# A Method for High-Level Street Network Extraction of OpenStreetMap Data in OpenScienceMap

**Falko Schmid*, Hannes Janetzek****

*IRTG Semantic Integration of Geospatial Information, University of Bremen, Germany, schmid@informatik.uni-bremen.de
**Cognitive Systems, University of Bremen, Germany, hannes.janetzek@gmail.com

**Abstract.** In order to generate visually and conceptually meaningful digital maps for small scales, it is necessary to select only the most relevant topological information to be displayed. On zoom levels covering continents, countries, or states there are usually far too many cities and a too dense street network to be shown on small or medium size displays. However, the decision of which cities and network links to show on the map cannot be done by an attribute-based selection of features. An adaptive method is required to thin out information in densely populated areas while at the same time showing enough information for sparse areas. In this paper we introduce our approach to place and street network selection for mobile maps as implemented in OpenScienceMap.

**Keywords:** Small-scale Maps, Generalization, Multiscale Maps

## 1. Introduction

Digital topographic maps became ubiquitous on the Internet and all kind of mobile devices. As the maps are viewed on a large variety of devices with fundamentally different display sizes, it is required to thin out and generalize the street network especially on low zoom levels covering continents, countries, states. This is required to reduce the amount of visual information and to enhance map legibility. Depending on the size of the display, we need to tailor the amount of information in the map to the visual interface: the smaller the display, the less information can be shown. As visual entities need space to be displayed and a certain size to be clearly visually discriminated by the user, the amount of information is also largely independent from the physical resolution of the display.

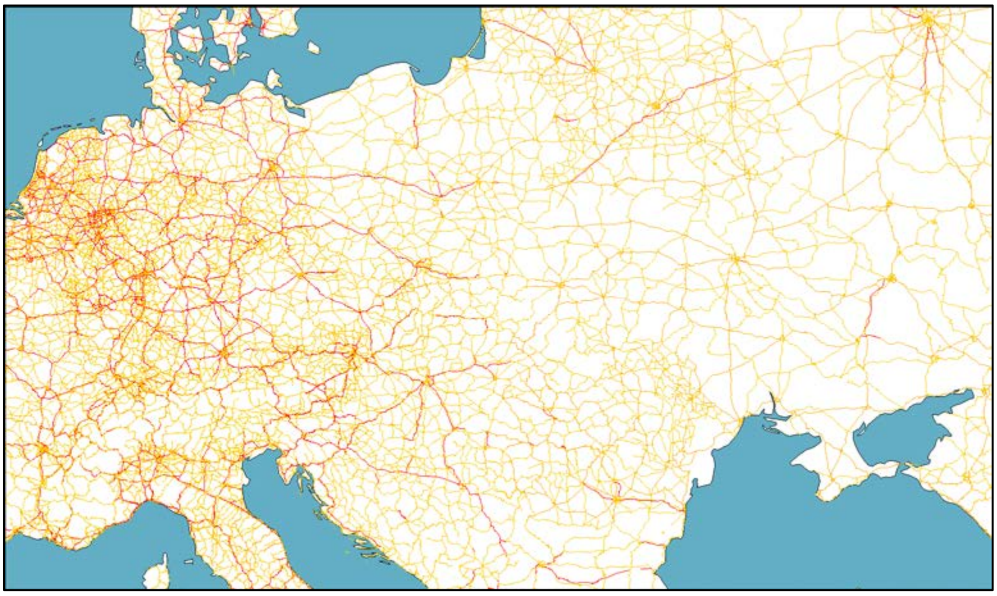

**Figure 1:** Street network only displaying *highway:motorway*,*highway:trunk*, and *highway:primary*. The network is too dense and cluttered for understanding significant connectivity and routes on the map, especially on small displays. There are too many individual entities displayed, as a consequence most of them are too small and too thin to be recognized without effort.

In dense regions in the world, e.g., parts of central Europe like the Ruhr-region in Germany, it is not possible to just apply a selection of entities based on a classification hierarchy. E.g., a naive heuristic is to display only settlements e.g., "cities" or places with a population of more than 100000 inhabitants on a given zoom level. Depending on the geographic area, the resulting set of places still can be far too dense, as all displayed places need to be labeled on the map and labels require space to be displayed. In contrast to that, in sparsely populated regions or regions with smaller places, too few representative places might be found and the areas can appear empty although they are populated. As a consequence, it is necessary to carefully select which places and corresponding labels are allowed to consume display space, and which places in turn get superseded by them. In addition to showing where places are, it is necessary to show the connectivity between them. A straightforward approach to select the street network is to only display streets of certain types. Similar to the amount of places, the resulting street network can be far too dense (see Figure 1). However, depending on the classification scheme and the actual street network, many of the displayed places will not be connected by a coherent network (see Figure 2).

Our method addresses both goals: a meaningful distribution of cities of varying sizes for dense and sparse areas at the same time, as well as the selection of a coherent street network between them independent from the network classification.

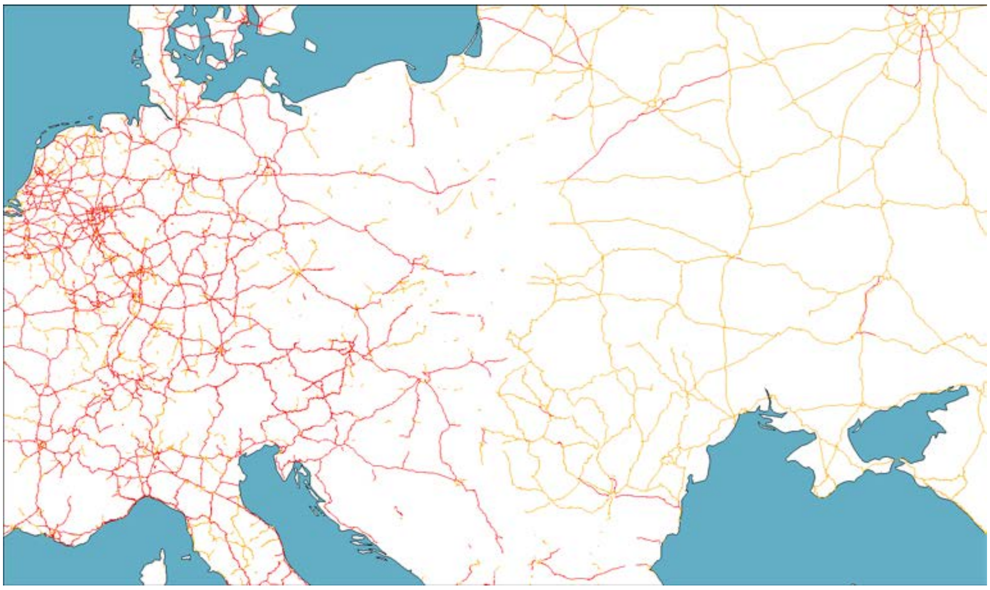

**Figure 2:** Disconnected street network in OpenStreetMap only displaying streets tagged with *highway:motorway*, *highway:trunk*. Not all places of e.g., Figure 4 are connected and the network is highly fragmented.

## 2. Related Work

The automatic selection of places and street network generalization is a major challenge in multiscale cartography. It always demands a rigorous reduction of the amount of places displayed on maps, which is typically done with clustering and/or ranking algorithms (e.g., Yan & Weibel (2008)). These places need to be connected with a street network which should be chosen such that significant connectivity and structural properties are preserved. In Chaudhry & Mackaness (2006) the authors develop method to generalize street networks for lower zoom levels according to the 'rules of good continuation' introduced by Bruce et al. (2003). Their model selects salient streets with respect to human visual perception. Touya & Duchene (2011) propose a complex method incorporating different strategies for different thematic generalization rules. However, their framework targets medium to large scale maps, where the variety of rules are applicable. In a recent work, Samsonov & Krivosheina (2012) develop an integrated algorithm combining Voronoi diagrams with a multi-criteria weight for the selection of places and a accumulative global route selection for the generation of the street network.

## 3. Our Approach

Our goal is to develop an integrated method for selecting representative places and a corresponding street network for higher zoom levels. Our particular focus is on an even distribution of reference places on small scales, incorporating settlements of varying sizes with the target of making geographic areas rapidly recognizable.

This work is implemented for our vector-tile project *OpenScienceMap*[1] (Schmid et al. 2013). *OpenScienceMap* itself is built upon the voluntary collected *OpenStreetMap*[2] (*OSM*) dataset. All classification notions in this paper refer to the tagging scheme and guidelines of *OSM*.

[1] http://www.OpenScienceMap.org
[2] http://www.osm.org

## 3.1 Identification of Relevant Places

The identification of suitable places as spatial references on a map is a fundamental requirement for topographic maps. As each selected place potentially supersedes a neighbored place with comparable properties (e.g., similar population, historical relevance, popularity, connectivity) we need a method to select a *good* candidate. In many cases there is an obvious *best* choice, however, in many other cases there are good reasons for displaying each of them. An example is the Rhine-Ruhr region in Germany with an agglomeration of cities of similar population, importance for the region, and popularity. In all cases the reasons for favoring a place over another is a mixture of algorithmically graspable parameters (e.g., population, network connectivity, geographic location, etc.) and rather random contextual information (e.g., tourism, infrastructure, culture, etc.).

In *OpenScienceMap* we select the places according to their *population* and *geographic location*. We use a population-based clustering of places and a street network computation based on Voronoi tessellation between candidate places: with the population-based clustering we identify the best candidate for a given geographic area, and with the tessellation we ensure a well-connected street network between the selected places.

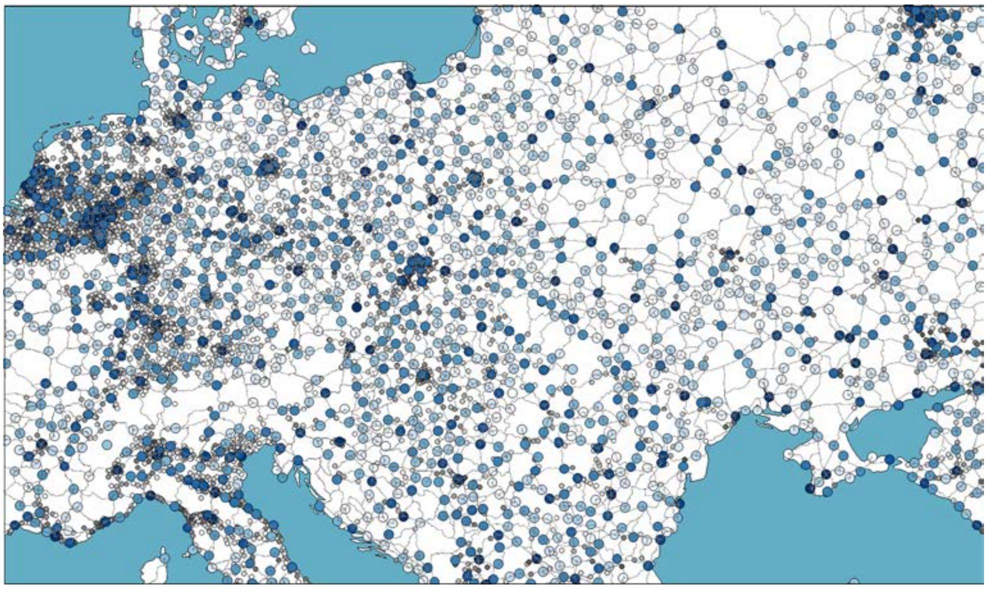

**Figure 3:** Result of SELECT-PLACES with a minimal population of 10.000 as a lower threshold. Places represented by large dots are selected, brightness indicates the size of the population (dark = large, bright = small).

## Selection of Reference Places

As we are interested in a geographically balanced distribution of places, it is not suitable to select places solely according to their classification or actual population. In *OSM* cities (*tag:place=city*) are defined as places with a population of 100.000 inhabitants and above. However, as the dataset is collected and annotated by volunteers, some of the tags and values are either not present or not correct (e.g., some users tag settlements with 10.000 inhabitants as cities). In this case we would not only obtain a set of places biased by agglomerations but also by wrongly annotated data in *OSM*.

In order to identify the best candidate for a given area in the map, we implemented SELECT-PLACES (see Algorithm 1). SELECT-PLACES clusters places based on a lower bound of population, independent from the type tagging (e.g., *tag:place=city* vs. *tag:place=town*) for a given portion of space on the map. A city is selected if there is no larger city within a given distance.

SELECT-PLACES works as follows: The input is given as a set of all places *places* represented by a tuple of *location* and *population*, and a minimal distance definition *minDist*. *minDist* defines a mapping of population ranges to distances: It is used to differentiate between less populated places that require a greater distance to other places to be selected compared to more populated ones. As a result of this definition, in areas with two equally large cities the two cities can be closer to each other than two cities with large difference in population; in this case the larger city will supersede the smaller one.

Figures 3 & 4 show the results of SELECT-PLACES with different parameterizations: Figure 3 shows the result of a lower threshold of 10.000 for input places, Figure 4 shows the result for 50.000 as the lower bound of the input. The result of SELECT-PLACES is a *candidate* set for the final map generation step. Although already rigorously reduced, the set might still be too large to be rendered along with the corresponding labels. The final selection of the places takes place in the applied labeling algorithm and its parameters (compare Figure 4 with the final maps in Figures 7 & 8).

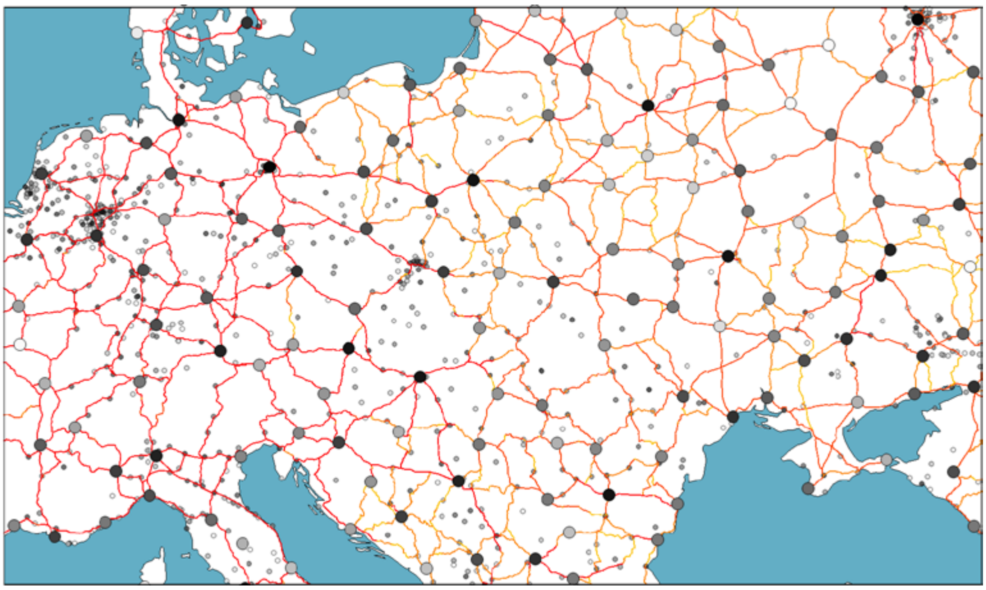

**Figure 4:** Result of SELECT-PLACES with a minimal population of 50.000 as a lower threshold. Places represented by large dots are selected, brightness indicates the size of the population (dark = large, bright = small).

| | |
|---|---|
| Algorithm: | SELECT-PLACES |
| Input: | Set of places where place is a tuple of (*location*, *population*)<br>minimal-distance definitions *minDist* |
| Output: | Reduced set of places |

```
result ← places
forall p1, p2 in asPairs(places):
if (population(p1) < population(p2):
  d = distance(location(p1), location(p2))
  if getMinimalDistanceForPopulation(minDist, p1) < d:
        result ← result ∖ {p1}
return result
```

**Algorithm 1:** Identification of reference places.

If a place *p1* has a smaller population than *p2*, and if the distance between them is smaller than the minimal distance for the population size as defined in *minDist*, we remove *p1* from the result set. This process is repeated for all pairs of places. This simple algorithm guarantees that all areas are represented by locally important places: places are selected if they are relatively larger than places in their surroundings. At the same time it guarantees an adaptive distribution of places with respect to population size *and* geographic area: it allows two or more equally large places to be displayed close to each other, while it ensures the selection of best reference places in sparsely populated regions. The minimal distance ranges in *minDist* need to be chosen according to the addressed map scale and display size: the smaller the display, the larger the defined distances needs to be; mutually preserved (visual) distance is the crucial parameter for the density of places on the map.

## 3.2. Network Generation Between Selected Places

To show the connectivity between the chosen places of algorithm SELECT-PLACES, we need to select a subset of the available street network between them. The network should be chosen such that the most important ways of the covered geographic area are represented. Although there exists a hierarchy of different types of streets (e.g. *highway:motorway*, *highway:trunk*, and *highway:primary* in *OSM*) it is not possible to just select a street or a series of streets between two places based on this classification. A street network is a complex mesh of different types of streets and in most cases it only is possible to connect two places with a combination of different street types. Reasons can be found in the physical street network itself, the geographic definition of a place in a dataset, and the user-based tagging in the case of *OSM*.

In a complex network built upon different types of streets it is impossible to connect all places with streets of equal types. Aspects like topography, population density, or climatic conditions require adapting the network to the environment it is built in. In addition to the physical network, places have a geographic definition, usually located in their center. However, the centers of places are typically streets of local types (e.g., *highway:living_street*, *highway:tertiary*, or *highway:secondary*) which makes it necessary to connect heterogeneous types to form a coherent connection between two places. And finally the voluntary collection and tagging of data for *OSM* naturally introduces inconsistencies as the classification of streets is very often subject to individual perception and experience.

### Selection of Street Network Subset

To compute a representative street network, we need to identify the structurally most important links between the places shown on the map. In contrast to other approaches we do not compute all (shortest) routes between all pairs of places and select the repeatedly occurring ones. In our approach we have a rather *local* view on the network: we only search for links between adjacent places as all identified links will sum up to an interconnected network. Neighborhood in a point cloud (like the set of places returned by SELECT-PLACES) is a difficult concept, as the local density of the places has strong influence on all parametric approaches (e.g., distance-based neighborhood) and will result in highly varying network densities: the more places are classified as neighbored, the denser will the resulting network be. A more conceptual view on neighborhood will guarantee an evenly represented network without overrepresentation in dense and underrepresentation in sparsely populated regions.

Our concept of neighborhood is based on a Voronoi diagram of the selected places as identified in SELECT-PLACES. With Voronoi cells we can identify the adjacent places without any numeric parameters by browsing the adjacent cells of the corresponding diagram. Our algorithm COMPUTE-STREETNETWORK (see Algorithm 2) computes a street network based on the Voronoi diagram of the selected places by considering Voronoi neighborhood of $2^{nd}$ degree. This practice ensures the detection of large infrastructural links, which otherwise would not be reliably represented by just considering the direct neighborhood. An equal method is the application of a Delaunay triangulation of the point set of places. In this representation one can identify the same neighbors as in our Voronoi approach by navigating along the mesh of triangles.

| | |
|---|---|
| Algorithm: | COMPUTE-STREETNETWORK |
| Input: | Set of places (*places*, *voronoi-diagram*, *street-network*) |
| Output: | Reduced street network between *places* |

```
network ← {}
forall p_i in places:
  forall n_j in getNeighborCells(p_i, voronoi-diagram):
     network ← network ∪ computeRoute(p_i, n_j, street-network)
     forall n_k in getNeighborCells(n_k, voronoi-diagram):
      network ← network ∪ computeRoute(p_i, n_k, street-network)
return network
```

**Algorithm 2:** Identification of route candidates.

COMPUTE-STREETNETWORK works as follows: it requires a set of places *places* (e.g., computed by SELECT-PLACES), the Voronoi diagram *voronoi-diagram* for them, and the corresponding source street network *street-network*. We now iterate through all places and identify their direct Voronoi neighbors. For each identified neighbor we compute a route and add it to *network*, the resulting reduced street network. As we want to identify the most important network links, we implement the route computation with a hierarchical router preferring higher classified street types (e.g. *highway:motorway* > *highway:trunk* > *highway:primary* > *highway:secondary*). This converges to a completely connected network with a focus on main infrastructural links. We repeat the steps for all 2$^{nd}$ degree neighbors of the selected place to ensure the detection of indirect links such as large motorways crossing a region without being part of a local hierarchical shortest route (see Figure 5 for an illustration of 1$^{st}$ and 2$^{nd}$ degree neighbors). When all routes are computed COMPUTE-STREETNETWORK returns the subset of the input network (see Figure 6). To render the street network on the final map (see e.g., Figures 7 & 8), it is necessary to apply geometric line simplification algorithms.

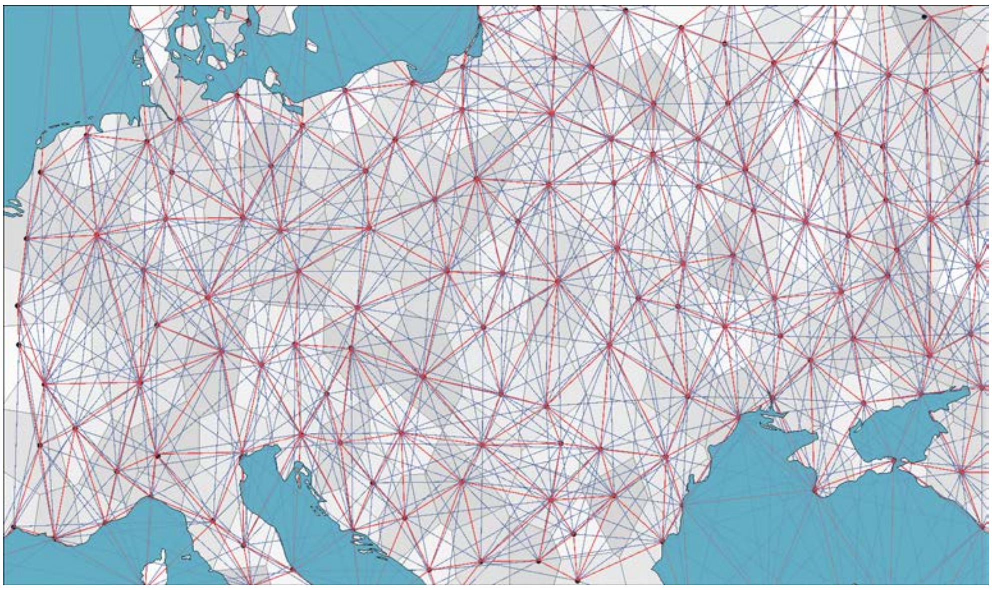

**Figure 5:** Voronoi-diagram for selected places, links between nearest neighbors are red, neighbors of 2nd degree blue. The extended neighborhood ensures the detection of important network links which otherwise would be neglected by direct adjacency.

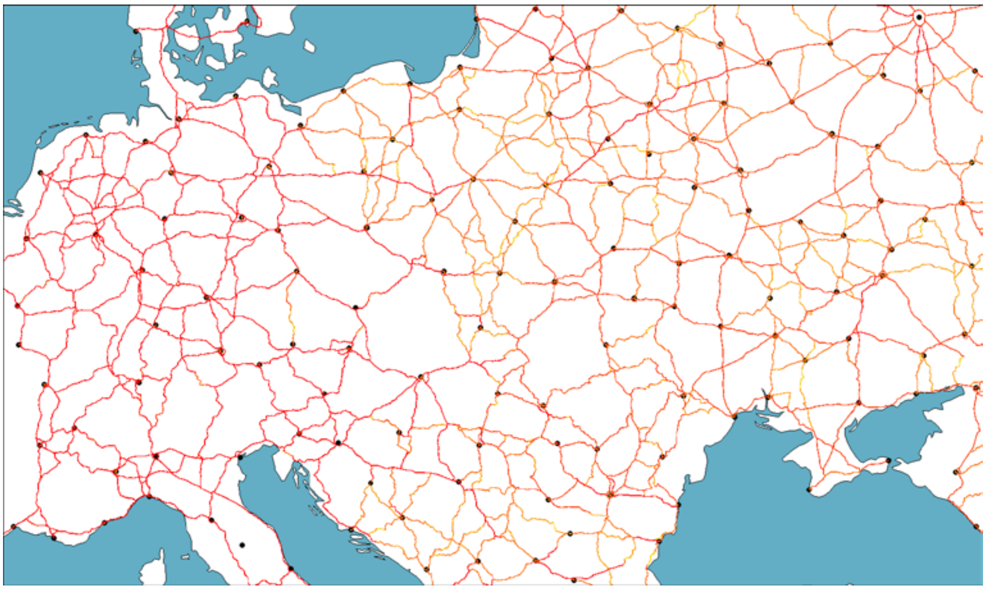

**Figure 6:** Result of the street network computation for the Voronoi diagram in Figure 5. The street network is homogeneously distributed across the map and at the same time sparse.

# 4. Conclusion

Digital multiscale maps require substantial generalization for small scales. This data reduction is necessary to make the displayed information understandable. One major challenge is to reduce the amount of displayed places and links of the street network between them. However, it is not possible to select places and streets only by their classification and fixed parameters, as the resulting information would be distributed unevenly across the map. In this paper we describe our approach for selecting reference places of varying sizes and a connecting street network for lower zoom levels such as continents, countries, and states.


**Figure 7:** Street network and reference places in OpenScienceMap on zoom level 4. The final labeling algorithm supersedes some of the places as computed in SELECT-PLACES.

Our approach consists of an adaptive clustering method for identifying reference places for geographic partitions in the map, and a Voronoi diagram-based method for determining the street network. Our results show an even distribution of places and a similarly evenly distributed street network across highly heterogeneously populated regions (e.g., western vs.

eastern Europe). Our work is implemented and available in the OpenScienceMap[3] dataset (see Figures 7 & 8).


**Figure 8:** Street network and reference places in OpenScienceMap on zoom level 7. The final labeling algorithm supersedes some of the places as computed in SELECT-PLACES.

## Acknowledgements

We gratefully acknowledge support of the German Research Foundation (DFG) under grant number GRK 1498.

## References

Bruce, V., Green, R. P. & Geogreson, A. (2003). Visual perception physiology, psychology, ecology. 4th edn, Psychology Press New York.

Chaudhry, O., & Mackaness, W. (2006). Rural and urban road network generalisation: deriving 1: 250,000 from OS MasterMap. Proceedings ICC 2005.

[3] http://www.OpenScienceMap.org

Samsonov, T. E., & Krivosheina, A. M. (2012). Joint generalization of city points and road network for small-scale mapping. Proceedings GIScience 2012.

Schmid, F., Janetzek, H., Wladysiak, M., & Hu, B. (2013). OpenScienceMap: open and free vector maps for low bandwidth applications. In Proceedings of the 3rd ACM Symposium on Computing for Development 2013.

Touya, G., & Duchêne, C. (2011). CollaGen: collaboration between automatic cartographic generalisation processes, Advances in Cartography and GIScience.

Yan H, & Weibel, R. (2008). An algorithm for point cluster generalization based on the voronoi diagram. Computers & Geosciences 34(8): 939-954.