# Visualization and Analysis of Geographic Information: Representation Algorithms and Data Structures

J.Valença[1], L. Paquete[1], C. Caçador[2] and P. Reino[2]
1-CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
valenca@student.dei.uc.pt; paquete@dei.uc.pt;
2-SmartGeo Solutions; carlos.cacador@smartgeo.pt; pedro.reino@smartgeo.pt;

## Introduction

**The goal**
Given a starting set of Geographic Points, find a representative set of points for the region:

- Reduce visual information
- All points must be represented.
- Keep properties of the original set.

- Find the representative set in real-time.
- Allow for a panning movement over a larger region.

## Minimum Coverage Subset

Starting with a set $N$ of $n$ points and a fixed number $k$; find a centroid subset $P$ of size $k$, such that the element not in $P$ farthest from its closest centroid is minimized.

$$\min_{\substack{P \subseteq N \\ |P|=k}} \max_{n \in N} \min_{p \epsilon P} \|p - n\|$$

The solution to this problem is known as Minimum Coverage Subset and the problem itself is an example of a Facility Location $k$-center problem. This can be solved through Integer Linear Programming and Branch-and-Bound algorithms.

**Naive Incremental Branch-and-Bound:**
At each recursive step, decide wether a point is a centroid or not and branch accordingly. Update the objective function:

- a new centroid may change assignements and lowers the objective function
- a new non-centroid is assigned to closest centroid and raises the objective function

At each recursion step, calculate a lower bound for the best possible outcome for the solution of the current branch by assuming all remaining points are centroids.

**Results**: Both Branch and Bound implementations were too slow. The simple Integer Linear Programming performed better, but still unsuitable for real time use.

**Geometric Incremental Branch-and-Bound:**
Use an incremental approach, but build and maintain a Delaunay Triangulation between the centroids.

Use a greedy routing algorithm to quickly find the nearest centroid for new non-centroids. Use Hilbert Curves to pre-sort the points and further accelerate the process by heuritically reducing the searching path for the routing algorithm.

**Results**: Fixing the number of points does not output a good representative subset without previous knowledge of the region, and how many clusters it has!

## Fixed Minimum Distance

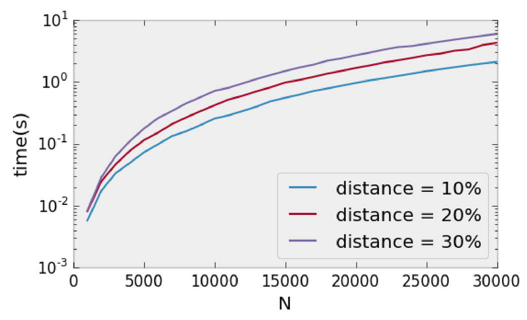
Starting with a subset of $n$ points and a minimum distance $d$. Find a subset of $k$ points such that no distance between two points is smaller than $d$ whilst minimizing $k$.

Optimal methods take too long to be used in real time so we use an approximation algorithm instead.

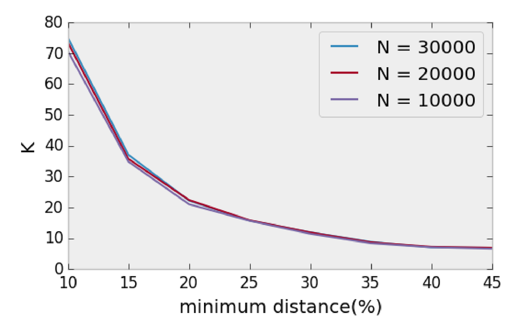
Using a k-d tree range search, we create a graph with edges connecting all pairs of points whose distance is smaller than $d$. This takes $O(n^2)$ time to perform.

Until the graph is empty, chose the point with the most neighbors to be selected and remove it and all its neighbors from the graph. Removing all points can be cast as the Set Cover, and this approach be performed in $O(n \log n)$ time.

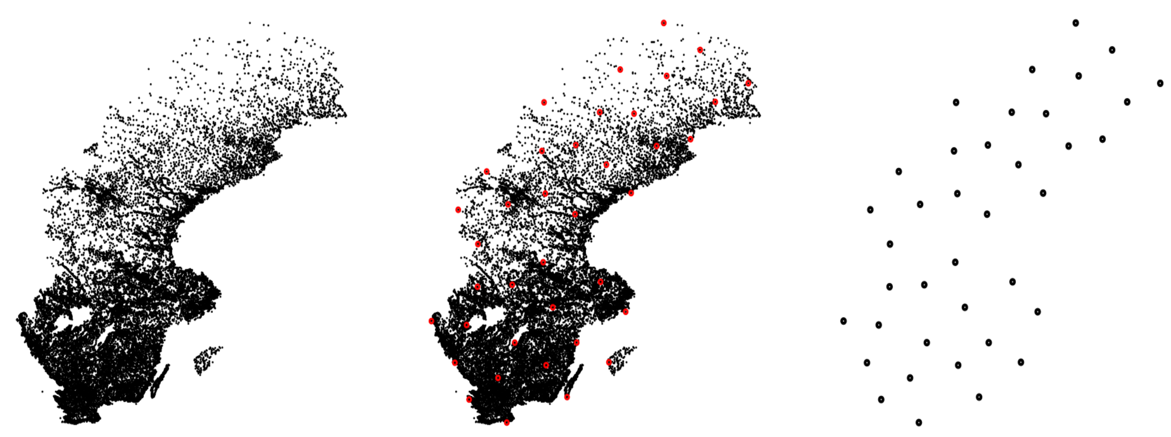
The selected set will have no more than $m \ln n$ elements, with $m$ being the optimal number for total coverage.




**Results**: This approach is fast enough to be used in real time. And depends on the established minimum distance as well as the density.




**Results**: The points chosen do not vary with the density. The algorithm can be sped up with a faster/preprocessed range search.



## Future Work

## Acknowledgements:

## References: