to other methods of polyline simplification, such as Visvalingam-Whyatt, Zhou-Jones and Wang-Mueller simplification alongside of Douglas-Peucker and different buffering methods, to produce less error during polygon simplification of TEMPO data swaths, and special spatial query geometries such as U.S. Environmental Protection Agency (EPA) non-attainment regions, and geopolitical boundaries.

Method

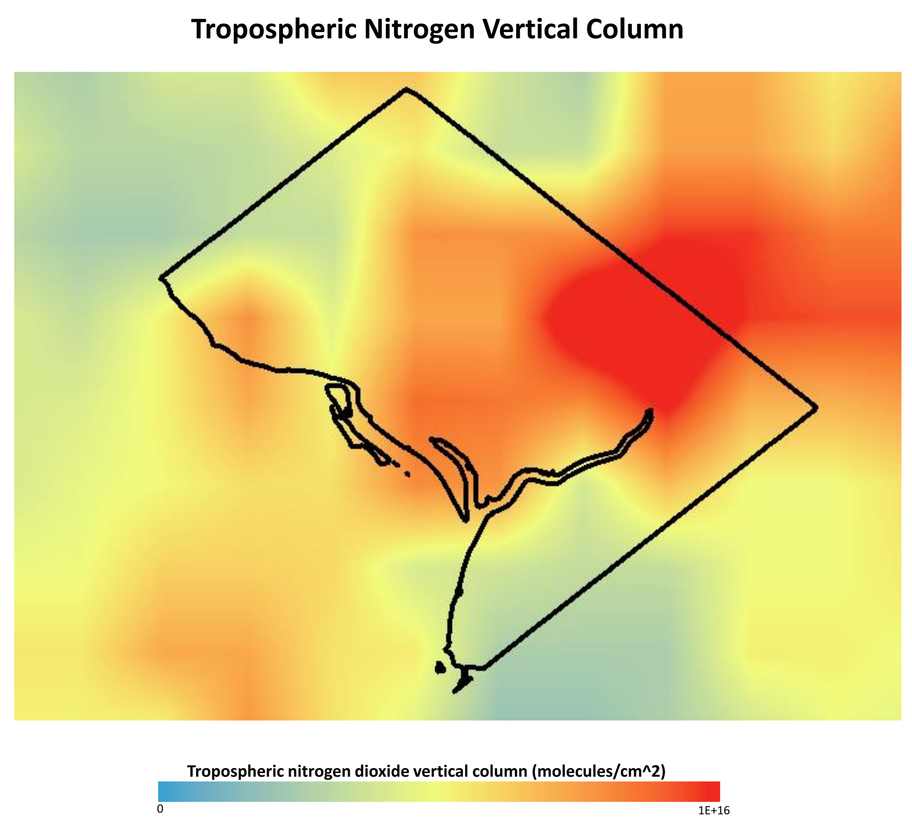
The polygon simplification algorithms were tested on the EPA non-attainment region over DC for 8-Hour Ozone (2015 Standard). This polygon was chosen because the location of DC is of high interest as it is in TEMPO's path and for both the complexity and simplicity of different sections of the polygon. DC has both a river running through it with islands and clear straight lines with 90-degree angles. This allows us to determine how the simplification algorithms perform under both circumstances.

Four different algorithms were tested in ESRI ArcPro: Douglas-Peuker (retain critical points), Visvalingan-Whyatt (retain effective areas), Zhou-Jones (retain weighted effective areas), and Wang-Mueller (retain critical bends). Each one was tested with a tolerance of 0.01 degree (~1km at equator). Each test was run with and without buffering of the original polygon. This was done to compare margin of error. Buffering helps reduce error by preventing omission of data from within the original spatial coverage. However, it can increase error by adding additional area to the polygon where it is not needed.

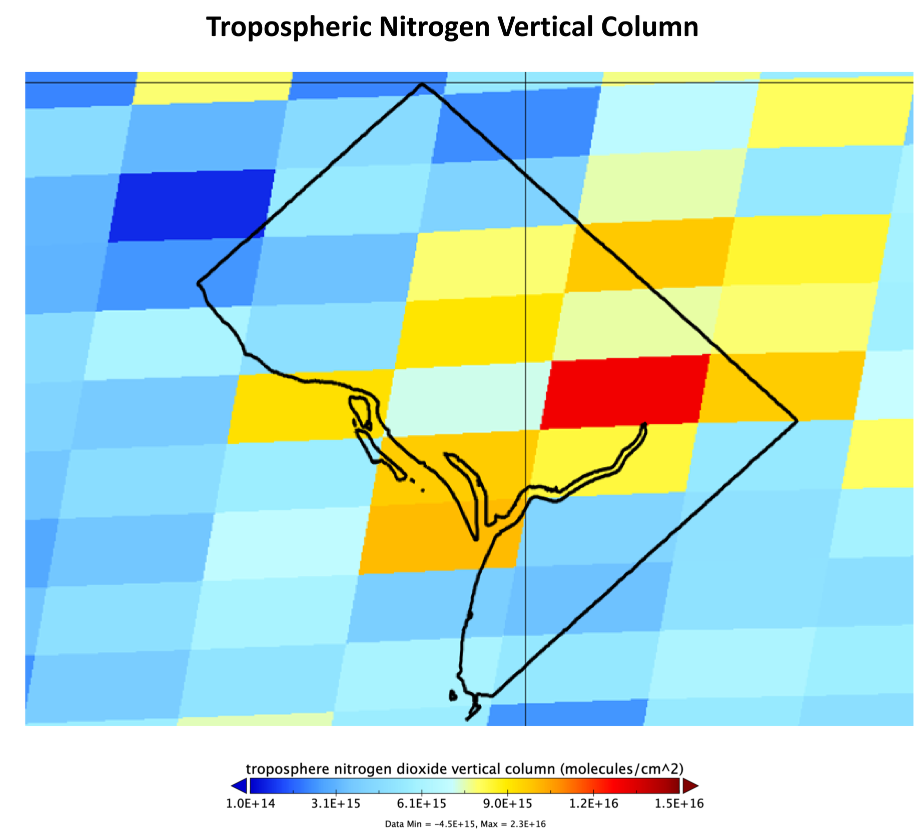
Douglas-Peuker, Douglas-Peuker preserve topology, and Visvalingan-Whyatt were tested in OpenJUMP, an opensource GIS application. They were tested using an epsilon value of 0.01 degree of tolerance.

Since different algorithms each have different use cases and best practices, the non-attainment region was overlaid with a granule of TEMPO data includes the Washington DC area. We used TEMPO NO2 tropospheric nitrogen vertical column as part of this analysis. This provided context to our evaluation of the effectiveness of the simplifications as applied to TEMPO data. Simplification needs change depending on resolution of data; Lower resolutions allow for the use of more simplification, while higher resolution requires that the simplification process does not cut out any area that would cause a loss of data coverage.

TEMPO Data Overlaid with Polygon



DC 8-hour ozone non-attainment region over TEMPO Proxy NO2 tropospheric data in ArcPro



DC 8-hour ozone non-attainment region over TEMPO Proxy NO2 tropospheric data in Panoply

Discussion:

TEMPO data is archived in NetCDF format. In order to use this data in ArcPro it must first be converted to a raster. This can be done by adding the data to a mosaic dataset or by using the add Multidimensional Raster tool. When the TEMPO Proxy data are added in ArcPro we get a smooth visual representation of the data as seen on the left. While this is ideal for most scientific needs, it does not allow us to see pixel bounds. The image on the right was done in Panoply, which does allow us to see pixel bounds. Seeing pixel bounds allows us to clearly see the resolution compared with the polygon and gives us and understanding of how much we need to worry about data not being included in the simplified polygon verses the original if some of the are from the original polygon is lost. Here we can see that there is a very course resolution of data compared to the polygon.

Conclusions

Each simplification method has its uses. While Douglas-Peuker and Visvalingan-Whyatt are effective at efficiently reducing the number of vertices, they do not preserve the shape's integrity at lower resolutions. This is evident in the Results section below. The number of vertices is significantly reduced; however, portions of the shape's coverage have been lost, which, when using high resolution data could become an issue if data points are removed due to using an over-simplified polygon. Buffering can help with this issue but adds on additional area to the polygon that may result in false positives spatial search results. In all four cases, a 200-meter buffer was not enough to ensure none of the original area was removed from the simplified polygon. Adding a buffer also causes the shape to lose some features, such as a river. It is important to keep that in mind if you are trying to retain those features. Wang-Mueller does a very good job at retaining the shape's integrity. With the 200-meter buffer, there is only a very small portion of original area lost with the simplification. In the buffered Zhou-Jones simplification, the corners of the polygon are cut off, which is problematic; however, the buffered version does slightly better.

Proposed future tests:

- Cut the polygon into parts. This would allow for two additional capabilities. First, is the ability to run different simplification algorithms on different parts of the polygon, allowing the process to get the benefits of multiple algorithms based the complexity of each polygon section.
- Only buffer certain parts of the polygon (convex sections) so additional area is not being added unnecessarily.
- Combine both techniques listed above.

The resolution of TEMPO Proxy data is low compared to the resolution of the Washington DC non-attainment region used in this investigation. When small sections of area are removed from the polygon it is not going to cause the elimination an entire pixel of data over the given area. This allows us to use larger epsilon values in the simplification algorithms and improve the efficiency of spatial searches that utilize this spatial metadata.

References

DOUGLAS, D. H., & 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. https://doi.org/10.3138/fm57-6770-u75u-7727

Douglas-Peucker algorithm. Cartography Playground. (n.d.). https://cartography-playground.gitlab.io/playgrounds/douglas-peucker-algorithm/

ESRI. (n.d.). *How simplify line and simplify polygon work*. How Simplify Line and Simplify Polygon work-ArcGIS Pro | Documentation. https://pro.arcgis.com/en/pro-app/latest/tool-reference/cartography/how-simplify-line-works.htm

Visvalingam, M., & Whyatt, J. D. (1993). Line generalisation by repeated elimination of points. *The Cartographic Journal*, *30*(1), 46–51. https://doi.org/10.1179/caj.1993.30.1.46

Wang, Z., & Müller, J.-C. (1998). Line generalization based on analysis of shape characteristics. *Cartography and Geographic Information Systems*, *25*(1), 3–15. https://doi.org/10.1559/152304098782441750

Zhou, S., & Jones, C. B. (2005). Shape-aware line generalisation with weighted effective area. *Developments in Spatial Data Handling*, 369–380. https://doi.org/10.1007/3-540-26772-7_28

Results

Polygon Simplification in ArcPro and OpenJUMP

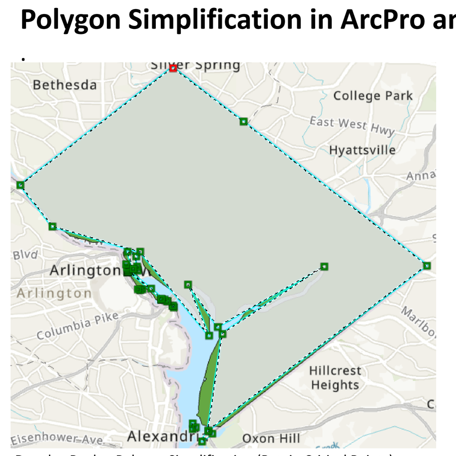
Douglas-Peuker

Polygon Simplification available in ArcPro , OpenJump, and Python as 'POINT_REMOVE'

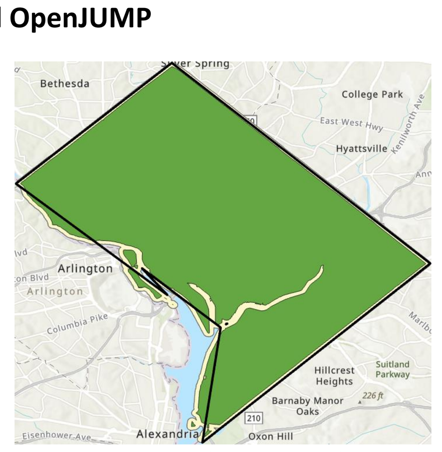
The Doulgas-Peuker algorithm is one of the first polygon simplification algorithms created and is currently the most commonly used in GIS. It is the fastest algorithm but also gives a coarse resolution. It is great for relatively small amounts of data compression and is best used when maintaining the integrity of the shape is not a concern and when high cartographic quality is not needed. The algorithm works by reducing the number of points through recursion to remove redundant vertices in order to smooth polylines.

This works by connecting the endpoints of a line with a trend line. A degree of ε is chosen as the tolerance, which defines the maximum perpendicular distance allowed between the original points and the simplified curve. Points are then removed if they are closer to the line than to the tolerance. The algorithm then recursively divides the line by the point farthest from the trend line, making two new trendlines, continuing the process until all the points within the tolerance are eliminated. This leaves only a subset of the points from the original polygon, which make up the new simplified polygon.

Polygon Simplification in ArcPro and OpenJUMP



Douglas-Peuker Polygon Simplification (Retain Critical Points) in ArcPro: Simplifies to 53 vertices



Douglas-Peuker Polygon Simplification on 200m buffer in ArcPro

Douglas-Peuker Preserve Topology Polygon Simplification in OpenJump: Simplifies to 77 vertices

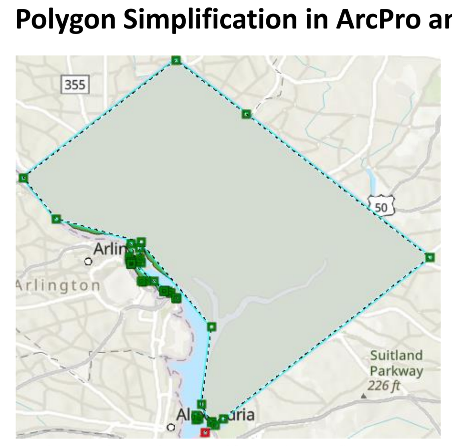
Douglas-Peuker Polygon Simplification in OpenJump: Simplifies to 13 vertices

Visvalingan-Whyatt

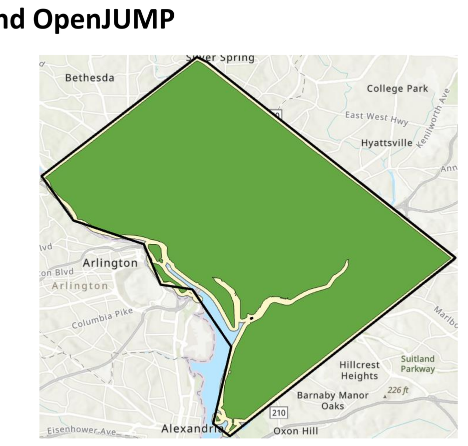
Polygon Simplification available in ArcPro, OpenJUMP, and 'EFFECTIVE_AREA' in Python

The Visvalingam-Whyatt algorithm works by connecting each point with its two neighbors forming a triangle. Based on the triangles, the least important point is removed first by eliminating points associated with the smallest triangles. This will have the smallest impact on the integrity of the original shape of the polygon. The process is iterative, so after each pass the triangles are recomputed for the remaining points allowing the process to repeat until there are no more triangles to remove within the tolerance. The square of the tolerance is the area that defines a significant triangle.

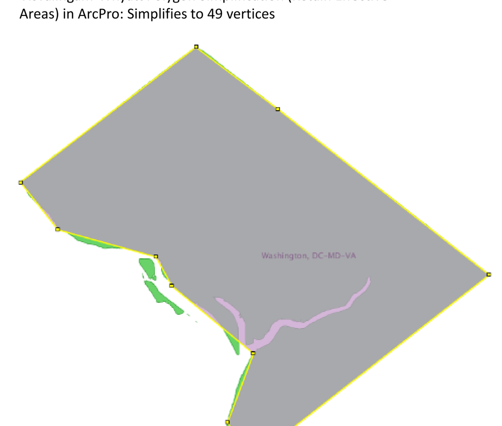
Polygon Simplification in ArcPro and OpenJUMP



Visvalingam-Whyatt Polygon Simplification (Retain Effective Areas) in ArcPro: Simplifies to 49 vertices



Visvalingam-Whyatt Polygon Simplification on a 200m buffer in ArcPro



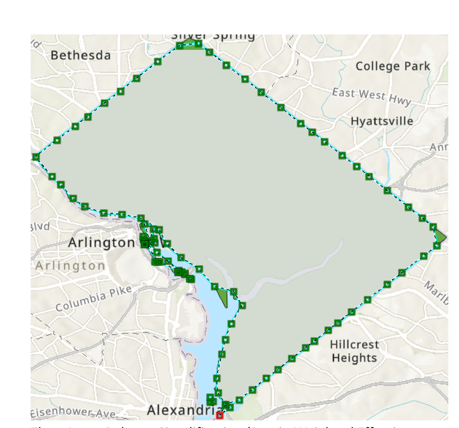
Visvalingam-Whyatt Polygon Simplification in OpenJump: Simplifies to 11 vertices

Zhou-Jones

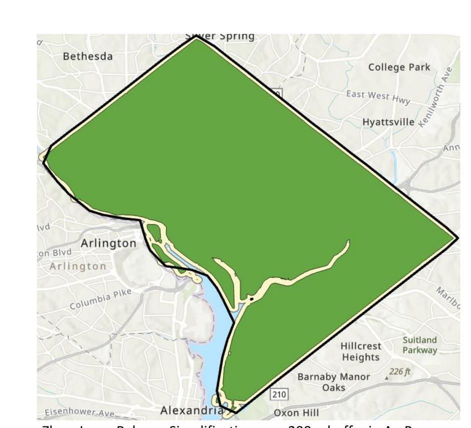
Polygon Simplification available in ArcPro and 'WEIGHTED_ALGORITHM' in Python

The Zhou-Jones algorithm works a little differently than the Visvalingam-Whyatt algorithm. While both use triangles of effective areas, Zhou-Jones weights them by a set of metrics that compares the flatness, skewness, and convexity of each area. Points are removed using the weighted areas as guides with the goal of retaining as much of the character of the original polygon. A significant triangle is recognized based on the square of the tolerance chosen. The more non-equilateral a triangle is, the higher weight the triangle is given, making it less likely to be eliminated. This is the reason why corners can end up removed with this algorithm.

Polygon Simplification in ArcPro



Zhou-Jones Polygon Simplification (Retain Weighted Effective Areas) in ArcPro: Simplifies to 101 vertices



Zhou-Jones Polygon Simplification on a 200m buffer in ArcPro

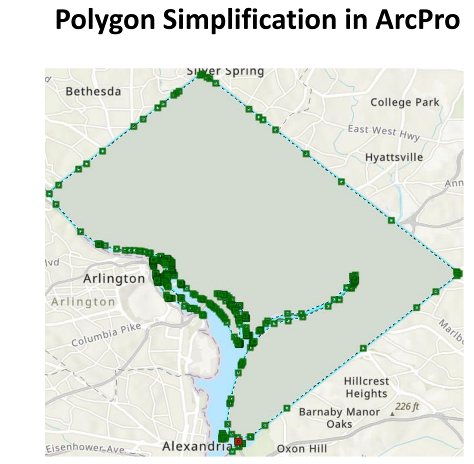
Wang-Mueller

Polygon Simplification available in ArcPro and 'BEND_SIMPLIFY' in Python

The Wang-Mueller algorithm is much more accurate in retaining the integrity of the original shape of the polygon. This is due to the fact that the algorithm uses shape recognition techniques to analyze the characteristics of bends. However, the algorithm does take a longer to run as does not reduce the number of points to the degree in which other methods do.

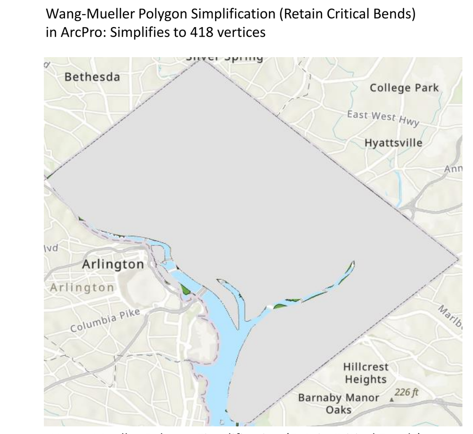
The algorithm works by identifying significant bends and removing the relative insignificant ones to simplify the polygon. The tolerance used is the diameter is a circle which is used to approximate significant bends. The reference half circle is used compare the geometric properties of each bend. Bends determined to be insignificant are replaced by the line connecting the endpoints of the bend. This is an iterative process where smaller bends that disappear form bigger bends, resulting in a line that maintain the integrity of the original shape more closely, thus giving better cartographic quality than other algorithms.

Polygon Simplification in ArcPro



Wang-Mueller Polygon Simplification (Retain Critical Bends) in ArcPro: Simplifies to 418 vertices

Wang-Mueller Polygon Simplification on a 200m buffer in ArcPro



Wang-Mueller Polygon Simplification (Retain Critical Bends) in ArcPro without vertices to allow for error to be seen