Problem description

The Displacement Operator

Frequently as the scale of representation is reduced it becomes difficult to distinguish individual objects in a visualisation of spatial data. When this happens, the displacement operator becomes important. In traditional cartography, the cartographer would move and or reshape the objects in question to achieve two goals:

1. the relative number, size and position of the objects is successfully communicated to the end user

2. the objects are separated far enough so that the medium of communication allows them to be successfully distinguished.

For example, consider the following group of islands. To visualize these effectively at a given scale it is important to be able to observe that they are physically separate. On a paper map the minimum gap that can be successfully printed is about 0.2mm. (On screen the minimal gap tends to be larger, but screen based cartography is not yet a mature field so it is difficult to provide an exact figure)

The overall goal is to communicate the fact that there are islands present, and their relative size and orientation. A map at a scale where the gaps between objects are near the limits of visibility for the map should not be used to make numeric measurements of those features. Therefore, positional accuracy is secondary for this application. A great deal of subjectivity is possible in the realization of a cartographic representation. Throughout this paper, we will use an relatively simple island cluster as a working illustration. Figure 1 shows this cluster as it exists in the 1:1 000 000 scale framework data [1] for Canada.

[1] At the time the work was being done, the data was not yet published as framework data. Since then the data has been published as the GeoBase Level 0 Hydrology and is available from http://geogratis.cgdi.gc.ca.





Figure 1





Improving Computational Efficiency

Improvements on 'Quick and Dirty' Approach

Although the software operated effectively in practice, it was clear from the beginning that more efficiency should be possible. There were a number of potential sources of inefficiency. Firstly, Perl is only a moderately efficient language. There are two sources of inefficiency in a Perl implementation. Perl is not a true scripting language as it is compiled to machine code when executed, however there is a performance hit on each execution as the code is compiled. A further performance issue is due to Perl's type flexibility. Although this makes for flexible coding, it means that type must be determined at run time, which adds a few steps to every operation.

GEOIDE

Arc/Info is the workhorse of the GIS world and is a dominant player in the high-end GIS market. However, the author's experience is that Arc/Info can be quite slow in operation. In fact, previous experiments involving recoding Arc/Info based routines into Perl achieved between a 10 and 60 times increase in speed [1].

Finally, the algorithm appears to have great potential to be run in parallel. Each cluster can be completely considered completely independently of every other cluster. In parallel computation, reducing the amount of communication necessary between the nodes is an important way of improving performance. Therefore, using this clustering as a means of distributing the necessary computations results in minimal inter-node communications and thus in efficient operation.

Thus there were two avenues of investigation to be explored to improve the run-time of the software. Firstly, porting to a lower level language should allow greater efficiency to be obtained. Secondly, running the algorithm in parallel to improve performance significantly.

A Parallel C++ Implementation

Steps 3-5 of the algorithm were ported to C++[2], using the MPI (Message Passing Interface) architecture [3] to be able to distribute the processing load. Our program takes as input the intermediate file from step 2, and produces as output a set of moved polygons and a list of intersections between them. See [4] for more information.

Using the MPI architecture, the algorithm could be run in parallel across a selected number of nodes. The division of work amongst the nodes used a simple model, where each node was delivered all the data, but only computed the necessary results for a its partition of the data. Where objects were not in the same cluster, they could quickly be classified as not interacting by examining their cluster ids.

Analysis

An experiment was conducted to measure the efficiency gains made by the recoding of the algorithm. Six datasets of varying sizes were chosen, and were run on the original NRCan system, on the Carleton System in the original Perl, and on the Carleton System using the C++ version on 1, 2, 4 and 8 parallel nodes. The improvements due to parallelization are shown in the graphs to the

Notes

[1] Brooks, R. (2) 2000. "Using Waltz Filtering for Rapid Stream Ordering of River Networks".

Unpublished Assignment for Applied Artificial Intelligence.

[2] The work was partially supported by C3.ca, NCE: GEOIDE and NSERC. [3] Gropp, W., Lush, E., Skjellum, A. 1994. Using MPI: Portable Parallel Programming with the Message Passing Interface. MIT Press, Boston.

Model:



Two objects that are far apart relative to their diameter can be approximated by point masses at their centres of gravity. See equation 1.

Two objects that are close together relative to their diameter cannot be approximated by point masses at their centers of gravity. Different regions of the objects are at such widely varying distances from each other, that their influence must be considered independently and integrated. See equation 2.

Suppose that each object (island, mainland) emits a field resembling reverse gravity. Then objects that are closely packed would experience a strong repulsive force. Objects spaced further apart would experience a lesser degree of repulsive force. In the usual formulation of Newtonian gravity, the attraction between two bodies is defined as:

 $G m_1 m_2$

Equation 1 : Standard gravity formula; force operates along the shortest path between centers of mass

where G is the gravitational constant, m₁ and m₂ are the masses of the two objects in question, d is the distance between their centers of mass, and e is the unit vector between their centers of mass. This formula is a simplification however, and treats each object as though it was shrunk to a point. This works properly so long as the objects are rigid and distance between them is very large relative to their size.

In this case, the objects of greatest interest are close together relative to their diameters. The objects are too close together for the usual set of formulae to apply. Thus it is necessary to go back to first principles and develop an integral formula for the force between the two objects (refer to Figure 2). Fortunately for the purpose of this model, three simplifications can be made. The objects can be assumed to be perfectly rigid, of constant density, and to not rotate. An equation respecting these parameters is shown in Equation 2.

 $\int (\frac{G^{2}}{d(x_{1}, y_{1}, x_{2}, y_{2})^{2}} e dx_{1} dy_{1}) dx_{2} dy_{2}$

Equation 2 : Gravitational force when objects are close relative to their size

A Parallel Displacement Operator R. Brooks, J. Chen, D. Nussbaum, and J.-R. Sack School of Computer Science, Carleton University, Ottawa, Ontario K1S5B6, Canada School of Computer Science, Carleton University, Ottawa, Ontario K1S5B6, Canada School of Computer Science, Carleton University, Http://www.scs.carleton.ca/~sack Http://www.scs.carleton.ca/~nussbaum

Ottawa, Ontario K1A 0E9, Canada http://www.nrcan.gc.ca/~brooks

Ottawa, Ontario K1S5B6, Canada





Potential Future Improvements

Defining a neighbourhood

To describe the relationships between these objects in an intuitive way, it is necessary to quantify the idea of neighbourhood. It is proposed that the set of objects that are directly related to a given polygon are those that are touched by the largest enclosing convex polygon that does not overlap any other features. This is shown in Figure 3; note that polygon A in this example would not be a direct neighbour, but an indirect one. A reasonable approximation of these relationships could be reached using a skeleton structure of the intervening space between objects. There are several possible such structures, most promising are the Voronoi diagram [1] and Aurenhammer's straight skeleton [2]. Such a skeleton defines a tesselation of the background space such that each area of space is associated with one and only one object. Those objects, whose areas of influence are adjacent, are considered to be directly related. This is illustrated in Figure 4.

The relationship graph

Given such an arrangement, it is clear that the result can be described as a relationship graph. Each vertex in the graph represents an object in the dataset, and each edge represents a relationship. Constraints may be applied to each relationship, then as global parameters are changed, each relationship must be re-evaluated to meet the constraints. Such a graph is shown in Figure 5

Notes

May 2, 2001.

[1] Held, M. 2000. ``VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments". Computational Geometry: Theory and Application, 18(2): 95-123, 2001. [2] Aichholzer, O. and Aurenhammer, F. 1998. "Straight skeletons for general polygonal figures in the plane." In A.M. Samoilenko, editor, Voronoi's Impact on Modern Sciences II, volume 21, pages 7-21. Proc. Institute of Mathematics of the National Academy of Sciences of Ukraine, Kiev, Ukraine, 1998. Available from



Before being processed by the displacement algorithm the framework data was processed into a form relevant to cartographic representation. Selection of hydrologic features had been carried out, and the data was separated into both marine and freshwater resided in the area feature file

First Implementation at the National Atlas of Canada

Move Polygons

polygon, the vertices of the polygons are moved along the displacement vector. In practice it was necessary to modify the output of step 3 - by limiting the maximum movement of a polygon to achieve a more reasonable result. I was important in this step to retain the original position of the polygon so that step 6 could be carried out.

Given the displacement for each



Canada Natural Resources Canada



Carleton UNIVERSITY

http://www.cis.tugraz.at/igi/auren/publications.html. Last accessed



Figure 3: neighbourhood



Figure 4: areas of influence



Figure 5: constraints graph

Clustering

Polygons outside a ertain radius of each other have no effect on one another. For this reason the polygons into clusters before computing the force acting on each

It was possible to reduce its



Compute Interactions

To compute the distance vector between a polygon pair, a simple vector between the centroids or the two closest points is not effective because each polygon may e oddly shaped. An approximate nethod used was to average the vectors between the 5 closest pairs of vertices in the boundary. The force is then computed proportional to the areas and inversely proportional to distance. .Finally we sum the forces acting on a polygon to determine its movement, which is inversely proportional to its



Check topology





Fix Topological Problems

To correct any topological errors that may have occurred, the intersecting vgons are moved back to their ment, it is necessary to check for moving the polygon back into place may have created another error



Practical Results:

The implementation described above was used in the production of the National Atlas map: Yukon Territory, Northwest Territories and Nunavut. The figures show first the raw data, the data just after displacement, and finally the final printed map. Although not perfect, the displacement algorithm described here was an important component of the production process for this map [1].

[1] Brooks, R. 2000. 'National Atlas of Canada Producing First Map Using Automated Generalisation of Framework Data.' Cartouche, Number 39, Fall 2000.



Southern Baffin Island - No Cartographic Generalization. This shows the appearance of the data with no cartographic generalization applied. This is at approximately twice the size of the intended map. Note the complex archipelagos along the coast.



Southern Baffin Island - Displacement of Islands

This shows the appearance of the data after the application of the displacement algorithm. The effect is subtle - it can be best seen by comparing the relative separation of small island groups between this figure, and the one above

Southern Baffin Island - Final Map

This shows the appearance of the same area on the final map. Many more cartographic processes besides the displacement algorithm have been applied to produce the final appearance.