

# Hierarchical Cluster Visualization in Web Mapping Systems

Jean-Yves Delort  
Macquarie University and Capital Markets CRC  
Sydney, Australia  
jydelort@ics.mq.edu.au

## ABSTRACT

This paper presents a technique for visualizing large spatial data sets in Web Mapping Systems (WMS). The technique creates a hierarchical clustering tree, which is subsequently used to extract clusters that can be displayed at a given scale without cluttering the map. Voronoi polygons are used as aggregation symbols to represent the clusters. This technique retains hierarchical relationships between data items at different scales. In addition, aggregation symbols do not overlap, and their sizes and the number of points that they cover is controlled by the same parameter. A prototype has been implemented and tested showing the effectiveness of the method for visualizing large data sets in WMS.

**Categories and Subject Descriptors:** H.5.2 [User Interfaces]: Graphical user interfaces (GUI); H.5.4 [Hypertext/Hypermedia]: Navigation

**General Terms:** Human Factors, Algorithms

**Keywords:** Visualization, hierarchical clustering

## 1. INTRODUCTION

Web-mapping systems (WMS) such as Google Maps and Bing Maps have become standard ways of sharing geographic information on the Web. Most WMS provide APIs to display maps in which users have the ability to zoom and pan interactively. These APIs also provide functionality to add layers of geographic information to maps. Although WMS are mostly used to visualize small data sets such as favorite places or personal photo albums, there is a growing need for WMS to support larger data sets. However, visualizing large data sets frequently causes technical scalability problems and clutters the map.

Cluttered maps are difficult to navigate for users as visual clutter not only obscures the background but also hinders the users understanding of the structure and content of the data. Hierarchical aggregation (HA) is a common visualization technique to make visual representations easily scalable and less visually cluttered [12]. In particular, HA techniques have been proposed for exploring spatial data sets [13, 17]. However, these techniques do not enable the selection from the hierarchical structure of clusters that can be displayed at a given scale without cluttering the map. A related approach to HA is regional aggregation where items are clustered using a pre-defined spatial subdivision, for exam-

ple into cities, regions/states, . . . However, this approach requires geographic knowledge and it heavily constraints clustering. A number of techniques have been proposed to reduce clutter in WMS [6, 7, 16]. A drawback of most of these techniques is that they focus on a given scale and do not retain hierarchical relationships at different scales.

In this paper, we present a visualization technique to reduce clutter in WMS based on hierarchical aggregation. The following section overviews existing techniques for visualizing large data sets in WMS. Section 3 describes our technique, and Section 4 presents preliminary results of an exploratory and a scalability analysis.

## 2. RELATED WORK

Three main types of clutter reduction techniques may be distinguished [11]: appearance (alter the look of the data items), spatial distortion (displace the data items in some ways) and temporal (animation). Appearance is the main approach for decluttering geographic objects. Indeed, since their geographic coordinates bind objects to the map, spatial distortion techniques can hardly be used. Most appearance techniques are based on filtering and clustering. Carmo et al. [7] filter objects on a map using a degree-of-interest function representing the importance of a point based on its apriori importance and its distance to the current center of the view. Burigat and Chittaro [6] have proposed a technique that identifies clusters of mutually overlapping icons and replace them by a selection of non-overlapping icons. Grid-based techniques can also be used to reduce clutter. For example, Girardin et al. [16] divide the view into a grid of squares the color of which depends on the density of items in the area.

The effectiveness of an appearance-based technique also depends on the relevance of the symbols used to represent the objects [4]. Techniques have been proposed that render clusters using icons [2], cells [16] and container shapes such as bounding boxes and hulls [8]. Space partitioning techniques, such as Voronoi tessellations that produce Voronoi polygons, have also been used to represent clusters [20]. Given a set of points, Voronoi polygons are polygons whose boundaries define the area that is closest to each point relative to all other points.

An alternative approach for visualizing large data sets is to create a bitmap representing the data, and to superpose it on the map as a layer of translucent tiles [14]. However, a problem with this approach is that user's interactions with the data items are harder to deal with, as they need to be handled at the pixel level. Furthermore, this approach does not reduce visual clutter.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.  
ACM 978-1-60558-799-8/10/04.

### 3. PROPOSED TECHNIQUE

Our technique for visualizing spatial data is threefold. First, points are clustered with respect to their distance on the map using single-linkage hierarchical clustering. The result is a binary tree representing a hierarchy of clusters where each node is the centroid of the data items in its subtree. Then, nodes that can be displayed on the map at a given scale without causing clutter are extracted from the tree. Finally, selected nodes are represented on the map.

#### 3.1 Node selection

The goal of the node selection step is to extract from the tree, nodes that can be displayed at a given scale without cluttering the map. Clutter is a difficult concept to measure, in particular because it is not only task and device-dependent but also subjective. For example, a map may look cluttered in the context of a target acquisition task where the objects displayed on the map are too close. To control for visual clutter, we introduce a threshold representing the minimum visual distance between two centroids. This threshold multiplied by the scale defines the minimum centroid distance (MCD). A node selection algorithm performs a depth-first search in the tree (Fig. 1). It only explores non-terminal nodes whose linkage distance is greater than MCD. It selects children of these nodes that are terminal or that are not terminal and have a linkage distance lower than MCD. Figure 2 illustrates the behavior of the algorithm. Selected nodes form an antichain, i.e., a subset of nodes such that any two elements in the subset are incomparable.

```

function nodeSelection(Node n, Float MCD)
  Sel := List[]
  if n is terminal then
    Append n to Sel
  else if  $d(n.left, n.right) > MCD$  then
    if n.left is terminal then
      Append n.left to Sel
    else if  $d(n.left.left, n.left.right) < MCD$  then
      Append n.left to Sel
    else
      Append items of nodeSelector(n.left, MCD) to Sel
    end if
    if n.right is terminal then
      Append n.right to Sel
    else if  $d(n.right.left, n.right.right) < MCD$  then
      Append n.right to Sel
    else
      Append items of nodeSelector(n.right, MCD) to Sel
    end if
  end if
  return Sel
end function

```

Figure 1: Node selection algorithm

An interesting property is that, any pair of two elements of the antichain has a distance of at least MCD. Indeed, let A and B be two nodes in the antichain extracted by the algorithm and P be the path of the tree connecting A to B. Clearly, there is a node G in P, the left and right children of which respectively dominate A (or B) and B (or A). The distance between the two children of G is necessarily greater than MCD (otherwise A and B would not have been selected by the algorithm). Since the tree is generated using single-linkage, the distance between the two closest elements in the children of a node is greater or equal to MCD. Thus, the

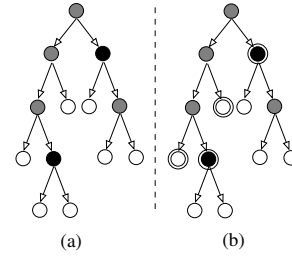


Figure 2: (a): a hierarchical tree where white nodes represent data items, and colored nodes represent cluster centroids. Nodes whose linkage distance is greater (resp. lower) than MCD are depicted by gray discs (resp. black discs). (b): nodes selected by the algorithm are double-circled.

distance between A and B is greater or equal to MCD which proves that MCD is the minimum distance between points of the antichain.

#### 3.2 Node visualization

The last step of the method consists in representing selected nodes on the map. We considered the following alternatives: icons, common container shapes and Voronoi polygons. Icons (and symbols) are useful for providing visual cues of what is in the space [4].

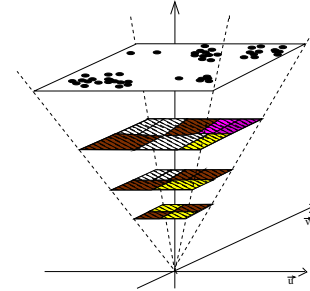


Figure 3: Space-scale diagram representing selected nodes by Voronoi polygons at different scales.

One of the main drawbacks of icons is that they do not show the area covered by the clusters. Container shapes do not suffer from this drawback, but they can create clutter if they overlap. Another problem is that dense and compact clusters covering small areas will be smaller and less visible than sparse clusters covering larger areas, which may mislead the user. This limitation may be reduced by using Voronoi polygons. Indeed, the minimum size of Voronoi polygons can be adjusted by controlling the space between nodes. A limitation of Voronoi polygons is that they do not necessarily contain (i.e., cover the area of) all the points in the cluster. However, items which have a distance to a node lower or equal to  $MCD/2$ , are covered by the corresponding polygon. Therefore, we decided on using Voronoi polygons to render selected nodes on the map. Note that Voronoi algorithms that take into account a weight for each region could also be used to improve the coverage [19].

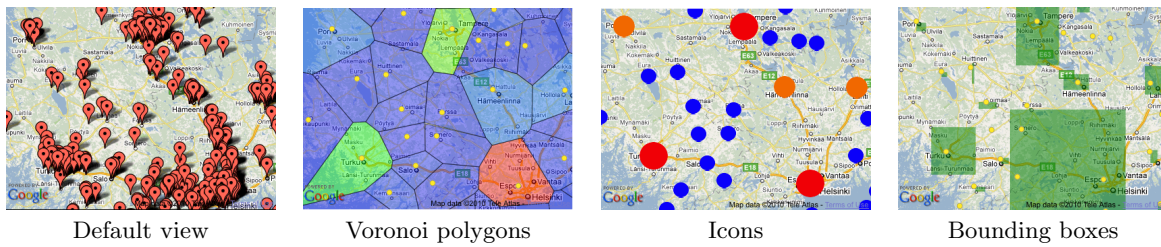


Figure 4: Default view vs clustered views using different types of footprints

### 3.3 Multiscale visualization

We now examine how nodes selected at different scales will be visualized in a multiscale user interface with a discrete number of zoom levels. We use space-scale diagrams [15] to show how centroids and data items are displayed at different scales. Space-scale diagrams are diagrams allowing the direct visualization and analysis of important scale related issues for interfaces.

Figure 3 shows a 2D diagram where the horizontal axes represent the original spatial dimensions while the vertical axis represents scale. The top level depicts all items of the data set at a given scale. Levels located below it represent views of the same data set at smaller scales using the proposed approach. Items have been hierarchically clustered and nodes have been selected to be displayed at different scales taking into account the MCD condition. Note that the level with the smallest scale is located below all the others.

## 4. EXPERIMENTS

This section our system prototype and preliminary results of an exploratory and a scalability analysis.

### 4.1 Experimental setup

In order to illustrate the different steps and the performance of the proposed technique, we designed a prototype and tested it with several real and generated spatial data sets. The prototype is accessible at:

<http://web.science.mq.edu.au/~jydelort/geoviz/demo.html>

The prototype has two main components. The first component clusters the data and generates the polygons. It takes as input a file containing a list of geographic coordinates, a list of scales and a visual threshold, and it outputs a file containing coordinates of polygons for each scale. The second component is a Google Maps mashup that displays the polygons. Coordinates of polygons are downloaded from a web server. When the user pans or zooms, the view is updated. Every time the current zoom level or geographical coordinates of the center of view are changed, they are sent to the server, which returns the coordinates of visible clusters. Voronoi polygons are computed using the computational geometry algorithm library CGAL [1].

For the exploratory analysis, we downloaded several KML files containing information about various topics from the Web. Keyhole Markup Language (KML) is an XML language focused on geographic annotation and visualization that has been widely adopted on the Web. Table 1 lists the type and size of two of these data sets.

Table 1: Type and size of used data sets

ID	Description	Size
D1	Bicycle stations in Paris	1208
D2	Participants to the Google Marketing Challenge	608

### 4.2 Exploratory analysis

Evaluating visualization techniques or systems is a well-known problem [10, 18]. Ellis and Dix [10] assert that, as generative artefacts (i.e., “things that are not something of value in and of themselves, but only yield results in some context”), the evaluation of visualizations is methodologically unsound. In many situations, it is impossible to undertake an evaluation capable of “proving” the effectiveness of a visualization technique. Such kind of evaluation would require too many tasks, data sets, implementations and users. Ellis and Dix distinguish between three types of evaluation: summative (i.e., comparison-based), formative (evaluation that leads to suggestions for improving the evaluated technique) and exploratory analysis (evaluation that is helpful to discover new ideas and concepts about the technique). Their sentiment is that exploratory analysis is the most effective approach for evaluating visualization techniques. The rest of this section presents preliminary results of an exploratory analysis of the visualization system.

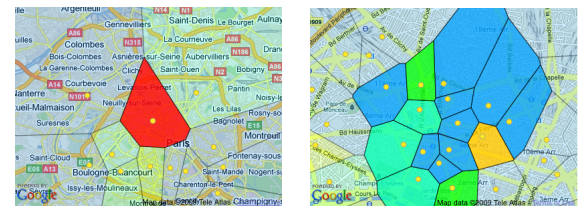


Figure 5: The same cluster at different scales

Figure 4 illustrates different types of cluster footprints of the same data set. While icons are easy to interpret, they poorly reflect the location of the data. Bounding boxes are also easy to interpret, but they can clutter the map. For example, in Figure 4, some bounding boxes are covering other cluster footprints, preventing the user from interacting with the clusters. One of the advantages of using Voronoi polygons as cluster footprints is that the user can effectively interact with the polygons as they cannot overlap. In addition, their minimum size may be controlled by MCD. Color is used to drive user’s attention on denser polygons. In the prototype, color is determined using a hot-to-cold color ramp where hot colors are assigned to dense clusters and cold colors to sparse ones [5].

**Table 2: Computation times for different data sets**

# points	Clustering	Selection	Generation	Overall
100	0s	0s	59s	59s
1000	3s	0s	487s	490s
5000	76s	7s	2099s	2182s

Using Voronoi polygons as footprints, the user may be confused by dramatic changes of polygons at different zoom levels. We propose the following solution to reduce this type of disorientation. When the user is interested in a particular region, he or she needs to select it which will highlight it (Figure 5, left). Then, when he or she zooms in or out, sub or super-clusters to that region will also be highlighted (Figure 5, right). Note that, this solution is only possible because clusters are hierarchically clustered.

The method also supports navigation by showing users the location of off-screen clusters [3]. Indeed, footprints may be displayed even if their centroids are located off-screen. However, if points are concentrated in small visible areas, polygons will cover a significantly wider area than the area of their points. In that case, using bounding boxes or hulls, as cluster footprints could be more effective.

### 4.3 Scalability analysis

This section discusses the processing time of the main components of the proposed technique. Issues regarding the computational complexity of the technique are addressed in [9]. For this experiment, we generate random geographical coordinates using a uniform distribution. We also need to control for the scattering of the data as scattering affects the clustering, the node selection, and the polygon generation steps. To control for scattering, we apply a scaling factor  $\lambda$  on the bounding box containing the geographic coordinates. For example, if the scaling factor is 1, data items in the generated data set can have coordinates all over the map, i.e. in the bounding box defined by  $(-178, -78) : (178, 78)$ . If the scaling factor is 0.5, then data items in the generated data set can have coordinates in the bounding box defined by  $(-89, -39) : (89, 39)$ . A scaling factor of 0.03 would roughly cover an area as large as France.

We generate data sets of size 100, 1000, and 5000 with  $\lambda = 0.03$ . For each set, we compute the processing time 1) to cluster the data items using single-linkage hierarchical clustering, 2) to extract nodes from the hierarchical structure for the 20 different scales supported by Google maps and 3) to generate Voronoi polygons at each scale. Table 2 reports the results using a 2GHz Intel Core 2 Duo with 2GB of RAM. For data sets with less than 1000 points, polygon generation is the most time consuming step while the processing time of clustering and selection are very low. For data sets with more than 1000 points, processing time of the clustering algorithm becomes a bottleneck.

## 5. CONCLUSION

In this paper we presented a technique for visualizing clusters of spatial data in interactive maps. The technique retains hierarchical relationships between data items at different scales. In addition, aggregation symbols do not overlap, and their sizes and the number of points that they cover is controlled by the same parameter. The main limitation of the method is the computational complexity of its hierarchical clustering algorithm. It prevents real-time processing

of large data sets ( $> 1000$  nodes). However, speed could be improved both on the method and the implementation sides. For example, data could be pre-clustered with a less time-consuming clustering algorithm (e.g., K-means) to speed up clustering.

## 6. REFERENCES

- [1] CGAL, <http://www.cgal.org/>.
- [2] Markercluster, <http://gmaps-utility-library.googlecode.com>.
- [3] BAUDISCH, P., AND ROSENHOLTZ, R. Halo: a technique for visualizing off-screen objects. In *Proceedings of the conference on Human factors in computing systems* (2003), ACM, pp. 481–488.
- [4] BERTIN, J. *Sémiologie graphique: Les diagrammes - Les réseaux - Les cartes*. Editions de l'Ecole des Hautes Etudes en Sciences, Paris, France, 1967.
- [5] BOURKE, P. Colour ramping for data visualisation, 1996.
- [6] BURIGAT, S., CHITTARO, L., AND GABRIELLI, S. Navigation techniques for small-screen devices: An evaluation on maps and web pages. *Int'l Journal of Human-Computer Studies* 66, 2 (Feb 2008), 78–97.
- [7] CARMO, M. B., FREITAS, S., AFONSO, A. P., AND CLÁUDIO, A. P. Filtering mechanisms for the visualization of geo-referenced information. In *Proceedings of the workshop on Geographic information retrieval* (2005), pp. 1–4.
- [8] CRISTANI, M., PERINA, A., CASTELLANI, U., AND MURINO, V. Content visualization and management of geo-located image databases. In *Proc. of the conference on Human factors in Computing Systems* (2008), pp. 2823–2828.
- [9] DELORT, J.-Y. Visualizing large spatial datasets in interactive maps. In *Proc. of the Int'l Conference on Advanced Geographic Information Systems, Applications, and Web Services* (2010).
- [10] ELLIS, G., AND DIX, A. An explorative analysis of user evaluation studies in information visualisation. In *Proc. of the 2006 AVI workshop on BEyond time and errors* (2006), ACM, pp. 1–7.
- [11] ELLIS, G., AND DIX, A. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1216–1223.
- [12] ELMQVIST, N., AND FEKETE, J.-D. Hierarchical aggregation for information visualization: Overview, techniques and design guidelines. *IEEE Transactions on Visualization and Computer Graphics* 99, 1 (2009).
- [13] ESTIVILL-CASTRO, V., AND LEE, I. Amoeba: Hierarchical clustering based on spatial proximity using delaunay diagram. In *Proc. of the 9th Int'l Symp. on Spatial Data Handling* (2000), pp. 7–26.
- [14] FISHER, D. Hotmap: Looking at geographic attention. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1184–1191.
- [15] FURNAS, G. W., AND BEDERSON, B. B. Space-scale diagrams: understanding multiscale interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), pp. 234–241.
- [16] GIRARDIN, F., CALABRESE, F., FIORE, F. D., RATTI, C., AND BLAT, J. Digital footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive Computing* 7, 4 (2008), 36–43.
- [17] GUO, D., PEUQUET, D. J., AND GAHEGAN, M. Iceage: Interactive clustering and exploration of large and high-dimensional geodata. *GeoInformatica* 7, 3 (2003), 229–253.
- [18] LIEBERMAN, H. The tyranny of evaluation, 2003.
- [19] OKABE, A., BOOTS, B., SUGIHARA, K., AND CHI, S. N. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.
- [20] PINHO, R., DE OLIVEIRA, M. C. F., MINGHIM, R., AND ANDRADE, M. G. Voromap: A voronoi-based tool for visual exploration of multi-dimensional data. In *IV '06: Proc. of the conference on InfoVis* (2006), IEEE Computer Society, pp. 39–44.