A ‘S eat of the Pants’
Displacement Algorithm

Rupert Brooks
Natural Resources Canada and Carleton University
brooks@nrcan.gc.ca
Dagstuhl, 06/05/2001
Generalisation at the National Atlas of Canada

- Like most NMA’s, moving to an infrastructure based approach
- No longer maintaining multiple datasets at varied scales
- Deriving many scales from one of a few sets of framework data.
- We talked the talk - but could we walk the walk?
A T est Project

• Complete reconstruction of 1:4M paper map of the Northern Territories.

• Hydrology to be derived from the 1:1M Canadian Framework Hydrology layer.

• Many challenges
  – the hydrology data was incomplete
  – only some software, and not “industrial strength”
  – skeptical cartographers
Philosophical Approach

- Never expected a perfect solution
- *Sought a solution which reduced cost and workload of the manual process*
  
- Never expected on-the-fly results
- *Storage of intermediate products was fine so long as they were automatically produced*
Model & Cartographic Generalisation

- Model or database generalisation
  - *Selecting* the set of features / attributes to appear
  - Used algorithms from Dianne Richardson and Robert Thomson

- Cartographic Generalisation
  - renders the features for visualisation
  - subjective; ultimate goal communication
Generalisation Operators

- Simplification / Characterisation
- Smoothing
- Aggregation
- Displacement
- Exaggeration
  - All have the elusive goal of maintaining the character of the cartographic feature
  - Success is subjective, therefore
It was difficult... but successful

- This process worked 1:4M northern map.
- The hydrology derived from 1:1M framework data
- Used automated tools that took the data part way to a final product.
- Experienced cartographers brought the result the rest of the way.
Lessons learned

- Highly structured and attributed data a necessity
- Data construction more expensive than generalisation - either automatic or manual
- Must rely on the many spin-off benefits of data structuring to make the proposition economical
The Production Process

- Prepare the Framework data
- Model generalisation (selection)
- Cartographic Generalisation
- Refinement by professional cartographer
National Atlas of Canada
Automated Generalisation Process
Cartographic Generalisation
National Atlas of Canada
Automated Generalisation Process
Cartographic Generalization
Displacement

- Required for islands
- Required in short time frame
- Implemented in Arc/Info and Perl
- Many corners cut at implementation time
- Significant room for improvement
- Computationally intensive - runs of hours quite common
The Model

- Rigid objects - Repulsive Force
  - (Similar to Lonergan and Jones (99) but less sophisticated)
- Based on the idea of reverse gravity
  - Objects too irregular and close to use centroids
  - Falls off more rapidly with distance
  - No rotation, or inertia
- Parameter: minimum visible distance, d.
Displacement Algorithm

1. Select features which may be displaced
2. Buffer (width d) those features to identify interacting clusters
3. Determine distance and bearing between each object based on an average of distance and bearing between closest 5 vertices
4. Compute the force exerted by object A on B as:
   \[ \bar{F} = \frac{10^6 \cdot \text{Area}(A)}{d_{AB}^3} \cdot \bar{e} \]
5. Sum Forces on each object
6. Compute the movement of each object as
   \[ \text{disp} = \frac{\bar{F}}{\ln(\text{Area}(B))} \]
7. Move the objects. (Limit move to d/2)
8. Check for interference.
9. While interference exists
   9.1 Move one of the interfering objects to its original position
   9.2 Check for interference again
1. Select features to displace
2. Cluster into groups
2a. Trim mainland involved
3. Calculate interactions
A close up example - both good and bad

3. Calculate interactions
4. Calculate forces
A close up example - both good and bad

4. Calculate forces
5-6. Calculate displacement
A close up example - both good and bad

5-6. Calculate displacement
7. Displace
A close up example - both good and bad

But oops here

This is good

7. Displace
8-9. Clean up topology
Result

- Simplistic
- Occasional failures
- Long running times
- But effective enough in practice
Rewriting the algorithm

- Computationally intensive
- Within a cluster, computation requires $O(v^2)$ operations ($v$=#vertices)
- Original implementation inefficient for many reasons
- Explore efficiencies due to hardware, language and parallelism
Reimplementation

• Reimplemented steps 3-5 at Carleton U.
  – using C++ on Intel / Linux systems
  – using MPI library for parallelisation

• Verified implementation by direct comparison with system running at NRCan
Efficiency gains

• Hardware / OS
  – Intel PII-400 / Linux is about 3 times faster than Sun Ultra10 / Solaris (and about 3 times cheaper)

• Language
  – C++ implementation at least 20 times faster than Perl
Parallelisation

- Clusters can be shown to be independent
  - Model has defined no influence between clusters
  - Each cluster has no neighbors closer than d
  - Each element in the cluster can move a maximum of d/2
  - Therefore, no topological problem can be created between clusters
Statistical behavior of clusters

Distribution of Cluster Size in Vertices

Size
Naïve assignment of data to nodes

- Clusters were assigned to nodes in an arbitrary order.
- For “large enough” dataset work should be assigned relatively evenly among the nodes.
- Preliminary results show that this is true, but “large enough” may be quite large.
Maximum Running time of parallel nodes

Time (Seconds)

1 node 2 nodes 4 nodes 8 nodes

pacific (35939 vertices)
searctic (39242 vertices)
yukon (45529 vertices)
thelon (80713 vertices)
Large (165484 vertices)
Average performance of parallel nodes

Time (seconds)

1 node 2 nodes 4 nodes 8 nodes

pacific (35939 vertices)
searctic (39242 vertices)
yukon (45529 vertices)
thon (80713 vertices)
Large (165484 vertices)
Conclusions

- The Generalisation techniques used to produce the National Atlas 1:4M map worked inefficiently
- We have shown significant performance increases through hardware changes, language changes and parallel implementation
Conclusions (2)

- We have shown that the algorithm behaves nicely in parallel if the dataset is large enough.
- Better assignment of clusters to nodes may give better performance on smaller datasets.