EVALUATION RESULTS OF AUTOMATED SCHEMATIC MAP TOOL FOR MOBILE LBS APPLICATIONS

S. Anand[1], J.M.Ware[2], M. Jackson[1] and J.Swan[1]

[1]Centre for Geospatial Science, University of Nottingham, UK
[2]Faculty of Advanced Technology, University of Glamorgan, UK

KEY WORDS: Automated schematic maps, simulated annealing, map generalization, spatial conflict, optimization algorithms, scale reduction, mobile LBS

ABSTRACT:

Mobile Location Based Services (LBS) refers to the infrastructure needed to provide various services to a user based on the user position. These applications act according to a geographic trigger, such as input of a place name, postcode, position information from a GPS (or in future, GALILEO), location information from mobile phone network etc. A schematic map is a diagrammatic representation based on linear abstractions of networks. Schematic maps can be used as visualization tool to help ease the interpretation of information by the process of cartographic abstraction especially for large scale digital network datasets. This paper presents the results of an extensive set of experiments carried out to evaluate the automated schematic map generating software developed for mobileLBS applications. The software makes use of the simulated annealing optimization technique. The results of extensive experimentation carried out to consider the effects of various constraints implemented, importance of setting constraint cost weightings, issues of consistency (since there is a large random element to the algorithm) are presented.

1. INTRODUCTION

Map Generalization is the process by which small scale maps are to be derived from large scale maps. This requires the use of operations such as simplification, selection, displacement and amalgamation of features that are performed subsequent to scale reduction (Ware and Jones, 1998). Mobile LBS refers to the use of geographic data in the field on mobile devices like networked Personal Digital Assistant (PDA). The main components for Mobile LBS are global positioning system (GPS), handheld computer i.e. PDA's, and communication network with GIS acting as the backbone. (Figure 1)

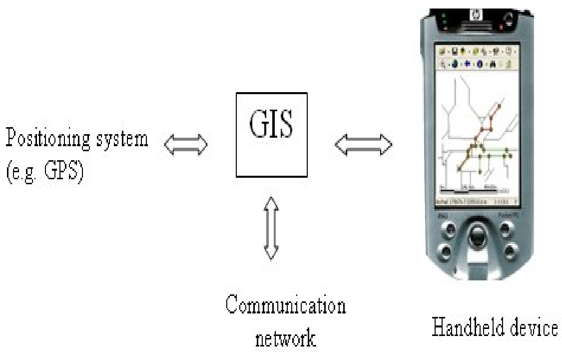

Figure 1: The basic components essential for Mobile LBS application. Also shows example schematic map generated from the prototype software displayed on an iPAQ PDA using ESRI ArcPad©

Generating schematic maps are an effective means of generalization of large scale network datasets. The aim is to enhance visualization at line networks and also make them user friendly for interpretation. The basic steps for generating schematic maps are to eliminate all features and networks (or portions of networks) that are not functionally relevant to the network system chosen for mapping. All geometric invariants of the network's structure are relaxed except topological accuracy. Routes and junctions are represented diagrammatically.

The schematization process was initially refined by Elroi (1988) as three main graphic manipulations. First, lines are simplified to their most elementary shapes. Next, lines are re-oriented to conform to a regular grid, such that they all run horizontally, vertically or at a forty-five degree diagonal. Third, congested areas are increased in scale at the expense of scale in areas of lesser node density.

This paper presents the summary of an extensive set of experiments carried out to evaluate the automated schematic map generating software developed for mobileLBS applications. The software makes use of the simulated annealing optimization technique. This technique has been successfully used to control operations of displacement, deletion, reduction and enlargement of multiple map objects to help resolve spatial conflict arising due to scale reduction. The results of extensive experimentation carried out to consider the importance of constraint cost weightings are presented.

2. SIMULATED ANNEALING APPROACH

The simulated annealing (SA) based schematization algorithm used in this work is similar to that used by Agrawala and Stolte (2001) to render easy-to-read non-schematic route maps. At the start of the optimization process SA is presented with an initial approximate solution (or state). The simulated annealing based algorithm is given below.

Algorithm SchematicMap
input: Initial, Annealing_Schedule, Stop_Conditions
$D_{Current} \leftarrow D_{Initial}$

```
t←GetInitialTemperature(Annealing_Schedule)
Cost_current=C(Current)
while NotMet(StopConditions)
    New←RandomSuccessor(D_Current)
    Cost_new=C(New)
    ΔE←Cost_current-Cost_new
    if ΔE >0 then
        Current←New
        Cost_current=Cost_new
    else
        p = e^(-ΔE/t)
        r = Random(0,1)
        if (r<p) then
            Current←New
            Cost_current=Cost_new
        end
    end
    t←UpdateTemperature(t, Annealing_Schedule)
end
Return(D_current)
```

In the case of the schematic map production, the input is the initial network: line features made up of edges, which in turn are made up of vertices. The initial state is evaluated using a cost function C; this function assigns to the input state a score that reflects how well it measures up against a set of given constraints. If the initial cost is greater than some user defined threshold (i.e. the constraints are not met adequately) then the algorithm steps into its optimisation phase. This part of the process is iterative. At each iteration the current state (i.e. the current network) is modified to make a new, alternative approximate solution. The current and new states are said to be neighbours. In simulated annealing algorithms the neighbours of any given state are generated usually in an application-specific way. In the algorithm presented here, a new state is generated by the function RandomSuccessor, which works by selecting a vertex at random in the current state and subjecting it to a small random displacement, subject to some maximum displacement distance. This compares to the random displacement methods favoured by Agrawala and Stolte (2001) and is in keeping with the random approach inherent to most simulated annealing based solutions. The new state is also evaluated using **C**. A decision is then taken as to whether to switch to the new state or to stick with the current. Essentially, an improved new state is always chosen, whereas a poorer new state is rejected with some probability p, with p increasing over time. The iterative process continues until stopping criteria are met (i.e. a suitably good solution is found or a certain amount of time has passed or a certain number of iterations have taken place without improvement).

As with other simulated annealing solutions, at each iteration the probability p is dependant on two variables: ΔE (the difference in cost between the current and new states) and t (the current temperature). p is defined as:

$$p = e^{-\Delta E / T}.$$

The variable t is assigned a relatively high initial value; its value is decreased in stages throughout the running of the algorithm. At high values of t higher cost new states (large negative ΔE) will have a relatively high chance of being retained, whereas at low values of t higher cost new states will tend to be rejected. The acceptance of some higher cost new states is permitted so as to allow escape from locally optimal solutions. In practice, the probability p is tested against a random number r ($0 \le r \le 1$). If $r < p$ then the new state is accepted. For example, if $p = 1/3$, then it would be expected that, on average, every third higher cost new state is accepted. The initial value of t and the rate by which it decreases is governed by what is called the annealing schedule. Generally, the higher the initial value of t and the slower the rate of change, the better the result (in cost reduction terms); however, the processing overheads associated with the algorithm will increase as the rate of change in t becomes more gradual.

The viability of any SA algorithm depends heavily on it having an efficient cost function, the purpose of which is to determine for any given element of the search space a value that represents the relative quality of that element. The cost function used here, C, is called repeatedly and works by assessing the extent to which a given state meets the set of constraints of the map.

When invoked initially, C evaluates a cost for each vertex in the network. This cost represents the extent to which each vertex meets the set of constraints. The overall cost is found by summing the individual vertex costs. A record of the individual vertex costs is maintained for future reference, meaning that, in any further call, C has to consider only vertices with costs affected by the most recent vertex displacement (Ware et al, 2006).

3. CONSTRAINTS

The schematic map production presented here considers five primary constraints (Anand 2006, Avelar 2002):

- Topological: The original network and derived schematic map must be topologically consistent;
- Orientation: If possible, network edges should lie in a horizontal, vertical or diagonal direction;
- Length: If possible, all network edges should have length greater than or equal to some minimum length;
- Clearance: If possible, the distance between disjoint features should be greater than or equal to some minimum distance (to ensure clarity);
- Angle: If possible, the angle between a pair of connected edges should be greater than or equal to some minimum angle (to ensure clarity).

Two secondary constraints i.e Rotation and Displacement are also included. Their purpose is to minimize unnecessary changes to the input network that are likely to occur due to the random nature of simulated annealing.

- Rotation: An edge's orientation should remain as close to its starting orientation as possible;
- Displacement: Vertices should remain as close to their starting positions as possible.

Each of these constraints can be evaluated using straightforward computational geometry functions, e.g. edge/edge intersection test and vertex to edge distance calculation. In order to work efficiently, certain of these functions require the use of a spatial index to avoid sequential scanning of the whole workspace. A simple regular two-dimensional indexing scheme was used in the implementation of the simulated annealing optimization approach.

4. SUMMARY OF EXPERIMENTS

This section provides summary on a series of experiments carried out to evaluate the schematic map software (and its underlying simulated annealing algorithm). First, the influence of the Douglas-Peucker algorithm used in pre-processing, and in particular the choice of weed tolerance value, is examined. Next, a series of experiments that consider the importance of using suitable annealing schedule parameters are presented. This is followed by a number of examples that demonstrate the usefulness of the various constraints, and the significance of setting constraint cost weighting appropriately. Also the issue of consistency of results is discussed.

4.1 Results with varying Douglas-Peucker weed tolerance values

In this work the network data presented as input to the simulated annealing schematic map software is pre-generalized using the ArcGIS ArcInfo Workstation Generalize tool. This makes use of an enhanced version of the Douglas-Peucker algorithm (1973). The enhancement ensures that, provided the point remove and topological error check options are selected, a topologically consistent simplification of the network. It achieves this by reintroducing into the generalized line vertices that would otherwise have been discarded.

Application of the Douglas-Peucker line simplification to a set of line features results in a new set of line features in which each feature is represented by a subset of its original vertices. The number of vertices removed during the process (i.e. the level of simplification) depends both on the complexity of the input data, the scale of the data and a user-defined parameter referred to as the weed tolerance. In general, the higher the value of this tolerance, the greater the number of vertices removed. It therefore follows that the choice of weed tolerance value used at the pre-generalization stage will ultimately affect the look and quality of the schematic map produced. As such, an experiment was carried out to assess the influence of the weed tolerance value.

The experiment simply involves pre-generalizing the datasets using a range of weed tolerance values, generating a schematic map for each of the pre-generalized datasets, and visually inspecting and assessing the resulting schematic maps. A sample of the outputs is shown in Figure 2. For this sample dataset it was found that tolerance values above 2m produce good results (though it is noted that this observation is quite subjective). Values of about 2m or less give schematics in which there is too much detail. Finding a method for automatically setting the tolerance value for any given data set would be useful, and this will form part of future work.

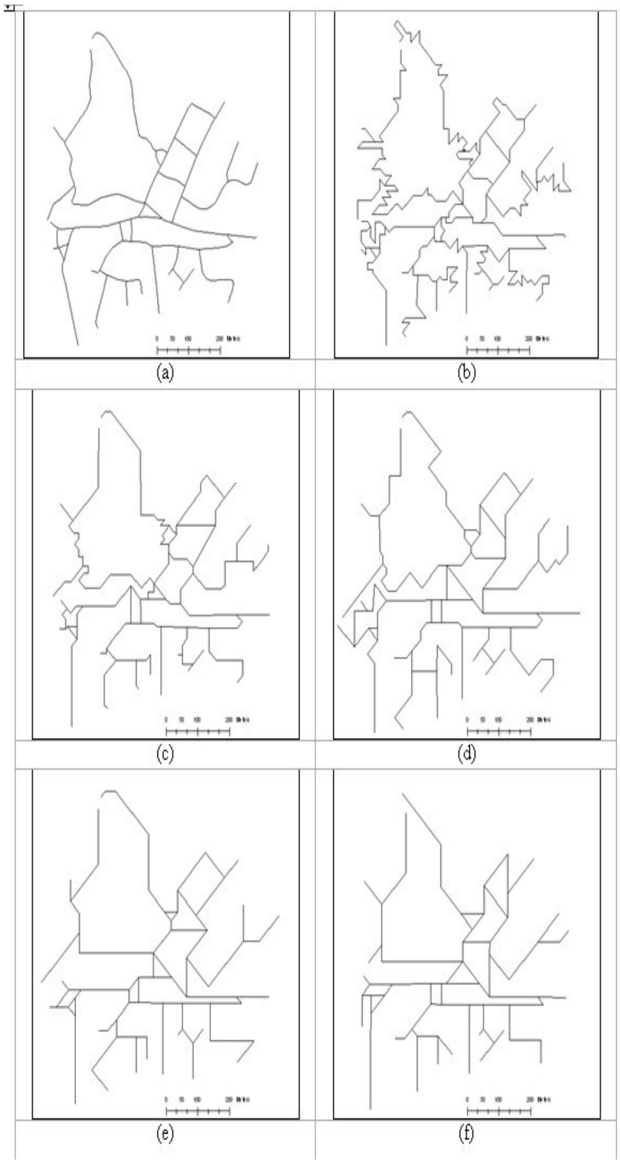

Figure 2: Schematic maps produced from the example dataset pre-generalized using a range of Douglas-Peucker tolerance values. (a) Original data (b) 0m schematic (371 vertices); (c) 1m (254 vertices) ; (d) 2m (205 vertices); (e) 10m (150 vertices); (f) 30m (142 vertices).

4.2 Constraint cost weightings

The relative importance of each of the constraints (i.e. Orientation, Length, Clearance, Angle, Rotation and Displacement) is controlled by an associated constraint cost weighting. Note that the Topological constraint is dealt with separately (i.e. any displacement that gives a topological error is rejected automatically). Varying the relative value of constraint weightings will produce schematic maps with varying characteristics. This is demonstrated by a series of examples, each generated from the sample dataset given below. (Figure 3)

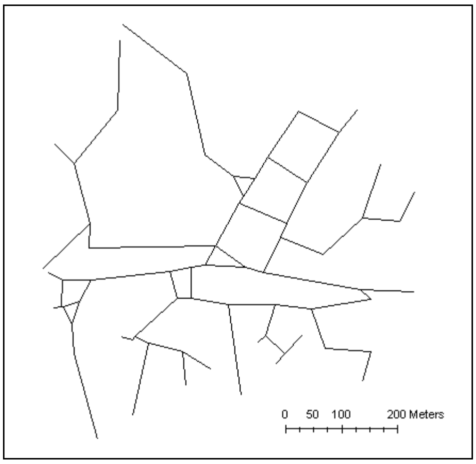

Figure 3: Sample dataset used for the experiments

4.2.1 Orientation

Figure 4. shows a simple schematic generated with Orientation cost weighting = 50. All other weightings were set to 0. At first glance it might appear to be a good schematic. Indeed, the edges all appear to have been re-oriented correctly. However, in at least one situation, in an effort to become correctly aligned, edges appear to have become coalesced (note that the middle "triangle" appears to have disappeared – compare with Figure 3). In fact, the edges are close to, but not quite coalesced (coalesced edges would have triggered a topological error).

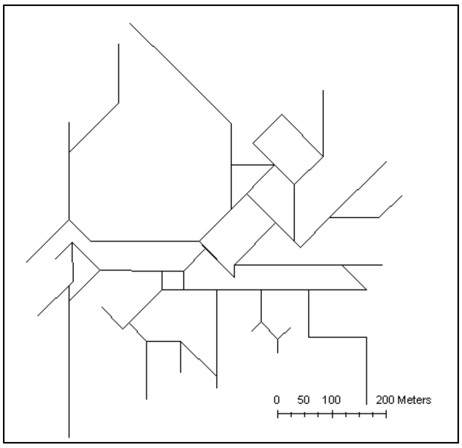

Figure 4: Schematic generated with Orientation weighting = 50.

4.2.2 Orientation and Angle

In order to address this problem, the schematic algorithm is again applied to example dataset. This time the Angle constraint weighting is set to 5 (Orientation weighting = 50). This has the desired effect of reducing the likelihood of edges becoming coalesced (Figure 5). However, in this particular example two edges now appear to join (in fact they do not, otherwise a topological error would be identified) whereas in the original data they were disjoint (the two top most edges in Figure 3). This has happened by chance. Connectivity is an important consideration for end user and hence likelihood of edges appearing to join should be avoided.

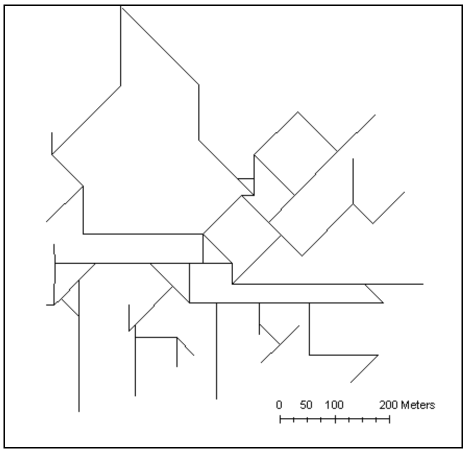

Figure 5: Schematic generated with Orientation weighting = 50 and Angle weighting = 5.

4.2.3 Orientation, Angle and Clearance

In order to resolve this problem the Clearance constraint is activated. This is achieved by setting a minimum clearance value and Clearance weighting > 0. The map shown in Figure 6 is generated using a minimum clearance value of 20m and a Clearance weighting of 1.

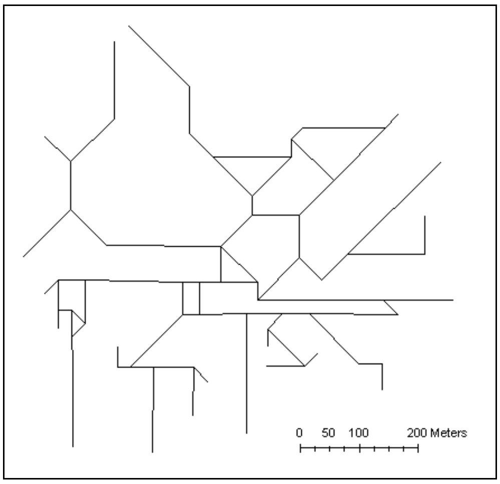

Figure 6: Schematic generated with Orientation weighting = 50, Angle weighting = 5 and Clearance weighting = 1. Minimum clearance distance = 20.

4.2.4 Orientation, Angle, Clearance and Length

It could be for certain display scales, or because of line and node symbolisation, a minimum edge length is required for reasons of legibility. This is achieved by adding in the Length constraint. In Figure 7 this has been achieved by setting the Length constrain weighting to 0.5 and the minimum edge length value = 50

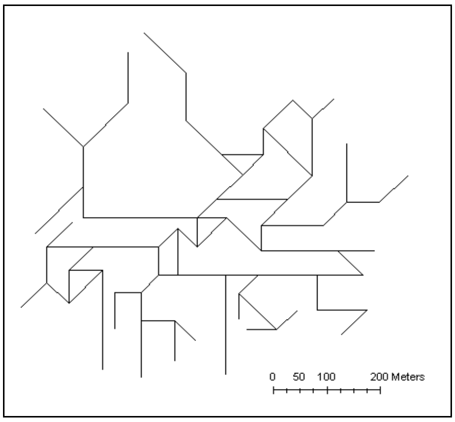

Figure 7: Schematic generated with Orientation weighting = 50, Angle weighting = 5, Clearance weighting = 1 and Length weighting = 0.5. Minimum clearance distance = 20 and minimum edge length = 50.

4.2.5 Orientation, Angle, Clearance, Length and Rotation

It will almost always be desirable to minimise any unnecessary change that takes place during the simulated annealing process. Here (Figure 8) the difference between original edge orientation and final edge orientation is minimised by introduction of the Rotation constraint (achieved by setting its weighting > 0). Care must be taken in setting the weighting – if it is too high then it will prevent other, possibly more important, constraints from being met.

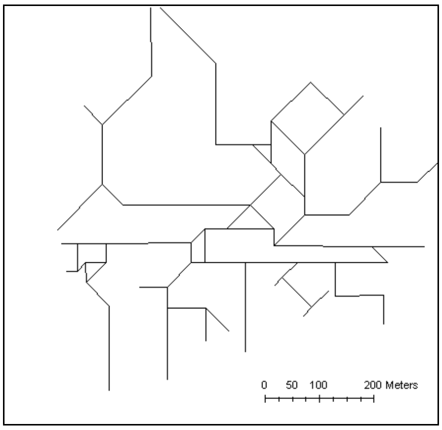

Figure 8: Schematic generated with Orientation weighting = 50, Angle weighting = 5, Clearance weighting = 1, Length weighting = 0.5 and Rotation weighting = 5. Minimum clearance distance = 10 and minimum edge length = 20.

4.2.6 Orientation, Angle, Clearance, Length and Displacement

In Figure 9 the difference between original vertex position and final vertex position is minimised by introduction of the Displacement constraint. As is the case with the Rotation constraint, the weighting value most be set carefully – if it is too high then it will prevent other, possibly more important, constraints from being met.

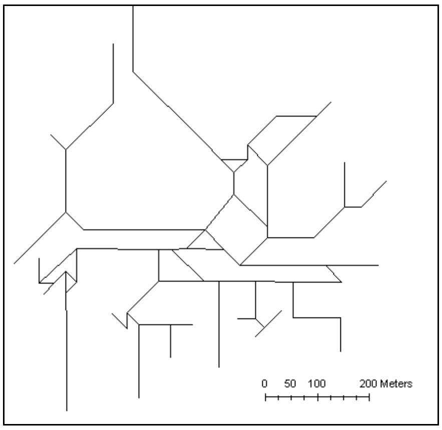

Figure 9: Schematic generated with Orientation weighting = 50, Angle weighting = 5, Clearance weighting = 1, Length weighting = 0.5 and Distance weighting = 1. Minimum clearance distance = 10 and minimum edge length = 20.

4.3 Consistency of result

The simulated annealing algorithm described involves the random displacement of randomly chosen vertices. Running the algorithm repeatedly on the same dataset and with the same algorithm parameters will produce different results each time. In order to assess how consistent these results are, 100 schematics were generated from the example dataset (in each case the same parameters were used and the algorithm was allowed to run for 20 seconds). The average final cost over the 100 executions was 236, with a standard deviation of 18.5 (minimum cost = 200 and maximum cost = 260). This suggests that there is a reasonable amount of consistency in result.

Furthermore, it is pointed out that the simulated annealing implementation makes use of the VBA *Rnd* function to generate random numbers. The function is initialised with some arbitrary seed value. Each initialising value will typically produce a different random sequence, and solutions will vary. However, the same initialising value will always produce the same random sequence. This property provides a mechanism for reproducing previous solutions (which is achieved by simply keeping a record of seed values used).

5. CONCLUSIONS

This paper has presented an experimental evaluation of the simulated annealing schematic map algorithm. These maps are especially of great usability in mobileLBS applications. It has been shown that while the use of the Douglas-Peucker algorithm as a pre-process leads to good schematics, the choice of weed tolerance value is important. The usefulness of the various constraints has been verified, as well as the significance of setting constraint cost weighting appropriately. Consistency of results (which may have been in doubt due the random nature of the algorithm) has been confirmed.

Further work is still necessary to find dataset characteristics and costs to decide for the optimal schematization strategy to be applied. We expect to repeat the experiment for various datasets and observe outputs for datasets of varied nature. Also finding a method for automatically setting the tolerance value for the pre-generalization on any given data set would be useful, and this will form part of future work.

6. REFERENCES

Agrawala M. and Stole C. 2001 Rendering effective route maps: improving usability through generalization. In Proceedings of *SIGGRAPH 2001*. Los Angeles: 241-250.

Anand S. 2006 Automatic derivation of schematic maps from large scale digital geographic datasets for MobileGIS, PhD Thesis, University of Glamorgan.

Avelar S. 2002 *Schematic Maps on Demand: Design, Modeling and Visualization*, PhD Thesis no.14700, Swiss Federal Institute of Technology: Zurich.

Douglas D. H. and Peucker, T.K. 1973. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112-122.

Elroi D. 1988 Designing a Network Linemap Schematization Software Enhancement Package. In Proceedings of the Eighth Annual *ESRI User Conference*, Redlands, CA.

Ware J. M., Anand S., Taylor G.E. and Thomas N. 2006 Automatic Generation of Schematic Maps for Mobile GIS Applications, Special issue on Location Based Services and Mobile GIS, *Transactions in GIS*, Blackwell Publishing.,10(1): pp 25-42.

J.M. Ware, C.B.Jones, 1998, Conflict Reduction in Map Generalization Using Iterative Improvement", GeoInformatica2:4, 383-407

7. ACKNOWLEDGEMENTS

The authors express thanks for the Ordnance Survey, UK for the data used in this work.

.