EFFICIENT SETTLEMENT SELECTION FOR INTERACTIVE DISPLAY

Marc van Kreveld*
René van Oostrum*
Department of Computer Science
Utrecht University
email: {marc,rene}@cs.ruu.nl

Jack Snoeyink**
Department of Computer Science
University of British Columbia
email: snoeyink@cs.ubc.ca

ABSTRACT

Three new models for the settlement selection problem are discussed and compared with the existing models. The new models determine a ranking rather than a selection, which has advantages both from the efficiency and the geographic correctness point of view. We give figures of selections based on six different models, and explain how the models can be implemented efficiently.

1 INTRODUCTION

When a map is to be displayed on the screen, choices have to be made which cities and towns to include; this is called settlement or place selection (Flewelling and Egenhofer, 1993; Kadmon, 1972; Langran and Poiker, 1986; Töpfer and Pillewizer, 1966). It is intuitive that large cities should take precedence over smaller ones, but it is not true that if five cities are selected, these are the largest ones. A large city close to a yet larger city may be excluded, and a smaller city not in the neighborhood of any other larger city may be included because of its *relative importance.*

Settlement selection is performed just prior to generalization, although it can be considered as part of the generalization procedure as well. It has to be performed when a GIS user zooms out on a small scale map, or when a cartographer is in the process of interactively designing a map from a geographic database. On maps where cities fulfill a reference function, like on weather charts, clustering is undesirable, but on maps where for instance state boundaries have a reference function, clustering need not be avoided.

*Partially supported by the ESPRIT IV LTR Project No. 21957 (CGAL).

**Supported in part by grants from Canadian National Science and Engineering Research Council (NSERC), B.C. Advanced Systems Institute, and the Institute for Robotics and Intelligent Systems.

Settlement selection is followed by a name placement procedure, but we won't address that issue here. There is abundant literature on that topic.

This paper discusses models that have been described for settlement selection. We also describe a few new ones and discuss their advantages. We have implemented several of the models for comparison. In the process of interactive map design, it is useful if the cartographer has control over things like number of cities selected, and the degree in which clustering is allowed. We have included these controls in the interface of the implementation.

1.1 Previous work

Several decades ago, Töpfer and Pillewizer formalized a means to determine how many features should be retained on a map when the scale is reduced and generalization is performed (Töpfer and Pillewizer, 1966). Settlement selection itself starts by assigning an importance value to all settlements. The importance can simply be the population, but also a combination of population, industrial activities, presence of educational institutions, and so on.

Langran and Poiker report five different methods for the selection of settlements (Langran and Poiker, 1986). Most of them are incremental: cities are added from most important to least important, where the addition to the map is performed only if some spatial condition is not violated. In the *settlement-spacing ratio algorithm* and the *distribution-coefficient algorithm*, the selection of a settlement is determined by only one, more important settlement close by. In the *gravity-modeling algorithm*, selection is dependent on several settlements in the neighborhood. The *set-segmentation* and *quadrat-reduction* methods use recursive subdivision of the plane, and a direct application of the radical law by (Töpfer and Pillewizer, 1966).

Flewelling and Egenhofer discuss a number of factors that influence the selection of settlements (Flewelling and Egenhofer, 1993). Following (Mark, 1990), they assume that an importance attribute is assigned to the map features to allow for intelligent selection. Then they give a global discussion of ranking of settlements on non-spatial properties.

2 EXISTING AND NEW MODELS

Before describing the three models for settlement selection that we developed, we first discuss three existing models, reported by (Langran and Poiker, 1986): the *settlement-spacing-ratio* model, *gravity modeling* and the *distribution-coefficient-control* model. The other two methods that (Langran and Poiker, 1986) describe, *set segmentation* and *quadrat reduction*, require too much human intervention to be suitable for automated, or interactive, map design.

A disadvantage of the three existing models is that they don't directly give a *ranking* of the base set of settlements. A ranking is a display order; after computing a ranking of the base set beforehand, selecting any number of cities is simply a matter of choosing them in order of rank. For methods that don't determine a ranking, changing the number of selected settlements involves re-computation.

Adaptations can be made to the existing models to control the number of selected settlements from the base set, but this may have strange effects. For example, when selecting more settlements, it can happen that one of the chosen settlements is no longer selected, but instead a couple of others are. When selecting even more settlements, these discarded settlements can reappear. We say that a settlement-selection model has the *monotonicity property* if any selection of settlements necessarily includes the settlements of any smaller selection. Since our new selection models are based on a complete ranking of the settlements, they have the monotonicity property.

Although ranking facilitates the selection process, a model that produces a complete ranking is not necessarily better than a model that doesn't. The quality of a final selection depends on the data set used and the purpose of the resulting map. The quality of the existing models and our new ones can be assessed by comparing figures of selections (section 4) and by our implementation available on the World Wide Web.

In the models to be described next, we assume that an importance value is known for each settlement. The model defines which settlements are selected when their geographic location is incorporated as well.

2.1 Existing Models

Settlement-spacing Ratio In the settlement-spacing-ratio model, a circle around each settlement is placed around each settlement whose size is inversely proportional to its importance. More precisely, the radius is c/i where i is the importance and c is some constant (the same for all settlements). Settlements are added in order of importance, starting with the most important one. A settlement is only accepted if its circle contains none of the previously accepted settlements. In other words: small settlements will only be accepted if they are isolated.

The constant of proportionality c determines how many settlements are accepted; smaller values for c mean smaller circles and this generally leads to more settlements being selected for display. This is, however, not always the case, as is illustrated in Figure 1: settlements 1 is accepted, 2 is rejected, and 3 and 4 are accepted. But if c were slightly smaller, the circle of 2 would not contain settlement 1 anymore. So settlements 1 and 2 are accepted, but 3 and 4 are rejected, since their circles contain settlement 2. If we continue to decrease c, settlements 3 and 4 will reappear.

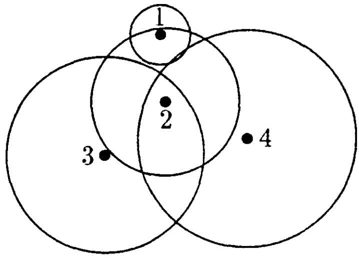

Figure 1: Settlement-spacing Ratio method

It follows that this method doesn't have the monotonicity property and that a complete ranking of the cities cannot be calculated. In fact, it can be that no value of c gives a desired number of settlements. It is also possible that two different selections have the same size. This is all caused by the fact the the monotonicity property is not respected by the model.

Gravity Modeling In the *gravity modeling* method, a notion of *influence* is introduced: the influence of one settlement on another one is computed by dividing the importance of the first one (the selected one) by the distance to the other. Settlements are tested in decreasing order of importance, and a settlement s is only accepted if its importance is greater than the summed influence of all already selected settlements on s.

As in the previous model, a constant of proportionality c is used. The next settlement under consideration is accepted if the summed influence of the already accepted settlements on the candidate is less than c times the importance of the candidate. By controlling c, the number of selected settlements can be adjusted. However, this model, like the previous one, doesn't respect the monotonicity property and doesn't give a complete ranking.

Distribution-coefficient Control The third method, *distribution-coefficient control*, uses the *nearest neighbor index* for the selection process. The nearest neighbor index is the ratio of the actual mean distance to the nearest neighbor and the expected mean distance to the nearest neighbor. Again, settlements are processed in decreasing order of importance. Starting with a small set of largest ones, settlements are only accepted if their addition to the already accepted set doesn't decrease the nearest neighbor index. The number of settlements in the final selection is fixed, but can be controlled by introducing a tuning factor. Again, this method doesn't result in a complete ranking of the settlements. A second disadvantage of the model is that the actual importance of a settlement is only used in the order of processing, not in the selection.

2.2 New Models

Circle Growth In the *circle-growth* method, a ranking of the settlements is determined as follows: for each settlement a circle is drawn with an area that is proportional to the importance of the settlement. The initial constant of proportionality c is such that no two circles overlap. The next step is to increase c, causing all circles to grow, until the circle of some settlement fully covers the circle of some other one. The former is said to *dominate* the latter; the latter has the lowest rank of all settlements and is removed. This process is repeated while assigning higher and higher ranks, until only the most important settlement remains.

This method satisfies two important conditions:

- When two settlements compete for space on the map, the most important one of the two will survive.
- Settlements of low importance will be displayed on the map if there are no settlements of higher importance in their proximity.

The drawback of this method is that a settlement with very high importance can have a global effect on the map: its neighborhood is a large part of the map, and too many settlements near to it are suppressed. At the same time, in large regions with no settlement of high importance several settlements are selected. One way of resolving this is instead of giving each settlement a circle with an area proportional to its importance i, letting the size of the circle be proportional to i^α, with $0 \leq \alpha \leq 1$. By tuning α the influence of the importance of the settlements on the selection can be reduced.

Circle Growth Variation I The drawback of the (unmodified) circle-growth model led to the observation that settlements with a very high importance have too much influence on the selection, and this resulted in the opposite of preserving density locally. Our second method, a variation on the circle-growth method, doesn't have this problem. We'll rank from first to last this time, and give all ranked settlements a circle of the same size, i.e., proportional to the importance of the most important settlement. All not yet ranked settlements have a circle with a size proportional to their importance.

The complete ranking is calculated as follows: the settlement with the highest importance is assigned the highest rank. Next, the settlement that is second in rank is determined by applying the circle-growth idea. We choose the settlement whose circle is covered last by the circle of the settlement with the highest rank, and set its importance to that of its dominating settlement. This process is iterated, ranking a next settlement when its circle is covered last by any one of the ranked settlements.

With this method the distribution of the selected settlements can be expected to be more even than the distribution of the selection resulting from the circle-growth method, since in our second method the size of the circles is the same for all selected settlements. Indeed, our implementation verifies this; an evenly distributed selection is the result.

Circle Growth Variation II In the previous two methods, all calculations are done with absolute importance values of the settlements. Our third method makes qualitative rather than quantitative comparisons. First, the settlements are sorted by importance from low to high. Each settlement receives as a number the position in the sorted order. This number replaces the importance value, after which the ranking is computed as before. Circles of selected settlements have equal size, and the size of the circles of the not selected settlements is proportional to their position in the list sorted on importance.

3 IMPLEMENTATION

We are currently in the process of implementing the three existing and the three new models. A preliminary stand-alone version is up and running; a java-script version is accessible via the World Wide Web at `http://www.cs.ruu.nl/~rene/settlement/`.

3.1 User Interface

The user interface will look like depicted in Figure 2: a large portion of the screen is reserved for displaying the settlements. Next to the display area are the controls: buttons for selecting which of the six implemented methods to use, or simply ranked by importance; buttons for displaying names and importance values with the settlements; buttons set the number of displayed settlements; buttons for increasing and decreasing the influence of the importance values on the selection; and a slider for adjusting the tuning factor used in the three existing models (see section 2.1).

3.2 Algorithms and Data Structures

In the first version of the program we didn't pay much attention to the efficiency of the algorithm, since our focus was on the outcome of the models rather than on speed. All models were implemented with a simple $O(n^2)$ time algorithm, or worse. However, in a practical situation like interactive map design, where data sets are large, efficiency becomes important.

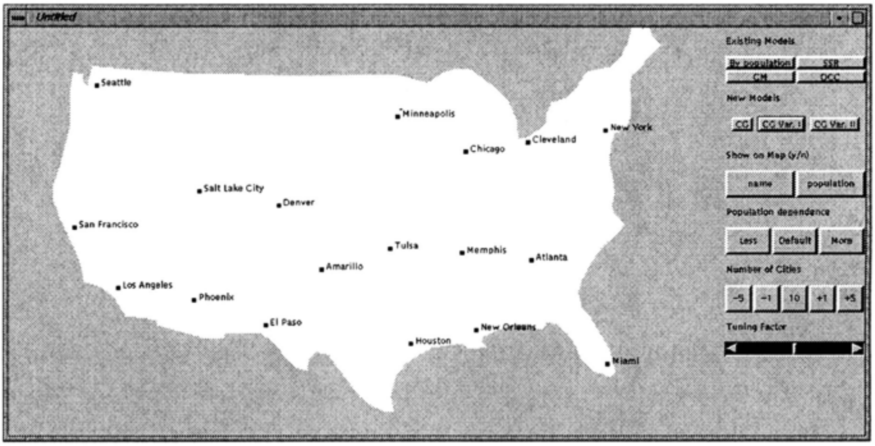

Figure 2: The user interface

Computational geometry techniques can be used to improve performance.

We'll concentrate on the implementation of the two variations of the circle-growth models, since they seem to give the best selection results. Both can be implemented by maintaining the Voronoi diagram of the selected settlements (see Figure 3). We start with one selected settlement, the most important one. Its Voronoi cell is the whole plane. Since the circles of all selected settlements have the same size, all non-selected settlements are dominated by their nearest selected neighbor. That is, their circle will be covered first by the circle of the nearest chosen settlement. So during the algorithm, we maintain for each Voronoi cell a list of non-selected settlements that lie in that cell. One of the settlements in each list is the last to be covered, and it is a candidate for the next settlement to be chosen. We maintain all these candidate settlements in a heap, which makes it possible to determine in $O(1)$ time the next settlement to be added to the set of selected settlements. Then we have to update the Voronoi diagram: a new cell is created, and a number of existing cells need to be changed. Also, the lists of non-selected settlements of the inflicted cells have to be updated, as well as the heap.

If the settlements are inserted in random order, the algorithm runs in $O(n \log n)$ expected time. In typical cases, the order in which the settlements are inserted will probably be sufficiently random for the running time to be closer to $O(n \log n)$ than to the $O(n^2)$ worst-case time.

Our first method, the unmodified circle-growth method, can be implemented in much the same way, but since here the circles of the selected settlements don't have the same size, we need weighted Voronoi diagrams

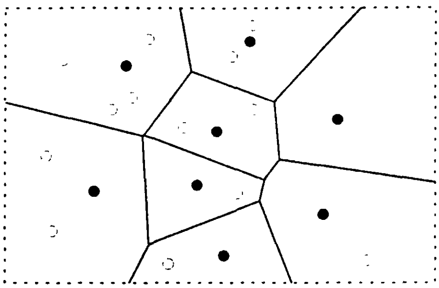

Figure 3: Maintaining the Voronoi diagram of the selected settlements

(Aurenhammer and Edelsbrunner, 1984; Okabe *et al.*, 1992), a variation of the standard Voronoi diagram that uses a different metric.

Of the existing methods, the settlement-spacing ratio method can also be implemented by incrementally constructing the Voronoi diagram of the selected settlements; a settlement is only accepted if its circle does not contain its nearest neighbor. Since settlements are added in order of importance, we don't need to maintain lists of non-selected settlements for each Voronoi cell. Note that only one complete selection is computed in $O(n \log n)$ time, not a complete ranking. So if more settlements are needed, the algorithm has to be started all over with a different constant of proportionality. In our algorithms a complete ranking of the settlements is computed in asymptotically the same time; after that, determining a selection takes time proportional to the number of settlements to be selected.

For the gravity modeling method, computing even one selection takes $O(n^2)$ time. It is not clear how to improve the performance of this model.

In the distribution-coefficient control method, testing each settlement involves determining its nearest neighbor, and determining for which of the already selected settlements the new settlement becomes the new nearest neighbor. With a straightforward algorithm this will take $O(n^2)$ time in total, but this can be improved to $O(n \log n)$ time for typical cases by incrementally constructing the Delaunay triangulation; the techniques are analogous to those for the circle-growth method. Again, this is the time needed for computing a single selection.

4 TEST RESULTS

We tested the three existing and the three new models on a (somewhat outdated) data set consisting of 158 cities of the USA. The population of the cities was used as the importance, and in each model 15 cities were displayed (see Figure 4). For the existing models, this involved using a tuning factor. Observe that of the existing models, the gravity model and the distribution-coefficient control method show some clustering. Our unmodified circle-growth algorithm also doesn't perform very good in that respect, but the to variations result in a well-distributed set of settlements.

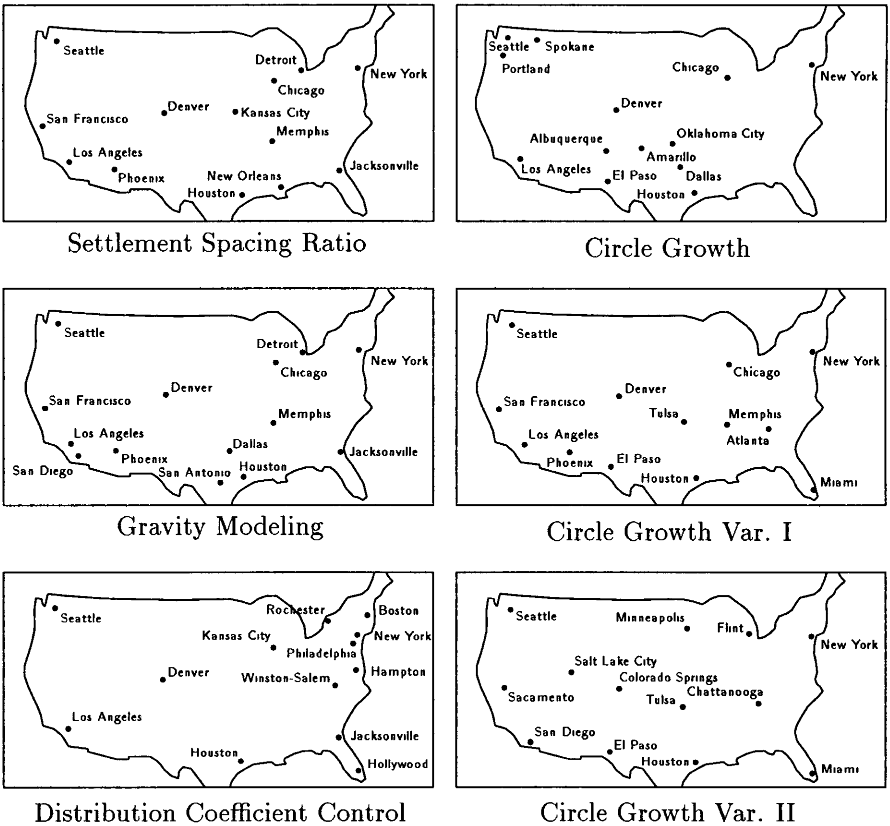

Figure 4: Results of the different methods

5 CONCLUSIONS AND FURTHER RESEARCH

We developed three new models for the settlement-selection problem and compared them with existing models. While the existing models compute a single selection, the new models determine a complete ranking of the settlements. After ranking, selecting any number of settlements is easy. Moreover, when selecting more settlements, all previously selected settlements remain selected, which is not the case in the existing models. The new methods allow efficient implementations, and result in evenly distributed selections compared to the existing models. Adjustments to the model can be made to select the most important settlements primarily, or make a very evenly distributed selection of the important settlements.

We are planning to investigate some more variations on the circle growth model, and to come up with better ways of fine-tuning the importance dependency of the various models. Another aspect we want to look into is the effects of panning and zooming on the selection. It would also be interesting to develop methods for selection of map features that are not represented by points, such as roads, lakes, and rivers.

REFERENCES

Aurenhammer, F. and Edelsbrunner, H. (1984). An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recogn.*, 17:251–257.

Flewelling, D.M. and Egenhofer, M.J. (1993). Formalizing importance: parameters for settlement selection in a geographic database. In *Proc. Auto-Carto*, pages 167–175.

Kadmon, N. (1972). Automated selection of settlements in map generation. *The Cartographic Journal*, 9:93–98.

Langran, G.E. and Poiker, T.K. (1986). Integration of name selection and name placement. In *Proc. 2nd Int. Symp. on Spatial Data Handling*, pages 50–64.

Mark, D.M. (1990). Competition for map space as a paradigm for automated map design. In *Proc. GIS/LIS'90*, pages 97–106.

Okabe, A., Boots, B., and Sugihara, K. (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, Chichester, England.

Töpfer, F.T. and Pillewizer, W. (1966). The principles of selection. *The Cartographic Journal*, 3:10–16.