UNIVERSITY OF Hull

The Explorations in Digital Cartography Discussion Paper Series is a continuation of the CISRG Discussion Paper Series. This series is edited by:

Mahes Visvalingam
Reader Emeritus
Dept of Computer Science
University of Hull
HULL HU6 7RX, UK

Explorations in Digital Cartography

DISCUSSION PAPER 3

Testing Implementations of Visvalingam's Algorithm for Line Generalisation

by

Mahes Visvalingam

No part of this Discussion Paper should be copied, cited or distributed without written permission from the editor of the series.

Explorations in Digital Cartography Discussion Paper Series,
Department of Computer Science, University of Hull

List of publications

1.	Visvalingam, M and Whelan, J.C (November 2014) "Implications of Weighting Metrics for Line Generalisation with Visvalingam's Algorithm" https://hydra.hull.ac.uk/resources/hull:10064	24 pp
2.	Visvalingam, M (March 2015) "The Visvalingam algorithm: metrics, measures and heuristics" https://hydra.hull.ac.uk/resources/hull:10596	12 pp
3.	Visvalingam, M (May 2015) "Testing Implementations of Visvalingam's Algorithm for Line Generalisation"	16 pp

ABSTRACT

There are a growing number of open source and commercial implementations of the Visvalingam algorithm for line generalisation. The algorithm provides scope for implementation-specific interpretations, with different outcomes. This is inevitable and sometimes necessary and, they do not necessarily imply that an implementation is flawed. The only restriction is that the output must not be so inconsistent with the intent of the algorithm that it becomes unusable.

This paper provides some ideas, data and sample output to help users compare the output from their implementations with those produced by Visvalingam. This may help them ascertain whether some problems they may encounter appear to be specific to their implementation or whether they are a general feature of the algorithm. This paper assesses the utility and limitations of the Mapshaper options for Visvalingam's algorithm.

Similar, but not identical, depictions of coastlines are produced by Visvalingam's implementation and by Mapshaper. However, the programs produce very dissimilar output for the rectangular Koch island, also known as the quadratic Koch island - Mapshaper's output is unbalanced for both its Visvalingam and Douglas-Peucker options. This suggests that the problem, which is not immediately obvious, is in some function inherited by both options. Both programs produce near identical output when Mapshaper's *Visvalingam/weighted area* option was compared using coastlines. This suggests that the problem arises from Mapshaper's treatment of equal-valued metrics; this can be changed. Implementers and users may wish to use the data and methods given in this paper to test their own implementations if and when necessary.

CONTENTS

1. Introduction

Visvalingam and Whyatt (1993) used the Visvalingam algorithm for line generalisation to demonstrate that it was possible to automate caricatural generalisation. Visvalingam (2015) explained that the algorithm can be driven by any metric, and that the Effective Area (EA) she first proposed is a heuristic measure designed for facilitating research into caricatural and model-based generalisation is also useful for line simplification and typification. Heuristic measures are indicators and are not guaranteed to provide optimal solutions in all situations. This paper is not concerned with the properties of the algorithm or with metrics and measures. It focuses instead on whether implementations conform to the specification of the algorithm.

Although an implementation may not be entirely consistent with the specification of the algorithm, it can be appropriate for its intended purposes(s). Some students used partial implementations of the algorithm successfully in their undergraduate and postgraduate projects at the University of Hull (Visvalingam. 2015). However, implementations can also be flawed. Comments based on an incorrect implementation can bring an algorithm into disrepute, undermine confidence in its wider use and lead to its rejection. It is hoped that Visvalingam's algorithm will not suffer this fate.

There are a growing number of open source and commercial implementations of Visvalingam's algorithm; pointers to some of these are provided in this paper. Users may be quite happy with an implementation but they should check that it conforms to Visvalingam's specification before directing adverse comments at the algorithm itself.

Not all implementers have publicised their source code. So, this paper considers how black box testing may be used to make inferences from observations of their output. It uses black box testing to demonstrate how Bloch's (2015) Mapshaper v 0.2.19 implementation of the Visvalingam algorithm was validated; this produced the same output as Mapshaper v 0.2.0 (Bloch, 2014) when tested in April 2015. As Visvalingam (2015) pointed out, the algorithm has multiple applications and, as such, validation focuses on assessing whether Mapshaper, in its present form, can be used in all of them. It starts off by showing how two implementations may pick the same subsets of points from coastlines but in a different order giving rise to identical output at some levels of filtering and not others. Visvalingam (2015) explained why such variation is sometimes inevitable and permissible; and, why this does not necessarily imply the presence of an error. However, Mapshaper did not appear to pick the least important point on each iteration.

Next, fractal curves were run through Mapshaper. Only the output from levels 1 and 2 of the quadratic Koch curve are illustrated here. They show that Mapshaper outputs unbalanced and inappropriate simplifications for its ***Douglas-Peucker*** and its ***Visvalingam / effective area*** options. This can be caused by the knock-on effects of points being chosen arbitrarily from a candidate set of equal-valued effective areas. To test this, Mapshaper's ***Visvalingam / weighted area*** option was selected this differentiates between otherwise equal effective areas. As expected, Visvalingam's program and Mapshaper produced near identical results.

These observations suggest that Mapshaper is a useful tool for visual mapping at the levels of simplification and typification of coastlines, especially when used with the *weighted effective area* option. As noted in Visvalingam and Whelan (2014), this option was not intended for caricatural generalisation. As it stands, Mapshaper may not be as helpful as Visvalingam's implementation for the segmentation of in-line features for model-based generalisation (see Visvalingam, 2015). The data, methods and lines of reasoning adopted in this paper could help implementers and users wishing to test their own implementations, if and when necessary.

2. Background

The Visvalingam algorithm for polyline generalisation is very simple. *"It consists of repeated elimination of the point which is least significant in a given line and treating the remainder as the new input line."* Visvalingam (2015) described how it can be expressed in different but consistent ways to suit different circumstances and purposes. She noted why some expressions have limitations and explained why she provided the specification published in Visvalingam and Whyatt (1993); even though Whyatt (1991) only needed to implement a part of it for comparing the performance of generalisation algorithms using individual lines. The full specification enables the filtering of a set of lines on maps with multiple filters and provides clues for segmentation and modelling of lines.

The above Pseudocode includes some implementation-specific features. The algorithm itself does not specify how the importance of a point should be measured – this depends on the application. Any metric can be used, but Visvalingam (2015) explained why the Effective Area (EA) was chosen initially and why EA functions as a heuristic indicator. The EA starts off as the triangular areal displacement which would occur if the point was to be dropped; this is the calculated metric. Its value is changed if the condition in line 5 is true, which usually indicates the presence of a line configuration suggestive of a feature. Visvalingam (2015) demonstrated how without lines 5 and 6, lines filtered on the metric will not correspond to the rank order of the points. This modification of EA is quite important since it can lead to a cascade of points being eliminated with the same EA (or rank if needed) on thin elongated features. Please note that Visvalingam (2015) favoured the conditional operator (<=) in line 5 over the original (<) operator published in Visvalingam and Whyatt (1993).

Lines 5 to 7 in the Pseudocode are implementation specific and are not an integral part of the basic generalisation algorithm. However, they are essential for using different filter values for filtering a set of polylines and were used in several CISRG projects as recounted in Visvalingam (2015). A full analysis of the algorithm identifying opportunities for further research will be provided in a forthcoming paper.

```
Let previous = 0.0                                                      [1]
Calculate EA for all internal points of the input line                  [2]
While there are internal points {                                       [3]
  Find the point with the least EA                                      [4]
  if (EA of this point <= previous)  EA = previous                      [5]
     else previous = EA                                                 [6]
  Record the EA of this point and note its rank (adjusted if and when needed)  [7]
  Recalculate EA for the two neighbouring points                        [8]
}                                                                       [9]
```

Pseudocode: Original specification of the Visvalingam algorithm

3. Implementations of the Visvalingam Algorithm

There are a growing number of implementations of Visvalingam's algorithm – some available as open source software and others have been incorporated within commercial GIS and mapping software. Some of these are listed below.

- In response to popular request, Vivid Solutions (2001) posted an implementation on sourceforge.net.for its users.
- The Zhou and Jones (2004) implementation is used by the Ordnance Survey for generalising coastlines (see Revell, 2011). Ordnance Survey is now using the 1Spatial 1Generalise implementation. Zhou (2014) provided a link to the download site for his demonstrator program, which includes the Java source.
- Harrower and Bloch (2006) announced Mapshaper; For the latest version, see Bloch (2015).
- Bostock (2012) posted a demonstrator using his Javascript implementation.
- Davies (2012) modified Bostock's implementation to preserve topological relationships between polygons.
- Kaefer (2012) implemented the C++ version within Mapnik.
- Aisch (2012-14) generated compact SVGs using Visvalingam simplification in the free to use version of Kartograph.
- Frye (2013) illustrated the use of the algorithm to simplify and compress auto traced coastlines by NASA.
- Mapbox Studio (2014) is an open source desktop software for designing maps. It uses Mapnik, which includes the Visvalingam algorithm, for rendering maps. Mapbox is not entirely free and has raised substantial funding to use compete with Google Maps (see Kolodny, 2013).
- Gaborit (2014) provided a Python implementation of Bostock's code.
- Dufilie and Grinstein (2014) used the Visvalingam algorithm for progressive transmission of vector data in web applications.
- OSGeo.org (2015) has a thread on implementation of Visvalingam's algorithm within this Open Source Foundation.
- Others are using the algorithm in applications beyond cartography; for example, for simplifying data visualizations for display on mobile devices. (Daniel Cascais, personal communication, 2014).

Incorrect implementations of an algorithm can bring it into disrepute and undermine confidence in its wider use. This paper provides sufficient background, data and sample output to encourage others to test implementations, using at least the black-box approach adopted here.

Any implementation which meets a specified purpose is valid so long as it eliminates the least important point on each iteration. The following section shows that it is not easy to reach definitive conclusions with coastline data and that fractal test data are more discriminating.

4. Observations

At the start of the Visvalingam and Whelan (2014) project, Whelan downloaded the source of Mapshaper v 0.2.0 and checked it using data for the section of a road used by Visvalingam and Williamson (1995). He noticed some differences in the values for EA. Visvalingam found that the discrepancies tended to occur on curved sections of lines, such as at a roundabout and at filleted road junctions. At these places, Mapshaper was picking a different point to that selected by Visvalingam's implementation, especially where these two points had the same EA. This can happen for a variety of reasons as explained by Visvalingam and Whyatt (1991) with respect to the RDP algorithm. So, Mapshaper was used for comparing its weighted area option with Visvalingam's EAs.

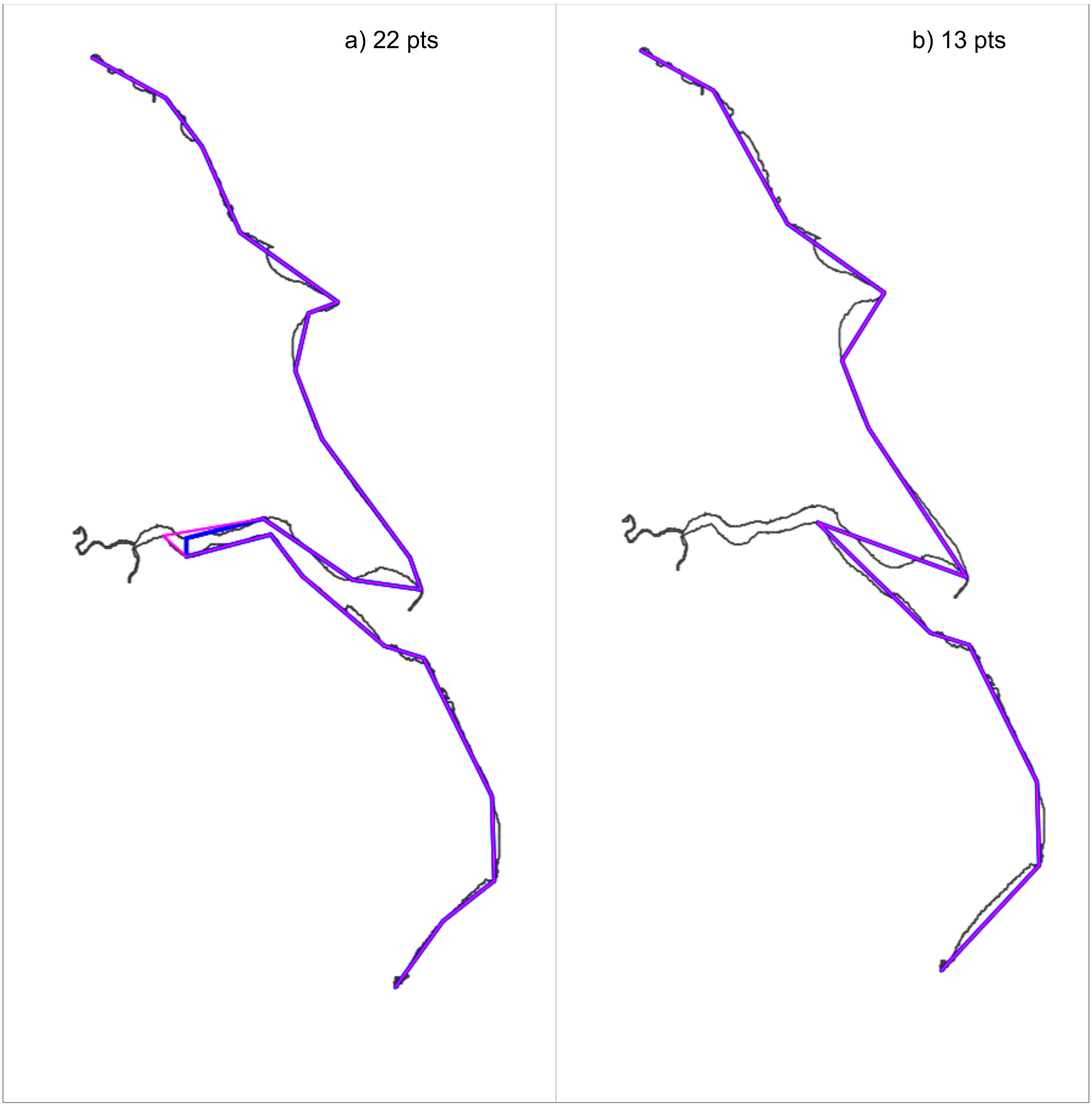
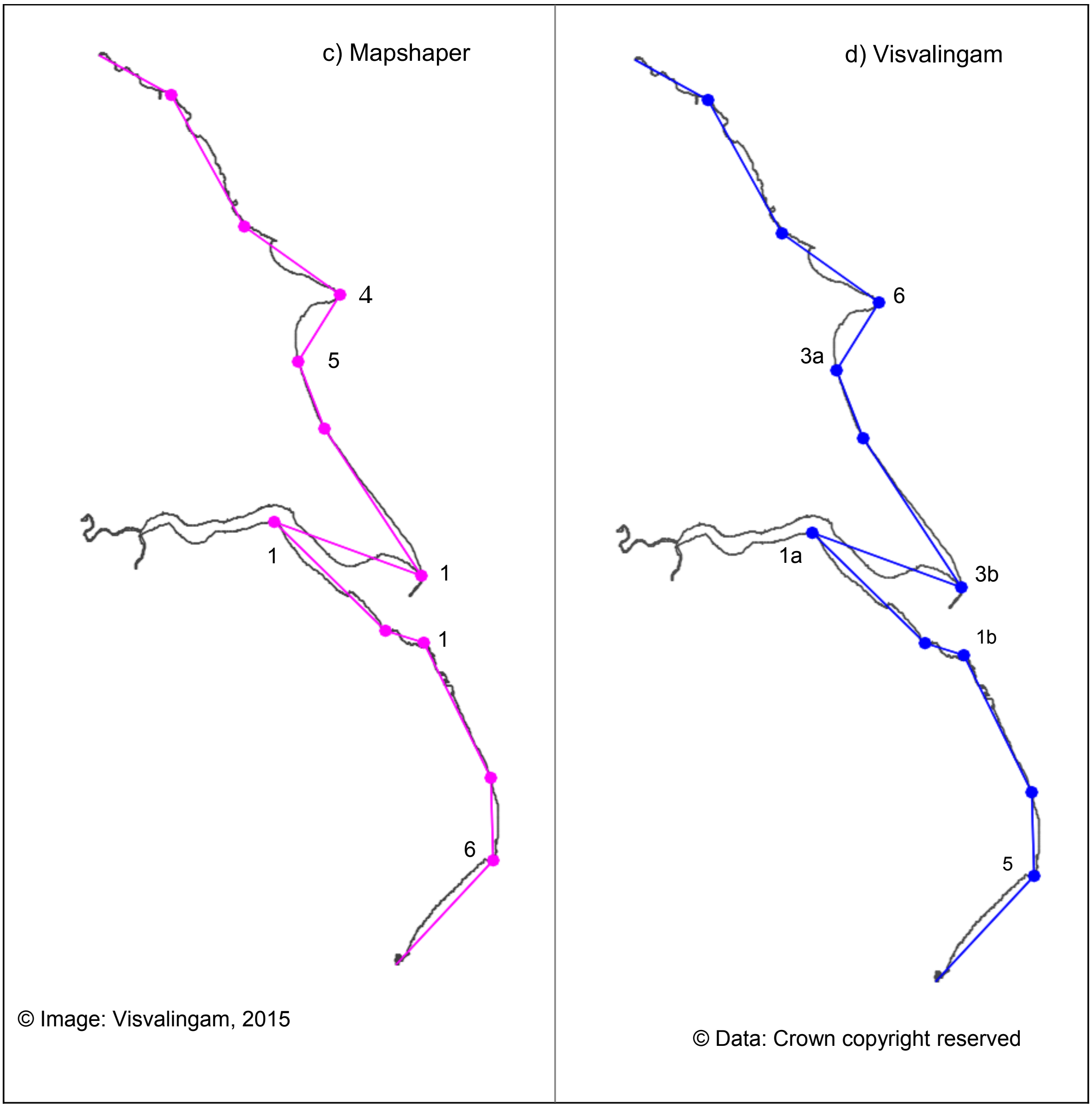


Figure 1: Comparison of points selected by Bloch (mauve) and Visvalingam (blue); (a) 22 points (0.9%); (b) 13 points (0.5%); order of removal of the last 6 points by c) Mapshaper v0.2.19; (d) Visvalingam

Visvalingam and Whelan (2014) used two stretches of coastlines, namely the SWURCC data and the OS VMD data, as described in their paper. These data sets can be downloaded from https://hydra.hull.ac.uk/resources/hull:9040, which provides information on the sources of these free data, maps and the co-ordinates of the coastlines.

4.1 1:50000 SWURCC Data

This was the main data set used by Visvalingam and Whelan (2014). Mapshaper produced comparable results to Visvalingam's implementation for this data set. Recent research suggests



that the differences seemed to be related to the special case (statements 5 and 6 in Pseudocode). When an EA was less than that of the previously eliminated point, Mapshaper did not always pick the point with minimum EA. It seemed to be picking the first point which fulfilled this condition. This can have a knock-on effect on the choice of some subsequent points. Different implementations of the RDP algorithm can produce different, but equally valid, results as observed and explained by Visvalingam and Whyatt (1990).

For reasons of consistency, Visvalingam and Whelan (2014) used Mapshaper v 0.2.0 to compare maps produced with the standard weight of 1 and Bloch's weighted EAs. The subsets of points drawn for a given percentage of points were very similar and often identical. In Figure 1a, there is a difference of just one point in the 0.9% of points retained. Mapshaper picks a point which gives a better shape, while Visvalingam's implementation picks a point which produces the chopped effect discussed in Visvalingam and Whelan (2014). Both implementations produce the same 0.5% subset of 13 points in Figure 1b. Stepwise visualization of the elimination of the points in Figure 1b showed that the two implementations were eliminating points in a different order after

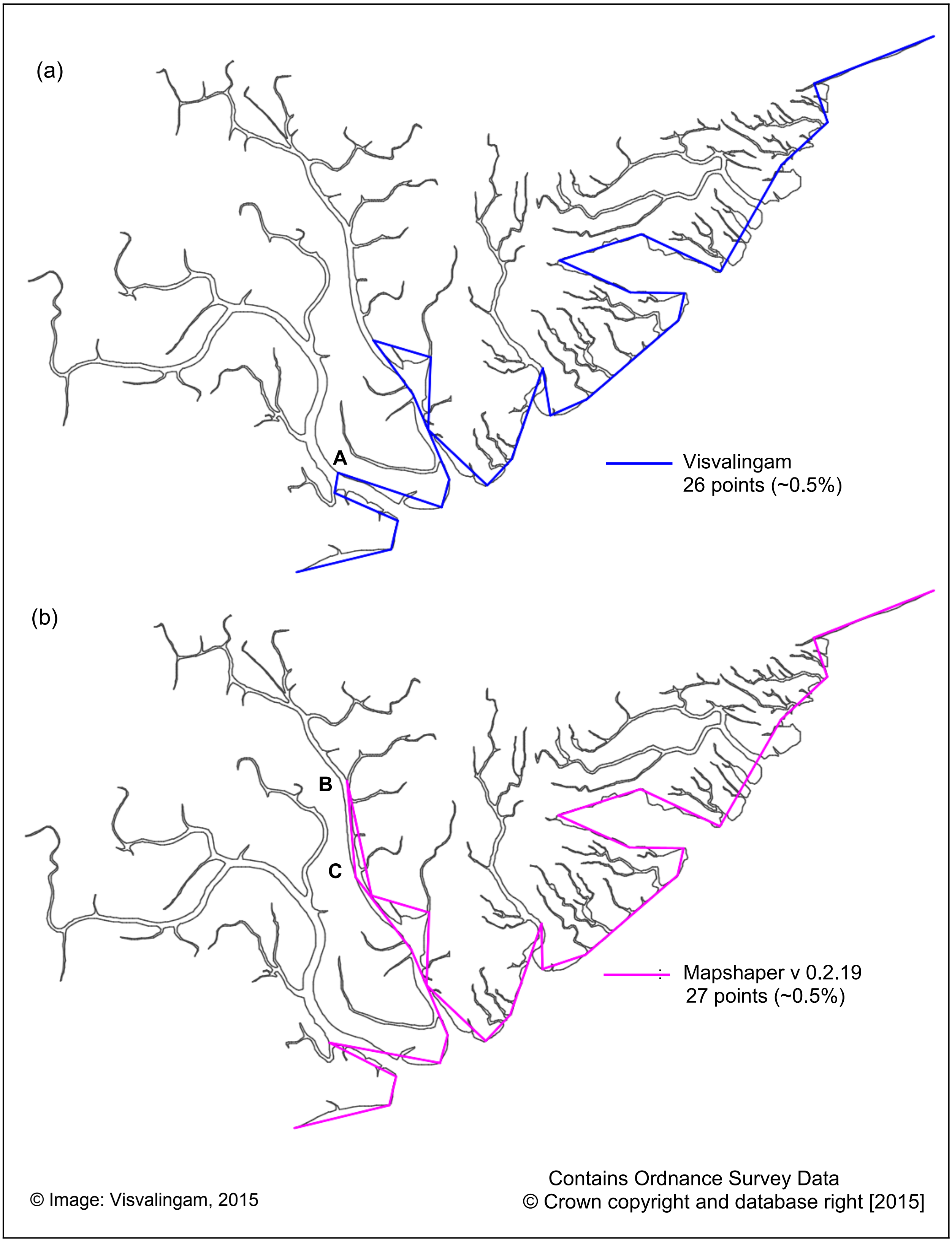
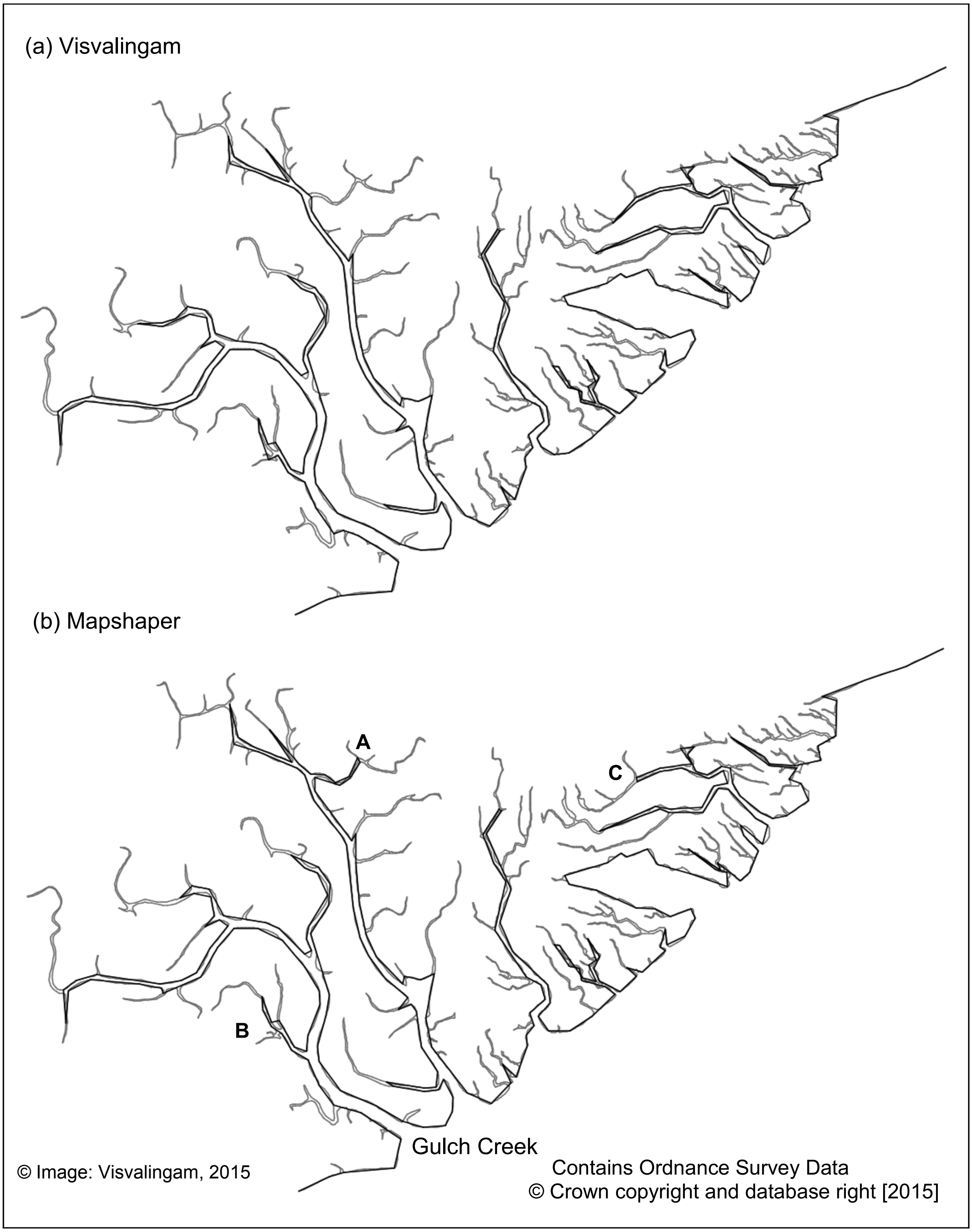

(a)
A
Visvalingam
26 points (~0.5%)
(b)
B
C
Mapshaper v 0.2.19
27 points (~0.5%)
© Image: Visvalingam, 2015
Contains Ordnance Survey Data
© Crown copyright and database right [2015]

Figure 2 : Filtering by (a) Visvalingam's implementation and (b) Mapshaper



© Image: Visvalingam, 2015

Contains Ordnance Survey Data
© Crown copyright and database right [2015]

Figure 3 : Five percent of points retained by (a) Visvalingam and (b) by Mapshaper

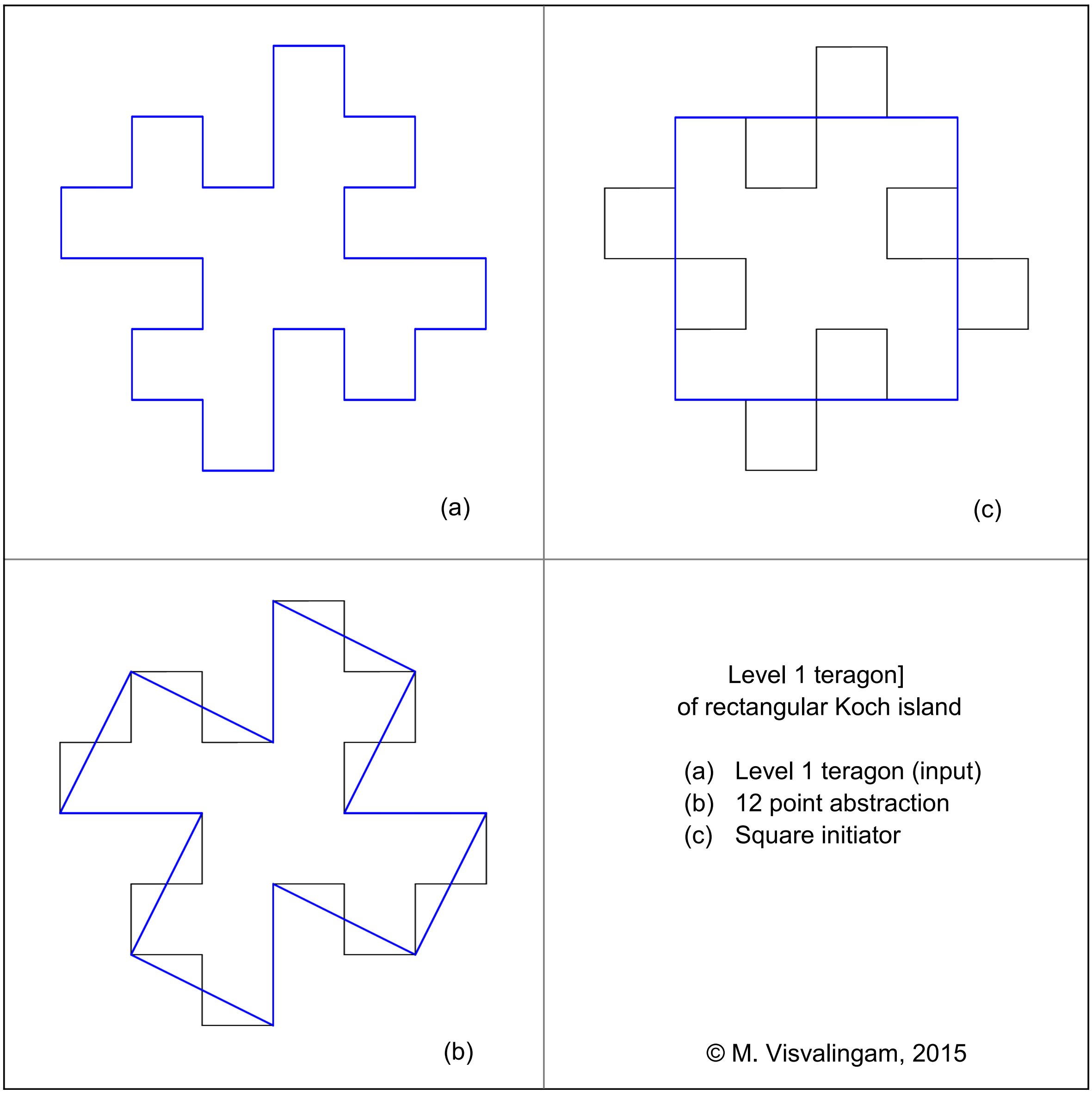


Figure 4: Visvalingam's implementation retains the 4-fold symmetry of the Koch island

encountering a special case (statement 5 in Pseudocode). Figures 1c & 1d show the last six points to be eliminated in their order of removal.

4.2 1:25000 OS VectorMap® District data (OS VMD Data)

This data related to an area known as The Scalp in Lincolnshire, which consists of wetlands drained by a complex of meandering creeks. Visvalingam and Whelan (2014) only compared output without and with weighting of EA using maps with 1.4% of points or more. Figure 2 shows the lack of correspondence between Mapshaper and Visvalingam's output when only 0.5% of points were retained. Visvalingam introduces A in Figure 2a before B & C in Figure 2b. At some levels of filtering, Mapshaper produces better results. In Figure 3, most of the retained features are remarkably similar – but there are some notable exceptions. The depiction of tributaries A and B by Mapshaper look more appropriate, even if C looks chopped. Gulch Creek and the trident shape created by the retention of A, make the streams instantly recognizable. This seemed to suggest that there may be a problem with Visvalingam's implementation.

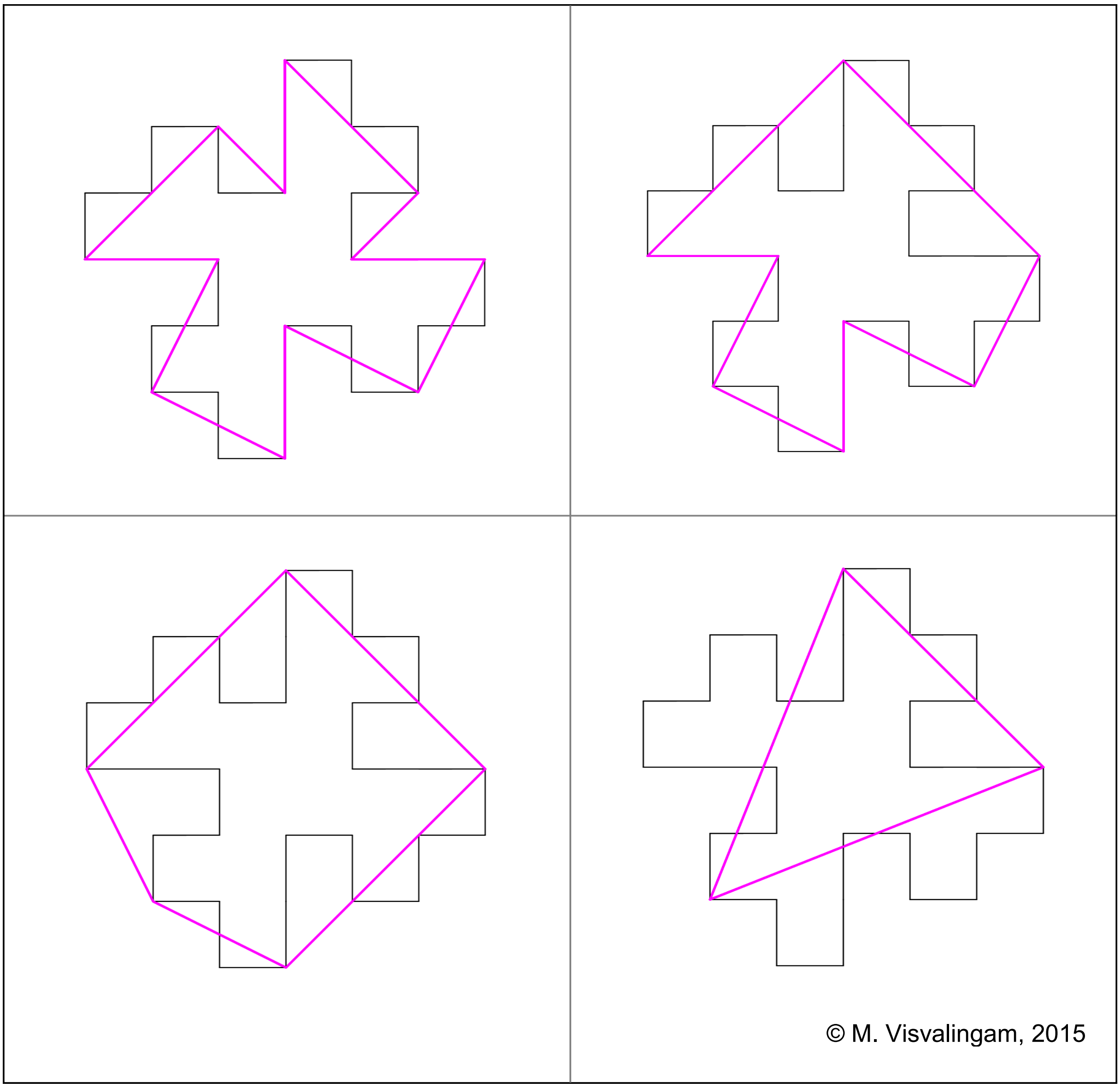


Figure 5 : Sample output from Mapshaper v 0.2.19

The output for coastlines showed that differences tended to occur a) when there were two or more candidate points with equal EA, and b) when the special case (statement 5 in Pseudocode) was triggered. The impact of equal-valued EAs on Mapshaper was investigated next using fractals. Only output for levels 1 and 2 of the quadratic Koch island are shown here.

4.3 Using fractals as test data

It is possible to abstract a range of unexpected patterns from even the level 1 teragon of the rectangular (quadratic) Koch island as demonstrated by Visvalingam and Brown (their Figure 2), not just by using different algorithms, but also by driving Visvalingam's algorithm with different metrics and in different directions. However, of those metrics tested with Visvalingam's algorithm, only EA was able to recover the original square initiator for teragons of orders 1 to 3 of the rectangular Koch curve.

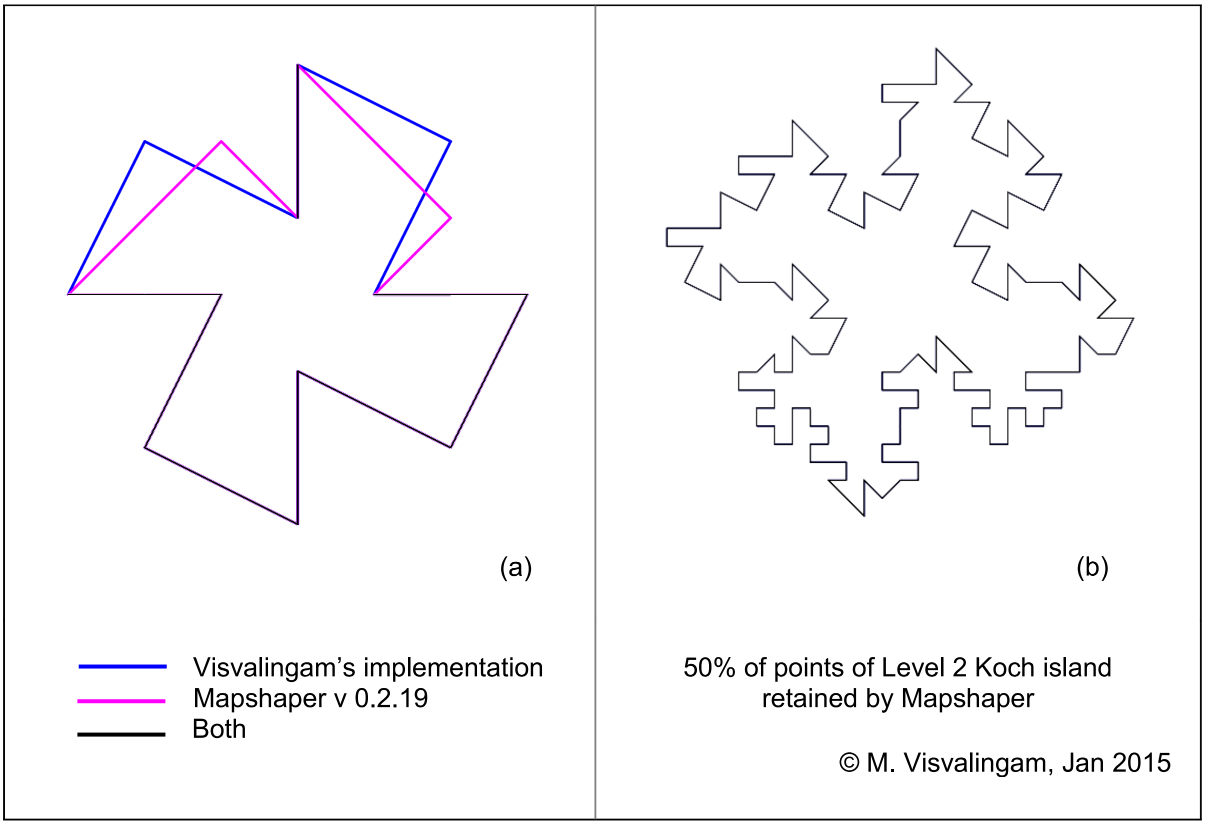

(a)
(b)
Visvalingam's implementation
Mapshaper v 0.2.19
Both
50% of points of Level 2 Koch island
retained by Mapshaper
© M. Visvalingam, Jan 2015

Figure 6 : Comparison of figures abstracted by Visvalingam's implementation and Mapshaper

Figure 4 shows the teragon and the simplifications produced by Visvalingam's program, using EA. The data for the teragon is provided in the Appendix. As pointed out by Visvalingam and Brown (p 164 -165), Visvalingam's algorithm retains the four-fold symmetry in the teragon and emulates the give-and-take rule used in manual cartography (Maling, 1989). They illustrated how rounding errors in implementation and the use of inappropriate start/end points can lead to a loss of symmetry.

Visvalingam and Herbert (1999) used coastlines and the quadratic Koch data to demonstrate that there were problems with the Arc/Info implementation of the Bendsimplify algorithm, which uses Visvalingam's idea of iterative elimination to remove bends instead of individual points..

Bloch's implementation of Visvalingam's algorithm produced rather unexpected results (see Figure 5). It was unable to retain the symmetry of the Koch island. Visvalingam's implementation only outputs one figure between the teragon and the initiator. Mapshaper outputs several but does not recover the initiator (Figure 5 only shows some of the intervening simplifications).

Mapshaper mimics the give-and-take rule in the bottom half of Figure 5a, but produces a different simplification in the top half of the figure; Figure 6a shows the difference. This inconsistency results in unbalanced and unacceptable results on further simplification (5b – c).

Visvalingam and Brown (1999, Figure 3a) produced symmetric generalisations of the level 2 Koch island as well. Mapshaper produced very unbalanced output from the same data, and it was difficult to figure out the cause of this problem. A screen image of Mapshaper's output with 50% of points filtered from the Level 2 Koch island is shown in Figure 6b.

Davies (2014) used a pair of quadratic Koch islands to demonstrate his approach to preserving the topology during Visvalingam simplification. His output is also unbalanced. Visvalingam and Brown (1999) noted that like the RDP algorithm, the Visvalingam algorithm is also sensitive to start and end points. The unbalanced output may partly be the result of the insertion of topological nodes to split the polygons into three polylines. However, this does not fully explain the lack of symmetry at even low levels of simplification.

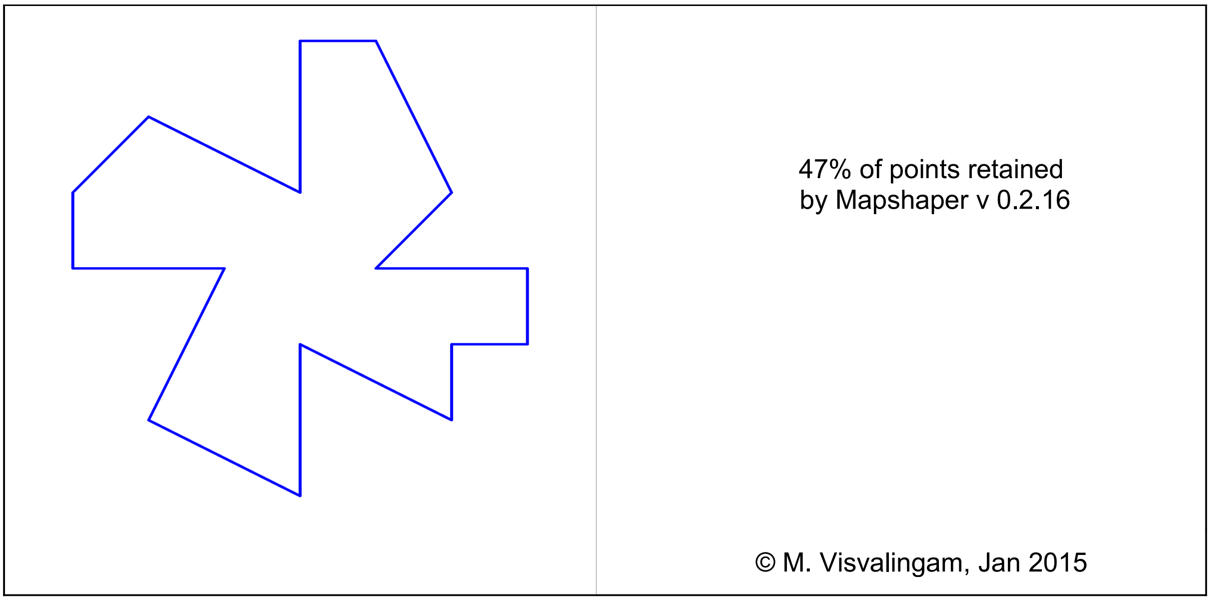


Figure 7 : Mapshaper's output for the RDP algorithm

The Koch data were run through Mapshaper's ***Douglas-Peucker*** option. Philip Wade's original Fortran programme (listed in Whyatt and Wade, 1988) produced 2 sets of five symmetrical figures (see Figure 2 in Visvalingam and Brown, 1999). Mapshaper produces many more unbalanced figures, including that in Figure 7, which shows varying levels of detail on different wings of the figure.

4.4 Comparison of weighted EAs

Despite the above observations, Mapshaper is a valuable tool for the following reasons. The problem seems to be due to arbitrary points being picked for removal from a set of equal-valued candidates with consequent effects on neighbouring points. If this is indeed the cause of the unbalanced output, we can draw two conclusions. First, the problem can be fixed. Second, Mapshaper has been largely targeted at simplifying coastlines. In this context, this problem can be ignored as a red herring for the following reason. As Visvalingam and Whyatt (1990) pointed out, cartographic data are inexact and representative. Given that digitising errors are much greater than rounding errors and that data are rounded to the nearest precision for dissemination, they can make prescriptive stipulations of how to choose a point from a set of equal-valued candidates somewhat pedantic,.

The author compared the output of her program with Mapshaper's using the ***Visvalingam / weighted* area** option. Equal EAs are very likely to become unequal after weighting. Unfortunately, Mapshaper only exports the co-ordinates for the filtered subset, which means that the weighted EAs could not be numerically compared using black box testing by the author. Also, it is difficult to step through Mapshaper to include/remove a point at a time since the percentage specified for selection is rounded to one decimal point. Initially, the output from the two implementations were not similar. After unsuccessfully testing several possible interpretations of her algorithm, Visvalingam's program was modified to eliminate the first point with an EA <= previously eliminated value instead of eliminating the point with the lowest EA. Visual checks suggest that both programs were producing very similar results, and removing points in the same order. Others may wish to check this. This lends further support to the belief that the unbalanced output is due to Mapshaper's processing of equal-valued EAs and its deviant interpretation of Visvalingam's algorithm.

5. Discussion and Conclusion

Visvalingam (2015) explained why her specification of the algorithm was not overly prescriptive. Some variation in implementation is inevitable and is sometimes necessary. Visvalingam and Whyatt (1990) had explained why the results from the RDP algorithm could vary between implementations and on different computers depending on the specific test condition used, the direction of parsing, machine rounding errors and digitising errors.

Visvalingam and Whelan therefore were not over-concerned that the output of Mapshaper v 0.2.0 was not identical to theirs. As illustrated in Figure 1, the output was not significantly different even in caricatures of coastlines. More recent investigations reveal that Mapshaper does not always remove the least important point when re-calculated EAs are less than that of the previously eliminated point. This was tested and did not fully explain the differences in output.

The rectangular Koch island, previously used by Visvalingam and Brown (1999), revealed that Mapshaper produces unbalanced simplifications for both the Visvalingam and the RDP options; Visvalingam's program preserves the symmetry of figures in scale-related steps, except when the start/end points were inappropriate. Mapshaper did not retain the four-fold symmetry in its filtering of the Koch island. Filtering of the quadratic Koch island by Davies (2014) also shows a loss of symmetry even at fairly low levels of simplification. The output for the fractal data lends support to the belief that the problem is in how Mapshaper processes equal valued EAs. Mapshaper's weighted area option produces comparable output since weighting differentiates between the original equal-valued EAs.

Fractals are by definition self-similar and any simplification has to retain the symmetry, which reflects their self-similarity. So, as it stands, Mapshaper is not appropriate for simplifying fractals nor is it suitable for applications outside of cartography, which require that simplified depictions of engineered components have to reflect the symmetry and balance in their original design. The distortion of shape will also impede the segmentation of in-line features and parts for model-based generalisation, which has applications in pattern recognition and not just cartography.

Generalised depictions are inevitably subjective in manual cartography and some variation in digital generalisations is equally acceptable. The processes involved in the production of cartographic data introduce a margin of error and data are rounded to a stated precision for distribution at different scales. So, equal-valued metrics can be regarded as fortuitous and the choice of the point for elimination can vary, not just with test conditions but also with the order in which data are presented as explained by Visvalingam (2015). She stated that *All versions should output similar, even if not identical, generalisations of coastlines. The only restriction is that the output must not be so inconsistent with the intent of the algorithm that it looks inappropriate for its intended purpose.*

Despite the unbalanced and inappropriate results for fractals, Mapshaper is a useful tool for coastline simplification and typification. It has been used by some cartographers working within the New York Times and by others recommending it in several online forums. It inspired the demonstrator by Bostock (2014), which has in turn inspired Davies and several others. Also, Mapshaper's weighted area option produces aesthetically pleasing simplifications.

There are a growing number of implementations of the Visvalingam algorithm, some of which have been noted in this paper. They too are becoming popular and need to be independently evaluated. Comments based on incorrect implementations can bring algorithms into disrepute. This paper has suggested how implementers and users could assess whether their implementations conform to Visvalingam's specification to help them and their readers assess whether their comments apply to the algorithm, a specific implementation or even to the heuristic being used.

Acknowledgements

The data used in this project are covered by Crown Copyright. The SWURCC data were provided for research purposes and contains public sector information licensed under the Open Government Licence v2.0. The data for The Scalp is Ordnance Survey data © Crown copyright and database right [2015]. The author is grateful to Patrick Revell and Sheng Zhou of Ordnance Survey, and to Nicolas Regnauld of 1Spatial, for their information on the use of Visvalingam's algorithm in the OS VMD Data prototype and more recent production systems.

This unfunded project by Mahes Visvalingam (a retired, independent research worker - IRW) relied on input from many others. Thanks to Matthew Bloch for freezing Mapshaper v 0.2.0 and for continuing to evolve and maintain his unfunded web service; Chris Brown who generated the co-ordinates for the various levels of teragons using the *lsys* program written by John Leech (published in Prusinkiewicz and Lindenmayer, 1990); and, to John Griffiths (a former retired colleague and fellow IRW) for writing an interactive graphics program, PolylineViewer, for visualising the output of the author's program in camaraderie and friendship. Special thanks to Philip Wade for his innovative ideas and programs, which were used in several subsequent CISRG research projects undertaken by this author and her students; and to John Whelan whose contributions to Visvalingam and Whelan (2014) provided the impetus for this research.

References

Aisch, G (2012-14) "What is Kartograph?" http://kartograph.org/ [Accessed Jan 2015]

Bloch, M (2014) An updated version of the original Mapshaper tool v 0.2.0
http://mapshaper.org/Visvalingam2014 [Accessed 2014]

Bloch, M (2015) Mapshaper v. 0.2.19, http://mapshaper.org/ [Accessed April 2015]

Bostock (2012) Line Simplification – demonstrator http://bost.ocks.org/mike/simplify/
Javascript source: http://bost.ocks.org/mike/simplify/simplify.js
Discussion: https://groups.google.com/forum/#!topic/d3-js/uPOao98T9mg] and
https://news.ycombinator.com/item?id=4055445 [Accessed Jan 2015]

CISRG (2015) **Cartographic Information Systems Research group (CISRG) Discussion Papers Series,** University of Hull; **T**hese contain preprints of our journal papers and are available at: https://hydra.hull.ac.uk/resources?f%5Bcreator_name_ssim%5D%5B%5D=Visvalingam%2C+Maheswari

Davies, J (2012) "Quadratic Koch Island Simplification",
http://www.jasondavies.com/simplify/koch/ [Accessed Jan 2015]

Douglas, D H and Peucker, T K (1973) “Algorithms for the reduction of the Number of Points Required to Represent a Digitised Line or its Caricature”, **The Canadian cartographer 10 (2) : 112 – 122**

Dufilie, A and Grinstein, G (2014) “Feathered Tiles with Uniform Payload Size for Progressive Transmission of Vector Data”, in (Pfozer, D and Li, K-J, eds.) **Web and Wireless Geographical Information Systems**. Springer Berlin Heidelberg, 2014. 19-35. http://www.cs.uml.edu/~adufilie/feathered-tiles.pdf

Frye, S (2013) “NASA’s Earth Observing One (EO1) Satellite Advanced Land Imager Observations and Products for Disaster Management”, Presented at the **15th Congres de l’AQT** 26 September 2013 http://www.laqt.org/congres2013/presentations/Frye.pdf (stuart.frye@nasa.gov)

Gaborit, M (2014) visvalingam 0.3.0, https://pypi.python.org/pypi/visvalingam/0.3.0 The source is at: https://github.com/Matael/Visvalingam-Wyatt [Accessed Jan 2015]

Harrower, M and Bloch, M (2006) “MapShaper.org: A Map Generalization Web Service”, **IEEE Computer Graphics and Applications 26** (4): 22-27; accessible as Bloch, M and Harrower, M (2006) “MapShaper.org: A Map Generalization Web Service”, **Proc AutoCarto 2006.** www.cartogis.org/docs/proceedings/2006/bloch_harrower.pdf [Accessed Jan 2015]

Kaefer, K (2012) " Simplification of geometries using the Visvalingam-Whyatt algorithm", https://github.com/mapnik/mapnik/pull/1385 . See also http://mapnik.org/ and http://dbsgeo.com/sotm-pdx/ {Accessed May 2015]

Kolodny, L (2013) Google Maps Open-Source Challenger Mapbox Raises $10M From Foundry Group http://blogs.wsj.com/venturecapital/2013/10/16/google-maps-open-source-challenger-mapbox-raises-10m-from-foundry-group/ [Accessed Jan 2015]

Maling, D H (1989) **Measurements from maps: principles and methods of cartometry**, Pergamon Press, Oxford

Mapbox Studio (2014) Mapbox/carto documentation https://github.com/mapbox/carto/blob/master/docs/latest.md#polygon-simplify-algorithm-keyword [Last accessed Jan 2015]

OSGeo-org (2015) Thread on implementation of Visvalingam’s algorithm Geo, http://osgeo-org.1560.x6.nabble.com/Sprinting-td5180250.html#a5180696 within the Open Source Geospatial Foundation http://www.osgeo.org/content/foundation/about.html [Last accessed Jan 2015]

Prusinkiewicz, P and Lindenmayer, A (1990) **The Algorithmic Beauty of Plants**, Springer, New York.

Ramer, U (1972) “An Iterative Procedure for the Polygonal Approximation of Planar Curves”, **Computer Graphics and Image Processing 1,** 244 – 256.

Revell, P, Regnauld, N, and Bulbrooke, G (2011) OS VectorMap® District: Automated Generalisation, Text Placement and Conflation in Making Public Data Public, in: **Proceedings of 25th International Cartographic Conference**. Paris, France. 3-8 July 2011. http://icaci.org/documents/ICC_proceedings/ICC2011/Oral%20Presentations%20PDF/D3-Generalisation/CO-358.pdf [Accessed May 2014]

Visvalingam, M (2015) “The Visvalingam Algorithm - Metrics, Measures and Heuristics” , **Explorations in Digital Cartography Discussion Paper 2**, University of Hull, 11 pp

Visvalingam, M and Brown, C I, (1999) “The Deconstruction of Teragons into Decogons”, **Computers & Graphics 23** (1), 155 – 167

Visvalingam, M and Herbert, S P (1999) "A Computer Science Perspective on the Bendsimplify Algorithm", **Cartography and Geographical Information Systems**, 253 - 270.

Visvalingam, M and Whelan, J C (2014) “Implications of Weighting Metrics for Line Generalisation with Visvalingam’s Algorithm”, **Explorations in Digital Cartography Discussion Paper 1**, University of Hull, 24 pp. https://hydra.hull.ac.uk/resources/hull:10064

Visvalingam, M and Whyatt, J D (1990) “Cartographic Algorithms : problems of implementation and evaluation and the Impact of digitising errors”, **Computer Graphics Forum 10 (3),** 225 – 235. Preprint at: https://hydra.hull.ac.uk/resources/hull:8336

Visvalingam, M and Whyatt J D (1993) "Line Generalisation by Repeated Elimination of Points**", Cartographic J., 30** (1), 46 – 51 DOI: http://dx.doi.org/10.1179/000870493786962263

Visvalingam, M and Williamson P (1995) "Simplification and Generalisation of Large-Scale Data for Roads : a comparison of two filtering algorithms**", Cartography and Geographical Information Systems 22** (4), 264 – 275 DOI:10.1559/152304095782540249 .
Vivid Solutions (2001) VWSimplifier.java http://sourceforge.net/p/jts-topo-suite/code/HEAD/tree/trunk/jts/java/src/com/vividsolutions/jts/simplify/VWSimplifier.java

Whyatt, J D (1991) "Visualisation and Re- evaluation of Line Simplification Algorithms" , Unpublished PhD Thesis, University of Hull, 259 pp

Whyatt, J.D., and Wade, P.R. (1988) The Douglas Peucker line simplification algorithm", **Society of University Cartographers Bulletin 22 (1)** 17-25.

Zhou, S (2014) “Partition and conquer: Improving WEA-based coastline generalisation” GISRUK 2014, Glasgow, 16-18 April 2014:
http://www.researchgate.net/publication/261723727_Partition_and_conquer_Improving_WEA-based_coastline_generalisation [Accessed 2014].
The original demonstrator and the Java source code is at:
http://www.researchgate.net/publication/262487021_WEADemo-original [Accessed Aug 2014]

Zhou, S & Jones, C B (2004) Shape-aware line generalisation with weighted effective area, Presented at: 11th International Symposium on Spatial Data Handling, Leicester, UK, 23 - 25 August 2004. Published in: Fisher, P. ed. **Developments in Spatial Data Handling**. Berlin: Springer, pp. 369-380. (10.1007/3-540-26772-7_28).
http://users.cs.cf.ac.uk/C.B.Jones/Zhou_JonesSDH04.pdf

Appendix : Output from Visvalingam’s program

X	Y	EA
0	0	9.00
0	1	0.50
1	1	-0.50
1	2	-2.00
0	2	0.00
-1	2	2.00
-1	3	0.50
0	3	-0.50
0	4	8.00
1	4	0.50
1	3	-0.50
2	3	-2.00
2	4	0.00
2	5	2.00
3	5	0.50
3	4	-0.50
4	4	8.00
4	3	0.50
3	3	-0.50
3	2	-2.00
4	2	0.00
5	2	2.00
5	1	0.50
4	1	-0.50
4	0	8.00
3	0	0.50
3	1	-0.50
2	1	-2.00
2	0	0.00
2	-1	2.00
1	-1	0.50
1	0	-0.50
0	0	9.00

KEY

X & Y Co-ordinates
EA Effective Area