

Partition and Conquer: Improving WEA-Based Coastline Generalisation

Sheng Zhou

Research, Ordnance Survey Great Britain
Adanac Drive, Southampton, SO16 0AS, UK
Telephone: (+44) 23 8005 5000
Fax: (+44) 23 8005 6159
Sheng.Zhou@OrdnanceSurvey.co.uk
<http://www.ordnancesurvey.co.uk/>

1. Introduction

Line simplification and line generalisation are two of the major map generalisation processes. Line simplification algorithms such as the Ramer-Douglas-Peucker (RDP) algorithm (Ramer 1972; Douglas and Peucker 1973) are often applied in model generalisation to remove redundant vertices and preserve maximum information in generated DLMs at smaller scales. On the other hand, line generalisation algorithms such as the Visvalingam-Whyatt (VW) algorithm (Visvalingam and Whyatt 1993) are normally used in cartographic generalisation to modify the profile of feature geometry in order to create alternative depictions that are appropriate for particular map products.

The VW algorithm computes an effective area (EA) value for each vertex of a line via an iterative process. The initial EA value of vertex is the area of the EA triangle (formed by the vertex and its two adjacent vertices in the line). After the EA is computed for all vertices, the vertex with the smallest EA is removed and the EA for its two adjacent vertices is re-computed (if the new EA is smaller, the original EA is retained). This process is repeated until only three vertices are left in the line (endpoints in an open line are normally assigned with the largest EA value obtained in the process). After the process, the vertices of a line may be filtered by different query EA values to retrieve the representations of the line at different levels of details.

The concept of EA was generalised and extended to weighted effective area (WEA) (Zhou and Jones 2004), which also takes into consideration the shape of the EA triangles (how flat is it? Is it skewed towards one side?), and the context of the lines (relative to the line's orientation, is it a convex or concave vertex?):

$$WEA = EA * W_{flat} * W_{skew} * W_{convex} \quad (1)$$

Here W_{Flat} , W_{Skew} and W_{Convex} are the flatness, skewness and convexity weight functions respectively.

The WEA scheme can produce both simplification and generalisation effects by adopting different weight functions. The WEA algorithm has been successfully applied to generalise coastline data in making Ordnance Survey's VectorMap® District product, which is freely available as a part of the OS OpenData (Chilton 2010).

Despite VW/WEA algorithms' success in real world applications, many issues remain to be addressed. While the issue of the value selection of some weights was studied using fuzzy inference system and the ANN (Olszewski, Fiedukowicz and Pillich-Kolpinska 2013), how the weight function parameter values affects the generalisation effects is still hard to quantify. Also, when the VW algorithm is applied to the entire coastline, those thin long details such as spit are tended to be “chopped back” (figure 1), which is often undesirable. In WEA the weighted generalisation filters often further enhance this “chopping” effect.



Figure 1. The chopping effect of VW algorithm before (left) and after using partition.

In this paper we report the initial results of our effort to address this “chopping” effect by partitioning a line into multiple sections and processing individual sections separately.

2. Improving WEA Generalisation by Partitioning

Landform details (e.g. spit, bay and tombolo) on a line may be defined by the key vertices constraining the profile of the details. A detail is retained if its key vertices are preserved during the generalisation process. If we are able to identify these key vertices, we may partition a line into multiple sections and line generalisation algorithms may subsequently be applied to these individual sections between key vertices, probably using adaptive parameter values as well.

Recognition of landform details on a line is often subjective and the results are normally not unique. For this proof-of-concept pilot study, we have chosen the RDP algorithm as a simple yet effective method to identify landform details and partition coastlines, for it is widely accepted that the RDP algorithm is able to retain the general profile of a geometric object while removing most points from it.

Firstly, we compute the RDP tolerance value for each vertex in a top-down manner. A technique of “tolerance promotion” (Zhou and Jones 2001a) is used to guarantee tolerance values does not increase along any path downwards, i.e. if the RDP tolerance of a vertex at a lower level of the vertex hierarchy is larger than that of its parent vertex, the larger tolerance value will be propagated upwards. Subsequently, for a given query tolerance value, any vertices with a tolerance not smaller than the query value will be marked as section boundary points.

The partitions under different tolerance value are presented in figure 2 (the line extends 973m by 298m). 4, 6, 10 and 21 sections are generated respectively using RDP tolerance of 250, 100, 25 and 10m.

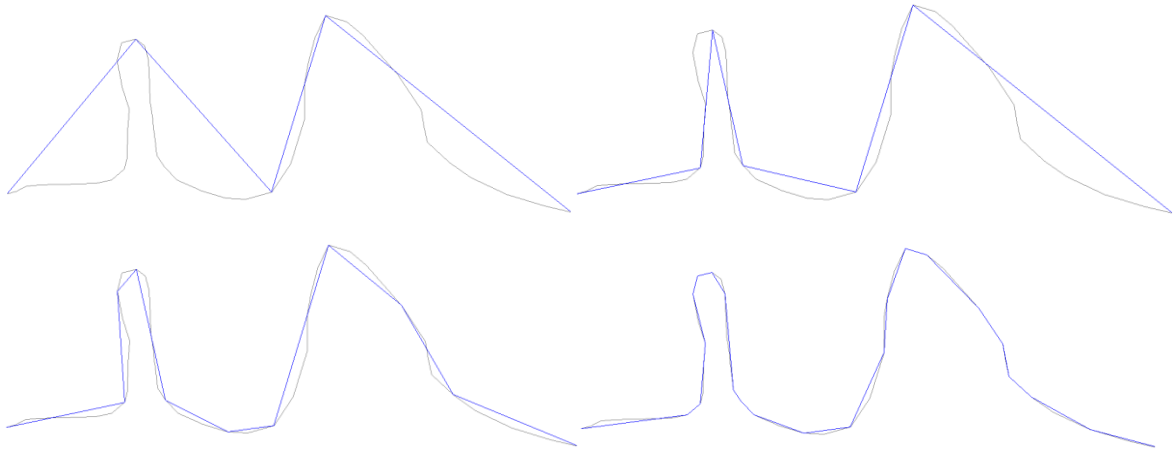


Figure 2. RDP-based Line partition (each segment of the blue line defines a partition).

3. Experiment Results

3.1 WEA computation under a partition scheme

After the partition is created, a two-round WEA computation is performed. Firstly, for each section, the WEA is computed for all vertices in the section. Subsequently, all section boundary vertices are selected (which is effectively a RDP-simplified version of the line) and the WEA value is re-computed for these vertices. If the new WEA value for a vertex computed in the second round is larger, the new value will replace the old one.

The following flatness weight function is used in current experiment:

$$WEA = EA * (ks * W/H)^k \quad (2)$$

Here W and H are the width and height of the EA triangle. 1.0 is used for the shape description parameters ks and k . The skewness or convexness weights are not applied at present.

3.2 Results

In figure 3 (10m tolerance, 21 sections), 13 of 48 original points are retained after generalisation. The original line (extent: 973m by 338m) is shown in grey; the result of WEA without partition is in black and the partitioned WEA in red. In this case, the chopping effect to the details on the left is slightly weakened under the partitioned scheme.

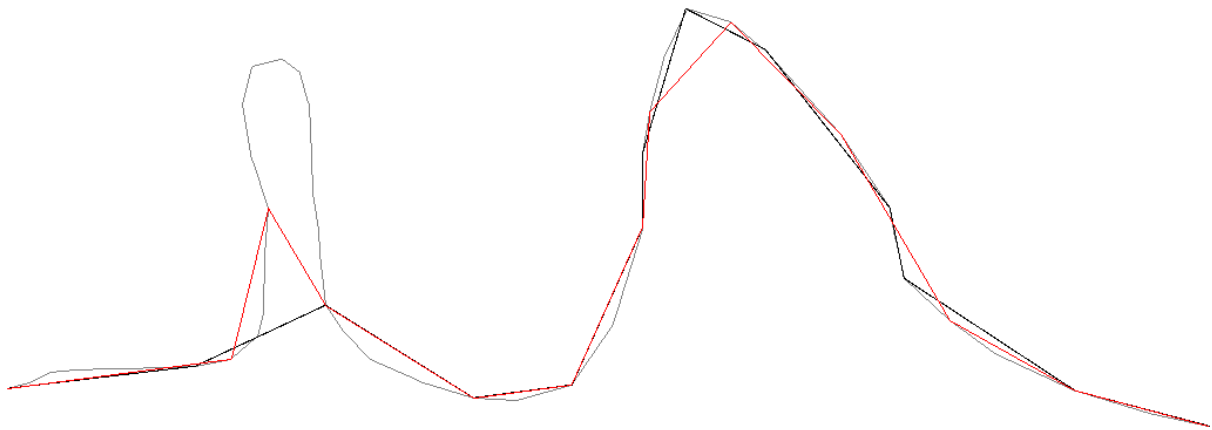


Figure 3. WEA Generalisation with RDP partition (1).

Figure 4 are the result of 13 points, 100m tolerance and 6 sections. The chopping effect is less obvious now.

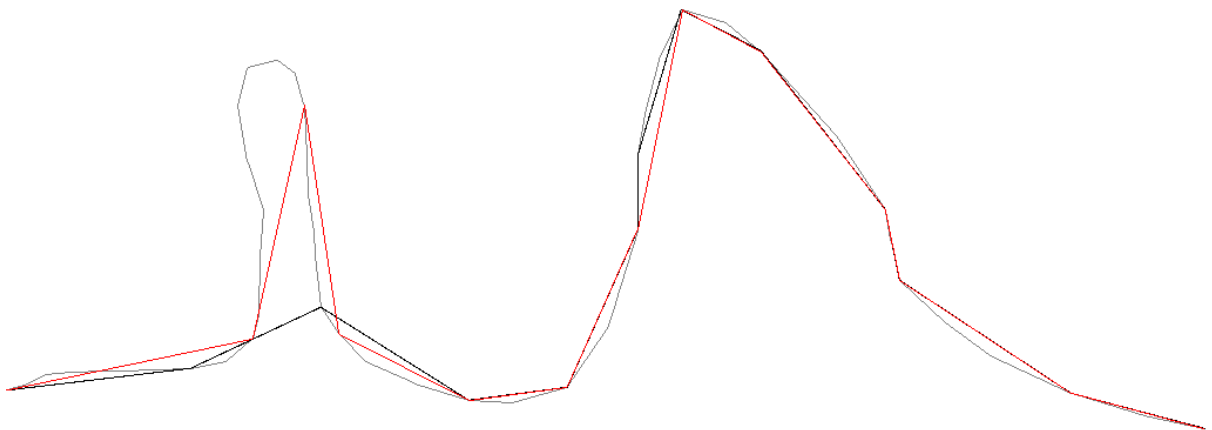


Figure 4. WEA Generalisation with RDP partition (2).

In figure 5 (13 points, 250m tolerance, 4 sections), the details are preserved almost perfectly.

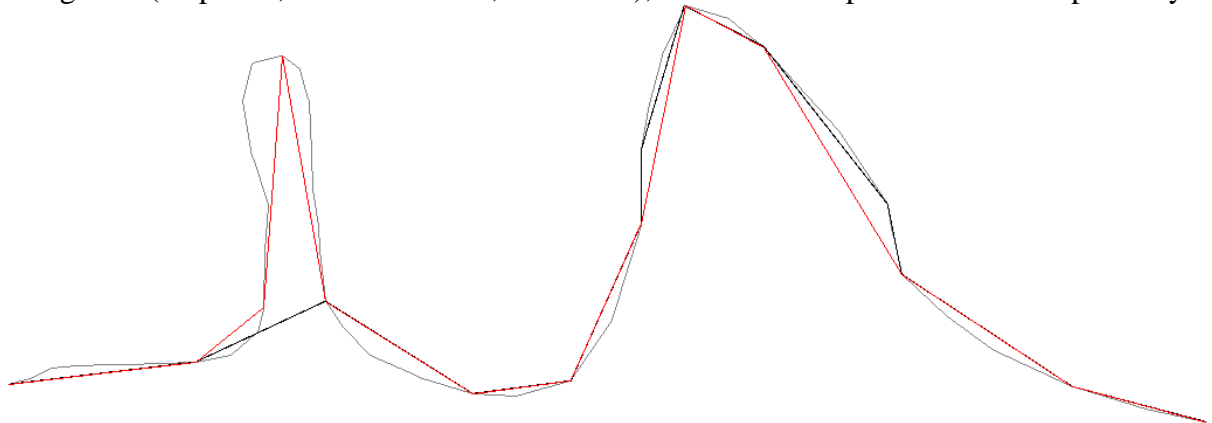


Figure 5. WEA Generalisation with RDP partition (3).

Figure 6 are results on OS MasterMap data. The tolerances are 50m (left) and 10m (right), with 85 of 1740 vertices retained.

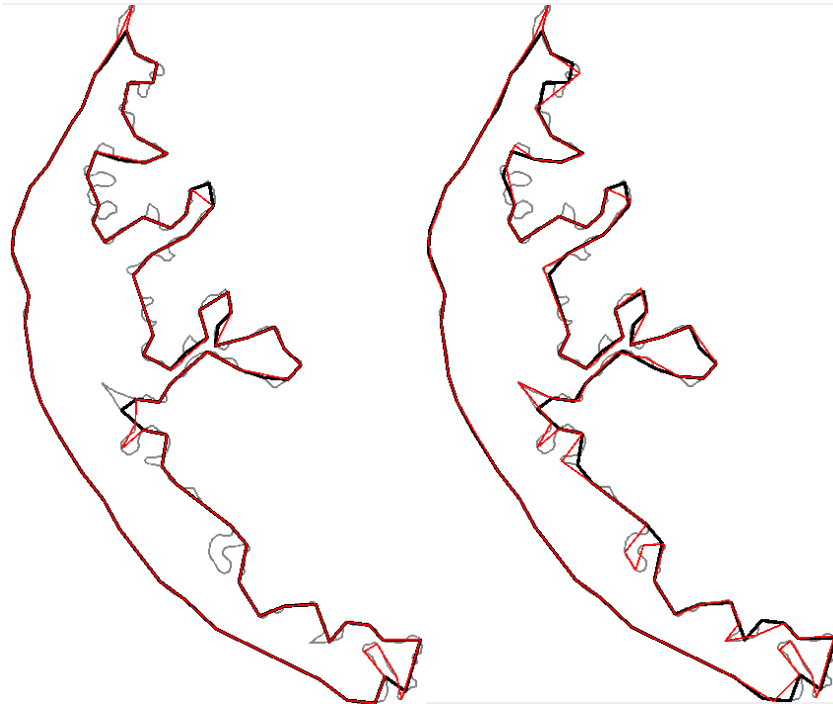


Figure 6. WEA Generalisation with RDP partition (4) - feature extent 120m by 203m.

Figure 7 is another example of OS MasterMap data. 246 of 2275 points are retained, with 10m (left) and 2m (right) tolerances.

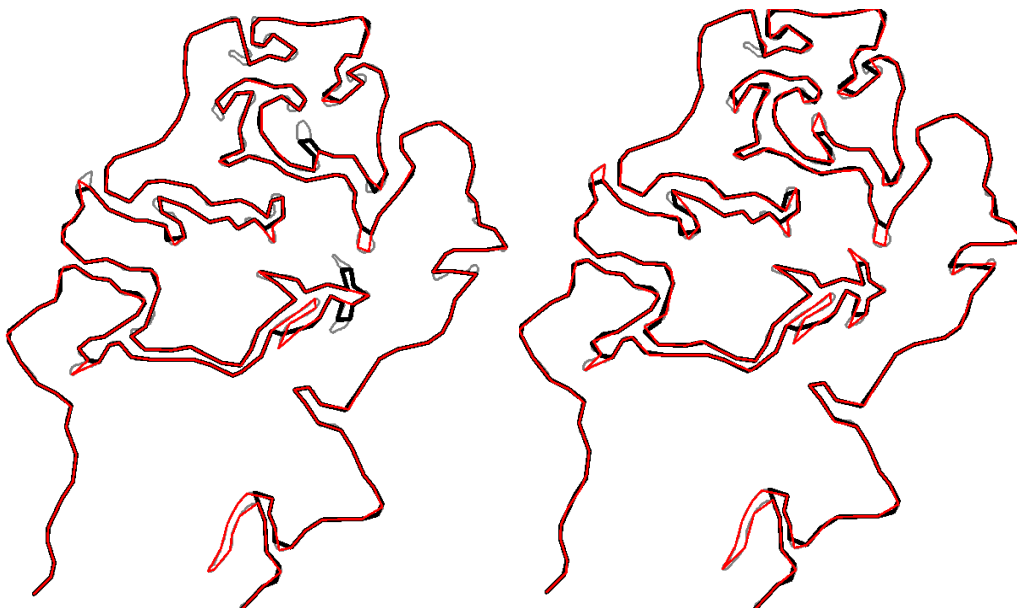


Figure 7. WEA Generalisation with RDP partition (5) - feature extent 108m by 129m.

Finally, figure 8 shows a local detail of the results from the original VW algorithm with and without partition scheme (500m tolerance). On the left, 257 of 2336 points are retained in the whole dataset; on the right, 183 points are retained.

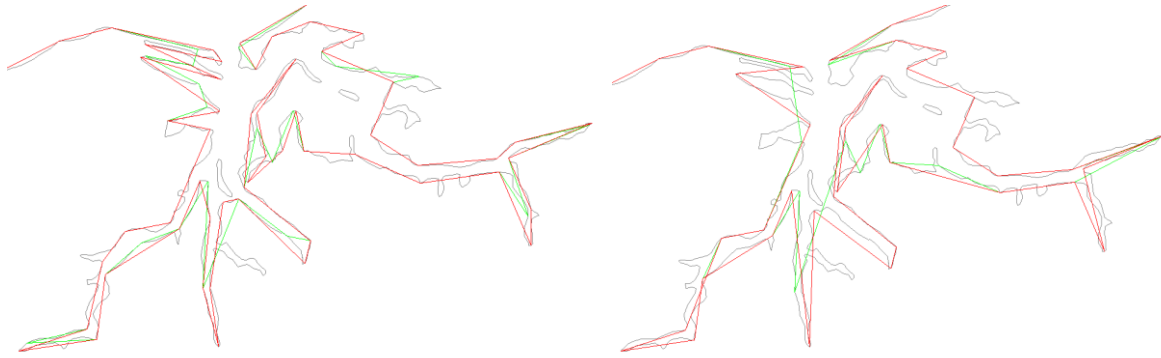


Figure 8. Original VW algorithm with (Red) and without (Green) partition.

4. Discussion and Future Work

4.1 Partition tolerance and generalisation effects

Comparing the results with different tolerance values, it is obvious that the RDP-based partition scheme is more useful under the circumstance of heavy generalisation. If the data is to be generalised only lightly, the partition scheme makes little difference. Also, the details in a feature are preserved under heavy generalisation if their “size” in general term is the same as or greater than the tolerance value. This may be served as a rough guideline for selecting partition tolerances.

4.2 Towards a flexible test platform for algorithm experiments

For our experiment we had the options of using either full-featured GIS packages or open-source GIS as the test platform. However, this experiment has some specific requirements/characteristics:

- As a proof-of-concept application, it has to be developed over a short period of time.
- It is a light-weight application and does not need to process large volume of data at present.
- We need to design and use non-standard data structures and require vertex-level access and manipulation to linear/areal objects
- We need to explore over a large range of the multi-dimension parameter space to compare results, recognise patterns and find appropriate combination of parameter values for different circumstances. For each of these combinations, graphic representations of the map features in the test data have to be generated in real time in order to visualise the results efficiently.

Although it is possible to achieve our objectives on a conventional general purpose platform, these options either introduce large development overheads or are likely to raise efficiency issues during the testing stage. In the end we derived our test platform from a GUI originally developed for demonstrating multi-scale spatial database (Zhou and Jones 2001b). This GUI is written in pure Java without using any third-party libraries. Figure 9 shows the current form of the GUI, which supports basic zooming and panning navigations in addition to various experiment specific features.

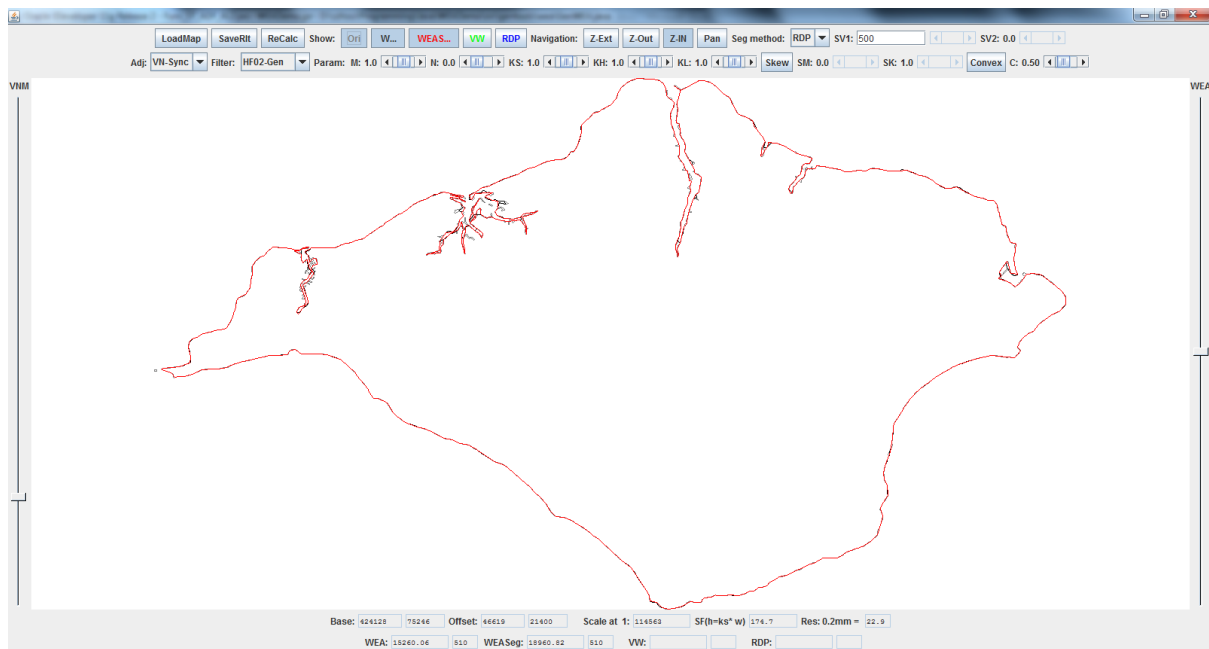


Figure 9. The GUI for current experiments.

Indeed, the original GUI has also been modified and used in several other previous research projects, including the original WEA development (Zhou and Jones 2004). We are exploring the feasibility of refactoring it to create a more generic environment for algorithm experiments, probably with support to some data structures (e.g. fully dynamic constrained Delaunay triangulation) seldom available as an integrated part in other environments. If this can be achieved, we wish we will be able to make it a contribution to the open-source/open-GIS communities.

4.3 Future work

Current experiment has demonstrated the usefulness of partition schemes in line generalisation. Nevertheless, the RDP partition scheme used in this pilot study is one-dimensional and its quality in capturing and describing complex details on linear features is not always satisfactory. In the next stage of our research we plan to extend the CDT and skeleton based method described by van der Poorten and Jones (2002) to model feature details explicitly and construct a hierarchy graph of feature details, hence offer more powerful controls (e.g. direct removal or dimension collapse of individual details) over the generalisation at both local and global levels.

Acknowledgement

Map data in this paper are used by permission of Ordnance Survey on behalf of HMSO. ©Crown Copyright. All rights reserved.

References

- CHILTON, S., 2010, OS OpenData and its effect on non-commercial cartography, *The Cartographic Journal*, **48(3)**, pp. 183-185.
- 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)**, pp. 112-122.
- OLSZEWSKI, R., FIEDUKOWICZ, A. and PILLICH-KOLIPINSKA, A., 2013, Utilisation of computational intelligence for simplification of linear objects using extended WEA algorithm. 16th ICA Workshop on Map Generalisation. Dresden, Germany 23th – 24th August 2013.
- RAMER, U., 1972, An iterative procedure for polygonal approximation of planar closed curves. *Computer Graphics and Image Processing*, **1**, pp. 244-256.
- VAN DER POORTEN, P.M. and JONES, C.B., 2002, Characterisation and generalisation of cartographic lines using Delaunay triangulation. *International Journal of Geographical Information Science*, **16(8)**, pp773-794.
- VISVALINGAM, M. and WHYATT, J.D., 1993, Line generalisation by repeated elimination of points. *Cartographic Journal*, **30(1)**, pp. 46-51.
- ZHOU, S. and Jones, C.B., 2001a, Multi-scale spatial database and map generalisation. 4th ICA Workshop on Progress in Automated Map Generalization, Beijing, China, 2-4 August 2001
- ZHOU, S. and JONES, C.B., 2001b, Design and implementation of multi-scale databases. In C.S. Jensen et al. (Eds.): *Proceedings 7th International Symposium on Spatial and Temporal Databases (SSTD'01)*, pp. 365-384. LNCS 2121 Springer, ISBN 3-540-42301-X
- ZHOU, S. and JONES, C.B., 2004, Shape-aware line generalisation with weighted effective area. In Peter F Fisher (Ed.): *Developments in Spatial Data Handling, Proceedings SDH'04*, Springer, ISBN 3-540-22610-9.

Biography

Sheng Zhou received his PhD in City and Regional Planning from Cardiff University in 2000. After post-doc research on multi-scale spatial database and multi-agent generalisation at Glamorgan and Cardiff, he joined Ordnance Survey Research in 2006. His current interests include: map generalisation, multi-representation spatial database, geometric computation and geo-semantics.