Technische Universität Dresden

Fakultät Umweltwissenschaften

Multiscale visualization approaches for Volunteered Geographic Information and Location-based Social Media

Dissertation zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

Vorgelegt von:

Mathias Gröbe, M.Sc.

Gutachter	Prof. Dr.-Ing. habil. Dirk Burghardt, Technische Universität Dresden
Gutachter	Prof. Dr.-Ing. habil. Jochen Schiewe, HafenCity Universität Hamburg
Gutachter	Prof. Dr. Pia Bereuter, Fachhochschule Nordwestschweiz

Datum der Einreichung: 05. Januar 2023

Datum der Verteidigung: 21. Juli 2023

Übereinstimmungserklärung

Die Übereinstimmung dieses Exemplars mit dem Original der Dissertation zum Thema:

Multiscale visualization approaches for Volunteered Geographic Information and Location-based Social Media

wird hiermit bestätigt.

....

Ort, Datum

....

Unterschrift (Vorname Name)

Abstract

Today, "zoomable" maps are a state-of-the-art way to explore the world, available to anyone with Internet access. However, the process of creating this visualization has been rather loosely investigated and documented. Nevertheless, with an increasing amount of available data, interactive maps have become a more integral approach to visualizing and exploring big datasets and user-generated data. OpenStreetMap and online platforms such as Twitter and Flickr offer application programming interfaces (APIs) with geographic information. They are well-known examples of this visualization challenge and are often used as examples. In addition, an increasing number of public administrations collect open data and publish their data sets, which makes the task of visualization even more relevant. This dissertation deals with the visualization of user-generated geodata as a multiscale map. The basics of today's multiscale maps—their history, technologies, and possibilities—are explored and abstracted. This work introduces two new multiscale-focused visualization approaches for point data from volunteered geographic information (VGI) and location-based social media (LBSM).

One contribution of this effort is a visualization methodology for spatially referenced information in the form of point geometries, using nominally scaled data from social media such as Twitter or Flickr. Typical for this data is a high number of social media posts in different categories—a post on social media corresponds to a point in a specific category. Due to the sheer quantity and similar characteristics, the posts appear generic rather than unique. This type of dataset can be explored using the new method of micro diagrams to visualize the dataset on multiple scales and resolutions. The data is aggregated into small grid cells, and the numerical proportion is shown with small diagrams, which can visually merge into heterogenous areas through colors depicting a specific category. The diagram sizes allow the user to estimate the overall number of aggregated points in a grid cell.

A different visualization approach is proposed for more unique points, considered points of interest (POI), based on the selection method. The goal is to identify more locally relevant points from the data set, considered more important compared to other points in the neighborhood, which are then compared by numerical attribute. The method, derived from topographic isolation and called discrete isolation, is the distance from one point to the next with a higher attribute value. By using this measure, the most essential points can be easily selected by choosing a minimum distance and producing a homogenous spatial of the selected points within the chosen dataset.

The two newly developed approaches are applied to multiscale mapping by constructing example workflows that produce multiscale maps. The publicly available multiscale mapping workflows OpenMapTiles and OpenStreetMap Carto, using OpenStreetMap data, are systematically explored and analyzed. The result is a general workflow for multiscale map production and a short overview of the toolchain software. In particular, the generalization approaches in the example projects are discussed and these are classified into cartographic theories on the basis of literature. The workflow is demonstrated by building a raster tile service for the micro diagrams and a vector tile service for the discrete isolation, able to be used with just a web browser.

In conclusion, these new approaches for point data using VGI and LBSM allow better qualitative visualization of geodata. While analyzing vast global datasets is challenging, exploring and analyzing hidden data patterns is fruitful. Creating this degree of visualization and producing maps on multiple scales is a complicated task. The workflows and tools provided in this thesis will make map production on a worldwide scale easier.

Zusammenfassung

Heute sind zoombare Karten Alltag für jeden Internetznutzer. Die Erstellung interaktiv zoombarer Karten ist allerdings wenig erforscht, was einen deutlichen Gegensatz zu ihrer aktuellen Bedeutung und Nutzungshäufigkeit darstellt. Die Forschung in diesem Bereich ist also umso notwendiger. Steigende Datenmengen und größere Regionen, die von Karten abgedeckt werden sollen, unterstreichen den Forschungsbedarf umso mehr.

Beispiele für stetig wachsende Datenmengen sind Geodatenquellen wie OpenStreetMap aber auch freie amtliche Geodatensätze (OpenData), aber auch die zunehmende Zahl georeferenzierter Inhalte auf Internetplatformen wie Twitter oder Flickr zu nennen. Das Thema dieser Arbeit ist die Visualisierung eben dieser nutzergenerierten Geodaten mittels zoombarer Karten. Dafür wird die Entwicklung der zugrundeliegenden Technologien über die letzten zwei Jahrzehnte und die damit verbundene Möglichkeiten vorgestellt. Weitere Beiträge sind zwei neue Visualisierungsmethoden, die sich besonders für die Darstellung von Punktdaten aus raumbezogenen nutzergenerierten Daten und georeferenzierte Daten aus Sozialen Netzwerken eignen.

Ein Beitrag dieser Arbeit ist eine neue Visualisierungsmethode für raumbezogene Informationen in Form von Punktgeometrien mit nominal skalierten Daten aus Sozialen Medien, wie beispielsweise Twitter oder Flickr. Typisch für diese Daten ist eine hohe Anzahl von Beiträgen mit unterschiedlichen Kategorien. Wobei die Beiträge, bedingt durch ihre schiere Menge und ähnlicher Eigenschaften, eher generisch als einzigartig sind. Ein Beitrag in den Sozialen Medien entspricht dabei einem Punkt mit einer bestimmten Kategorie. Ein solcher Datensatz kann mit der neuen Methode der „micro diagrams“ in verschiedenen Maßstäben und Auflösungen visualisiert und analysiert werden. Dazu werden die Daten in kleine Gitterzellen aggregiert. Die Menge und Verteilung der über die Kategorien aggregierten Punkte wird durch kleine Diagramme dargestellt, wobei die Farben die verschiedenen Kategorien visualisieren. Durch die geringere Größe der einzelnen Diagramme verschmelzen die kleinen Diagramme visuell, je nach der Verteilung der Farben für die Kategorien. Bei genauerem Hinsehen ist die Schätzung der Menge der aggregierten Punkte über die Größe der Diagramme die Menge und die Verteilung über die Kategorien möglich.

Für einzigartigere Punkte, die als Points of Interest (POI) angesehen werden, wird ein anderer Visualisierungsansatz vorgeschlagen, der auf einer Auswahlmethode basiert. Ziel ist es dabei lokal relevantere Punkte aus dem Datensatz zu identifizieren, die im Vergleich zu anderen Punkten in der Nachbarschaft des Punktes verglichen nach einem numerischen Attribut wichtiger sind. Die Methode ist von dem geographischen Prinzip der Dominanz von Bergen abgeleitet und wird „discrete isolation“ genannt. Es handelt sich dabei um die Distanz von einem Punkt zum nächsten mit einem höheren Attributwert. Durch die Verwendung dieses Maßes können lokal bedeutende Punkte leicht ausgewählt werden, indem ein minimaler Abstand gewählt und so räumlich gleichmäßig verteilte Punkte aus dem Datensatz ausgewählt werden.

Die beiden neu vorgestellten Methoden werden in den Kontext der zoombaren Karten gestellt, indem exemplarische Arbeitsabläufe erstellt werden, die als Ergebnis eine zoombare Karte liefern. Dazu werden die frei verfügbaren Beispiele zur Herstellung von weltweiten zoombaren Karten mit nutzergenerierten Geodaten von OpenStreetMap, anhand der Kartenprojekte OpenMapTiles und OpenStreetMap Carto analysiert und in Arbeitsschritte gegliedert. Das Ergebnis ist ein wiederverwendbarer Arbeitsablauf zur Herstellung zoombarer Karten, ergänzt durch eine Auswahl von passender Software für die einzelnen Arbeitsschritte. Dabei wird insbesondere auf die Generalisierungsansätze in den Beispielprojekten eingegangen und diese anhand von Literatur in die kartographische Theorie eingeordnet. Zur Demonstration des Workflows wird je ein Raster Tiles Dienst für die „micro diagrams“ und ein Vektor Tiles Dienst für die „discrete isolation“ erstellt. Beide Dienste lassen sich mit einem aktuellen Webbrowser nutzen.

Zusammenfassend ermöglichen diese neuen Visualisierungsansätze für Punktdaten aus VGI und LBSM eine bessere qualitative Visualisierung der neuen Geodaten. Die Analyse riesiger globaler Datensätze ist immer noch eine Herausforderung, aber die Erforschung und Analyse verborgener Muster in den Daten ist lohnend. Die Erstellung solcher Visualisierungen und die Produktion von Karten in verschiedenen Maßstäben ist eine komplexe Aufgabe. Die in dieser Arbeit vorgestellten Arbeitsabläufe und Werkzeuge erleichtern die Erstellung von Karten in globalem Maßstab.

Table of Contents

Figures

Tables

List of Abbreviations

API *Application Programming Interface*
CaGIS *Cartography and Geographic Information Science*
JLBS *Journal of Location Based Services*
JSON *JavaScript Object Notation*
LBS *Location-based services*
LBSM *Location-based social media*
MVT *Mapbox Vector Tiles*
OGC *Open Geospatial Consortium*
OWS *OpenGIS Web Services*
POI *Point of Interest*
REST *Representational State Transfer*
SVG *Scalable Vector Graphics*
TCJ *The Cartographic Journal*
TMS *Tile Mapping Service*
UGC *User-generated content*
VGI *Volunteered Geographic Information*
W3C *World Wide Web Consortium*
WCS *Web Coverage Service*
WebGL *Web Graphics Library*
WFS *Web Feature Service*
WMS *Web Map Service*
WMTS *Web Map Tile Service*

1 Introduction

Maps are ubiquitous and always available in today's world. Many people use smartphones to organize their lives, and each smartphone comes with a map application (app) such as Google Maps, Apple Maps, or Bing Maps. The apps include routing for different purposes, reviews, and photos for points of interest (POI), all of which are useable through the concept of multiscale maps. If desired, it is possible to interactively change the map's scale and the extent to explore the details.

A milestone in this course of development to the current state of multiscale mapping technology was the introduction of the iPhone and its software ecosystem by Apple in 2007 (Cohen 2007). Previous devices had similar features but did not offer the easy installation of additional applications through an app store. Today, smartphones offer multi-touch functionality and browsers that can access all websites, not just specially designed sites (Wikipedia 2021c). The map app that comes with the smartphone allows the user to change the map extent scale for the whole world with multi-touch gestures. With at least 1.3 billion smartphones sold in 2020 (Goasduff, 2021), around a quarter and a half of the world's 8 billion people should be able to use digital maps on their mobile devices. That is potentially more map users than ever before! However, how do those maps work in daily life?

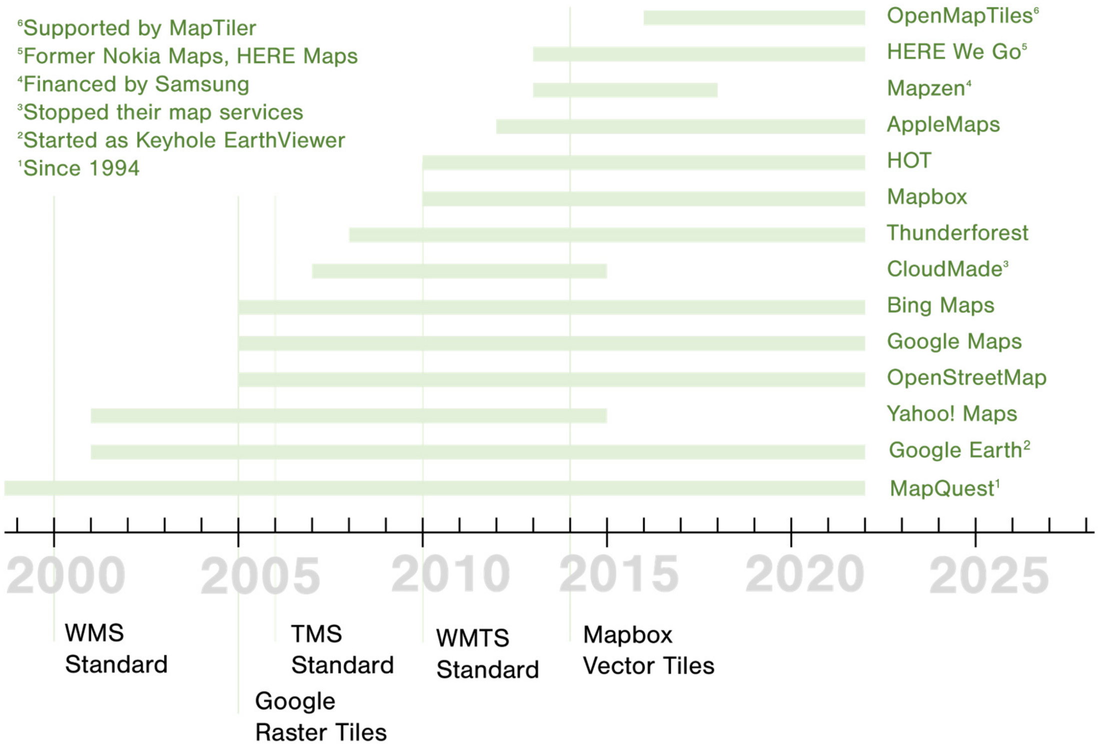


Figure 1: Timeline of technologies in use for multiscale mapping, and companies or organizations playing an important role in map creation or production in the last two decades.

A starting point for the advancements in this technology was the web map service (WMS), with version 1.0 standardized in the year 2000 (OGC, 2000) and already relevant for professional geographic information system (GIS) users then. However, this is not the technology of our everyday map for several reasons. Figure 1 shows the development of technologies and map providers over the last two decades. There were different maps services before Google Maps, but they did not reach such a broad audience (Veenendaal, Brovelli, and Li 2017; Kilday 2018). It was not until 2005 that the currently well-known approach of using tiles, zoom levels, and Web Mercator projection for multiscale mapping by Google began. The free access for everyone, including to the JavaSript source code, made re-engineering easy. It was not long before Yahoo! Maps, Bing Maps, and OpenStreetMap adopted the technology (cfis 2006), which was finally standardized as the tile mapping service (TMS) by the Open Source Geospatial Foundation in 2006 (OSGeo 2012a). Later, the approach was combined with the already known WMS, creating the web map tile service (WMTS) standard (OGC 2010).

Mapbox drove further technology development with the vector tiles (Mapbox [2014] 2021). Standardization through the Open Geospatial Consortium (OGC) is still in progress (OGC [2019] 2021). According to Figure 2, the introduction of vector tile technology resulted in some companies, like CloudMade and Yahoo! Maps, disappearing as map providers. With the current setup, it is possible to create high-quality and detailed visualization on different scales of the whole earth.

What is the source of those worldwide map services? A motivated crowd plays an essential role in the process, no matter if the app is Google Maps or OpenStreetMap. Google asks users to provide information about tourist spots, restaurants, and the opening hours of those locations. This has become an integral part of the Google Maps service, and the map data is official geodata. Concurrently, most other providers use OpenStreetMap for the map. So one way or another, today's maps need user contributions to stay current and to offer relevant information.

The concept of crowdsourced geodata or geo-related data matters here; this is a special kind of user-generated content. Users can edit and update geographic data that is later rendered as a "zoomable" map, resulting in a very detailed database for geographic information. Often, this data is called volunteered geographic information (VGI) (M. F. Goodchild 2007). Other data sources are location-based services and sometimes social networks (Zheng 2011). Some networks, e.g., Twitter or Flickr, offer data through an application programming interface (API) and use the collected geo-information for a wide range of analyses. Location-based social media (LBSM) is the technical term for those data.

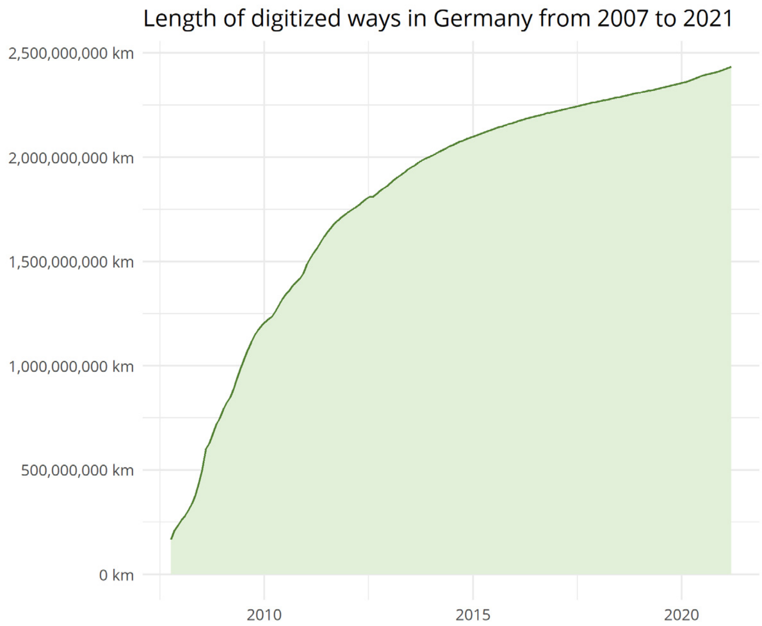


Figure 2: Kilometers of roadway tagged as “highway=*” in the OpenStreetMap database from 2007 to 2021

As Figure 2 shows, there has been a continuous contribution of user-generated data in OpenStreetMap. The OpenStreetMap (OSM) project was initiated in 2004 (“History of OpenStreetMap – OpenStreetMap Wiki” n.d.) and seemed to have reached good coverage in Germany, as shown in the diagram. Since 2012, it has focused more on other users because Google Maps introduced a pricing schema to integrate its map service into other websites or projects (Lardinois 2012).

Another aspect of the new data source is the speed of communication and change. Users can react to events on different platforms, deliver information about the event itself, or share their opinion (Dunkel et al. 2019). The Humanitarian OpenStreetMap Team (HOT) is an example of how professional this response can be. A nonprofit organization created after the 2010 earthquake in Haiti, HOT organizes mapping in during disaster events to assist responders with up-to-data geodata (“Humanitarian OSM Team – OpenStreetMap Wiki” n.d.). OSM is a valuable platform, allowing many people worldwide to provide edits for a known environment, contributing to a minutely updated map that is being further explored within the scientific community (Scholz et al. 2018).

1.1 Motivation

Crowdsourced geodata has new properties, creating the need for further analysis and visualization methods. The properties of this new data are similar to the “4 Vs” of big data—variety, veracity, volume, and velocity create new challenges for a cartographer (Schroeck et al. 2012; Tsou 2015), and visualization methods must address those characteristics.

Figure 3 is an example of LBSM visualization derived from a month of georeferenced tweets. It shows about a million points, color-coded by the language of the tweet, and filtered by the most spoken languages in the map extent. More than twenty languages have been hidden to create a cleaner look. Scaling this number to a year and worldwide coverage shows that this is an example of the new data's 4V properties. Nevertheless, this is still just a tiny piece of the actual data because only 2 percent of the tweets have a georeference (Leetaru et al. 2013), which in Europe means about 1,000,000 tweets per day with an available coordinate.

The visualization might be impressive, but it is also an excellent example of how this practice can help gain new knowledge. Visualization provides an overview of the data distribution and shows details, such as languages distinct to European national borders. There is no other known data source that can produce similar results. Examples from Eric Fisher demonstrate the potential by showing the languages of tweets and details from locals and tourists from Flickr (Fischer, 2011; The Geotaggers' World Atlas #5: Berlin, 2015). For those maps, a new kind of visualization technique shows each point in a non-aggregated form for such a massive dataset. Figure 3 shows all tweets with a small amount of transparency, color-coded according to the language of the tweet. As a result, the points overlap in areas with high data density, with the color becoming more intense than in areas with fewer data points. Additionally, it is possible to see which language is used in which region, most often on Twitter.

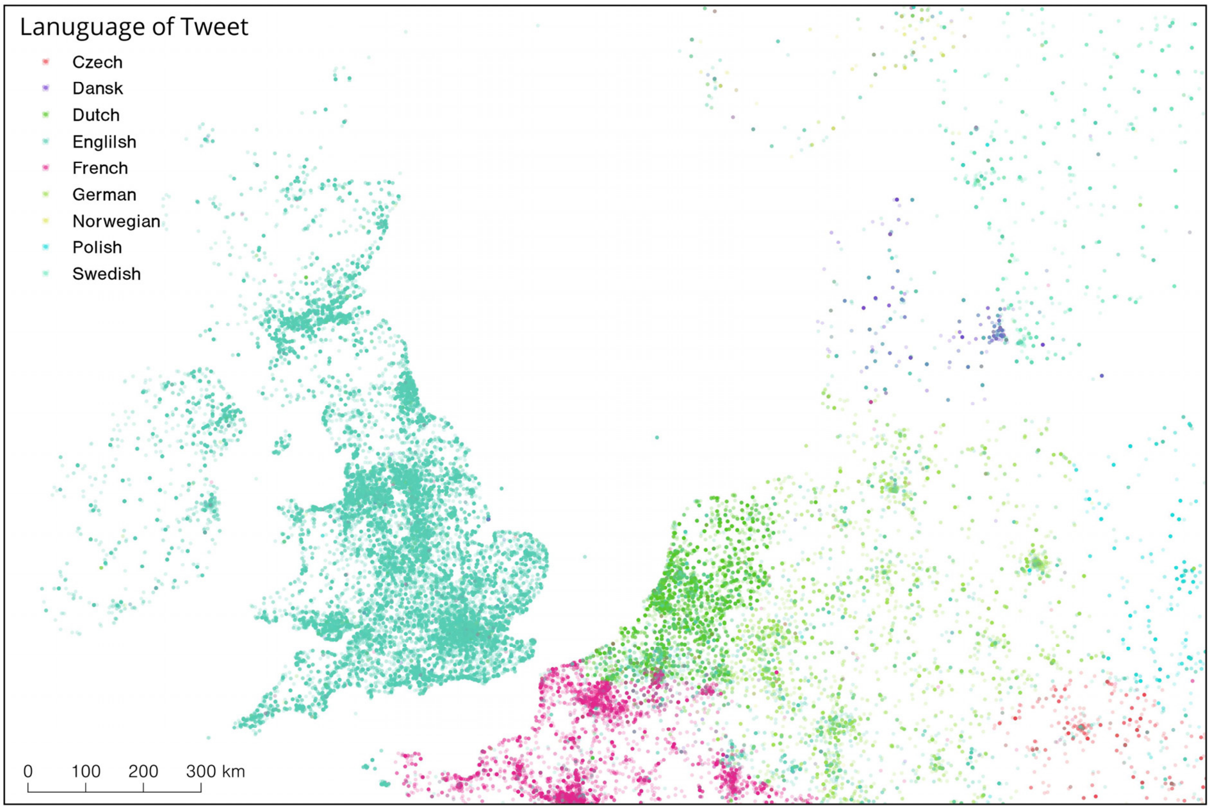


Figure 3: Visualization of tweet languages for part of Europe using the alpha blending technique. It is possible to recognize the landmasses and the borders just from the language of the tweets.

It is similar to the technique that used Eric Fischer's language map because there are no other possibilities for showing the volume and variety of the crowdsourced geodata. This amount of points data needs a new visualization method. A static map on one scale cannot display all of the data details. It is necessary to work on several scales to explore the data. The classic scale series in print map production is the zoom level in multiscale map production. Zoom levels are a set of predefined scales and resolutions bound to a projection, usually found in Pseudo Mercator projection. With the combination of the Figure 3 technique and a zoomable map, it is possible to explore all the details of a worldwide dataset—two indigents deal with the new properties of the crowdsourced geodata: *new visualization methods* and *multiscale visualizations*.

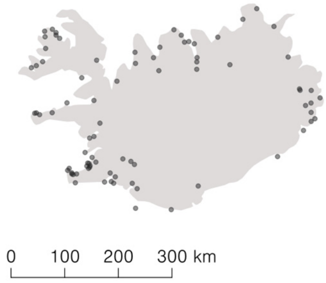


Figure 4: All populated places in Iceland from OpenStreetMap. To obtain an overview, a selection is necessary. (Data OpenStreetMap Contributors 2021 OdbL 1.0)

This type of visualization can facilitate data understanding and analysis or present the results of an analysis. This approach would also work well in *geovisual analytics* (Dransch, Sips, and Unger 2019) because visualization is necessary to explore the data and draw conclusions. This leads further to the *sensemaking loop* (Thomas and Cook 2005), which means a visualization is built on one understanding of the data and rebuilt with obtained knowledge to explore more details. In this manner, a data source can be explored piece by piece.

1.2 Visualization of crowdsourced geodata on multiple scales

After the need for new visualization methods became apparent, which worked well on multiple scales, the concrete visualization tasks followed the derivation of the research question.

Figure 5: Example map for Iceland showing a poor selection that considers only the population number. (Data OpenStreetMap Contributors 2021 OdbL 1.0)

The choice of visualization depends on the purpose, target group, and data. Due to the data-driven nature of this writing, the influence of the new data is the aspect that should be explored as a theoretical foundation. Data from location-based networks such as Twitter often create point data. That data is not comparable with the point features in OpenStreetMap, as LBSM points are more generic than most points for places in OpenStreetMap. According to (Bereuter 2015, 24), these different point data can be divided into point collection and POIs.

An example of a POI can be a place, a peak, or a restaurant, all of which have a specific unique identity, absolute position, and quantitative attributes. A tweet or Flickr photo is less unique, more abstract, and reproducible—it is possible to retake the same photo or write the same tweet. While point collection features have similar attribute values, such as language, time, creation app, and user time zone for Twitter, POIs have unique attributes, such as name, population, or elevation.

1.2.1 Research objective 1: Visualization of point collections

Point collection features can be categorized by the time, the language, and the app used for creation, as in Figure 3. The disadvantage of this technique is the mixed colors. While the overall color of the United Kingdom is prominent, the colors in some areas of continental Europe—such as Belgium—are intermingled. Nevertheless, these areas are also the most interesting, raising questions such as "Which languages are used more?" and "Where are the borders of language use?" The visualization allows only a relative qualitative estimation of the number of tweets, but it is clear which areas have more tweets than others.

First research question

How to visualize point collection data with quantitative categorized attributes in as much detail as possible?

That challenge leads to the first research question: *How to visualize point collection data with quantitative categorized attributes in as much detail as possible?* First, existing visualization methods should be explored to answer this question. Afterward, the author's *micro diagram* method (Gröbe and Burghardt 2020) is classified according to prevailing visualization ontologies, with its main parameters explained in detail. A user study demonstrates that the micro diagram offers advantages over existing visualization. It should be proven that map users can estimate the number of visualized values in a micro diagram map in contrast to the point visualization method shown in Figure 3, for example.

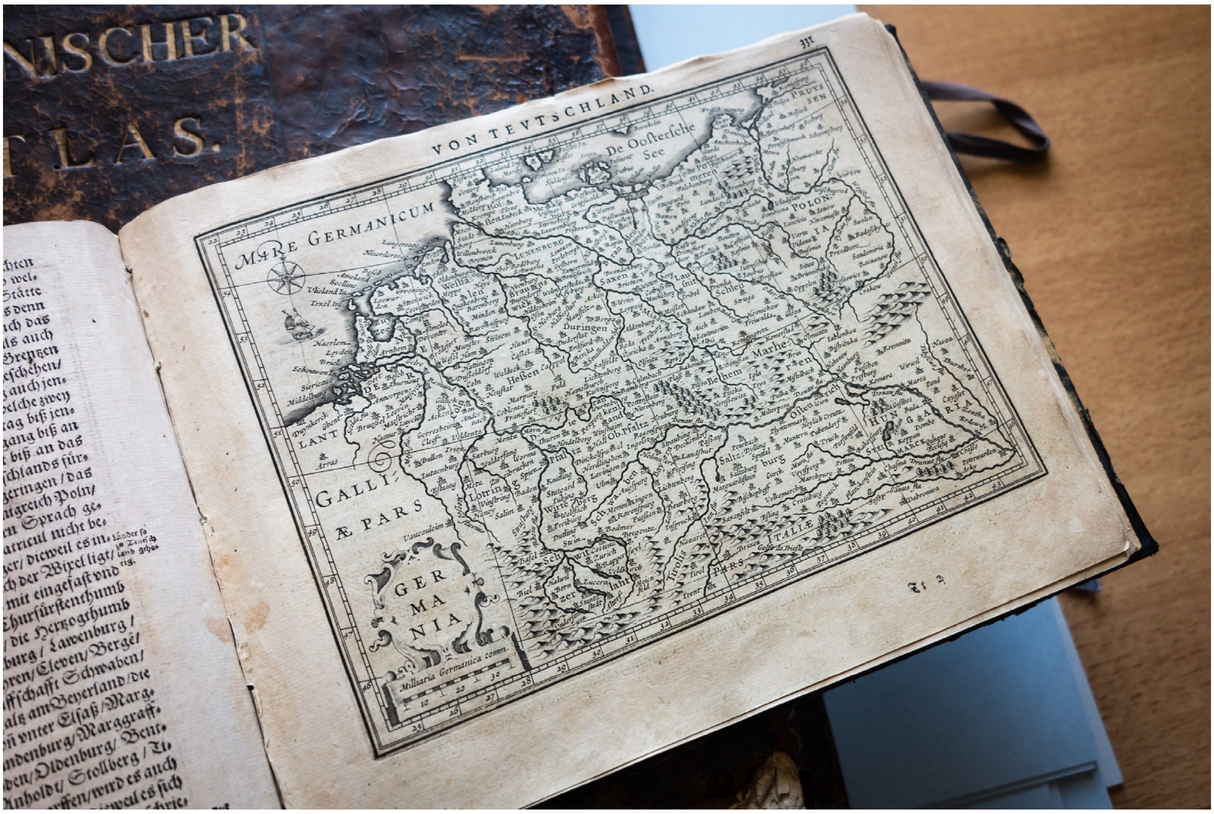


Figure 6: Photo of a map from an old atlas showing the area of Germany. The labels are nearly equally distributed, indicating locally important places for orientation and good use of the available space.

1.2.2 Research objective 2: Visualization of points of interest

Displaying all POIs, in the same manner, does not provide a good overview. It is better to visualize only the most relevant points, which requires an appropriate selection method. Figure 4 shows, as an example, populated places in Iceland, all of which are unique POIs. They have a name, a history, and a specific population, to name a few attributes. Usually, the population is the criterion of choice. The map in Figure 5 shows a negative example, where only the population number was considered. It seems necessary to consider the spatial distribution of the points for selection, similar to the old map in Figure 6. The result can be an informative map showing locally relevant places for better orientation.

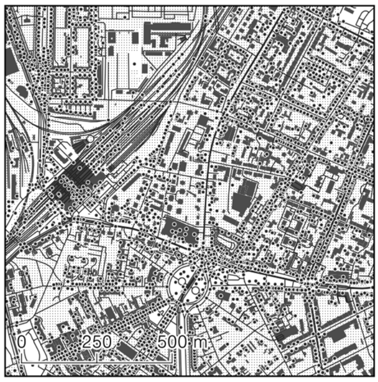


Figure 7: Example for the data density of OpenStreetMap in Dresden. All points, lines and polygons are shown. (Data OpenStreetMap Contributors 2021 OdbL 1.0)

This example leads to the second main research question: *How is a scale-dependent selection of POIs possible for visualization that considers spatial distribution and numerical attributes?* This task can be correlated with labeling the points but not with the focus of the research. Because maps today are usually zoomable and have several scales or zoom levels, e.g., OpenStreetMap, Google Maps, Bing Maps, or Mapbox, a suitable method for producing those kinds of maps is a relevant criterion. The vast map extent forces us to save computing time for an essential map update. This makes performance a critical measure for the selection methods. To fulfillment this task, the identified selection methods will be implemented similarly in order to make them comparable. A detailed comparison follows, including the visual results.

Second research question

How is a scale-dependent selection of POIs possible for visualization that considers spatial distribution and numerical attributes?

1.2.3 Research objective 3: Production of multiscale maps

Using maps is part of many people's everyday life, with at least one map app on every smartphone. Those apps offer maps in different resolutions, called zoom levels. Similar to the concept of scale, zoom levels are a set of scales in a Mercator projection, with a heavy distortion of lengths and areas that do not allow measurements using the scale. Nevertheless, these maps are global, easy to use, and usually combined with other location-based services (LBS). So far, the creation of these maps is less explored within the cartographic community, although there are a few available examples.

The OpenStreetMap project is unique in its data collection and worldwide coverage which as Figure 7 shows, results in enormous data density that can only be explored through a multiscale map. OpenStreetMap offers this type of map on its website with nearly real-time updates. There are several reasons to do this, including that contributors are motivated by seeing the map change and by finding errors in the visualization.

This map is extraordinary because of the community purpose, actual the real-time updates, and the style creation (Allan [2012] 2019). The data visualization is zoomable, making it possible and necessary to adapt the features and their styles to the needs of a multiscale map. In addition, different technologies—the raster and the vector tiling approach—are used to offer the map user good performance, which needs to be explained. Exploring,

Third research question

What are the possible workflows to produce a multiscale map, which is currently the most frequently used map?

abstracting, and applying the complex workflow behind today's maps is the third and last question: *What are the possible workflows to produce a multiscale map, which is currently the most frequently used map?*

For this task, a deeper look into map production is necessary, which is possible because of open source tools and workflow from OpenStreetMap Carto (Allan, 2019) and OpenMapTiles (Anon, 2021c). The results from both projects are comparable: a zoomable map with different scale levels. Nevertheless, there are differences in technology: OpenStreetMap updates by the minute and uses raster tiles with a fixed scale, while OpenMapTiles only performs a re-computation of its vector tile sets with stepless zooming. The used software will be classified, and similar products will be mentioned to offer a guideline for others who plan to create a comparable service.

Finally, the visualization approach for point collections and POIs from research questions one and two will be demonstrated in a multiscale environment. For instance, point collection data is a dataset of tweets and POIs of peaks or populated places from OpenStreetMap. The derived workflow from the third research question will be used to create a multiscale example. This will explain how today's maps are made and should demonstrate the applicability of the created knowledge under different technological aspects, such as vector- and raster-based web maps.

1.3 Reader's guide

This is a short explanation of the doctoral thesis structure. Some parts are already published, and this manuscript is a synopsis of the various works of the past year, connected by similar data sources and mapping on different scales and zoom levels.

1.3.1 Structure

The discussion starts with a general introduction of the new data sources, VGI and LBSN, their properties, and their resulting challenges. Also, the technical foundation of a multiscale map is part of this theoretical section, providing an overview of recent developments.

The following two chapters address visualization approaches for the different point data types. The new techniques visualize different kinds of point data and target multiscale visualization. Deeper insights into map production technology comprise the third chapter about creating multiscale maps and using the new technologies to formulate workflows for multiscale map production. Each chapter is dedicated to a research question and contains theoretical and methodical parts for the specific topic, use case, evaluation, and discussion. The last section summarizes the discussion and draws an overall conclusion. The Appendix offers additional tables and figures with further information, which is placed there to provide a better reading experience to keep the text compact. Table and Figure numbers set in this chapter start with an "A".

1.3.2 Related Publications

This dissertation refers to three different topics and their publications and, for some points, uses the same content:

I. **Gröbe, M.; Burghardt, D.:** Micro diagrams: visualization of categorical point data from location-based social media. In: Cartography and Geographic Information Science (2020), p. 1–16

II. **Gröbe, M.; Burghardt, D.:** Scale-Dependent Point Selection Methods for Web Maps. In: KN – Journal of Cartography and Geographic Information (2021)

III. **Hecht, R.; Artmann, M.; Brzoska, P.; Burghardt, D.; Cakir, S.; Dunkel, A.; Gröbe, M.; Gugulica, M.; et al.:** A Web App to Generate and Disseminate New Knowledge on Urban Green Space Qualities and Their Accessibility. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences Vol. VIII-4/W1-2021 (2021), p. 65–72

The work here reproduces adjusted contents of the first two publications, while only the experience gained through web mapping workflow is included for the third publication. The meinGrün projects' experiences and knowledge strongly influenced the third research question chapter and, therefore, is named here.

Information box

Offers additional information to clarify the complexities of the topic. Less scientific and more practical information.

Findings box

The green boxes help quickly locate important findings and provide summarized information.

1.3.3 Formatting and layout

This manuscript uses a three-column layout for easier reading. The outer column presents small images and information boxes. There are two types of information boxes: boxes with a green background for significant findings and boxes without background color for additional information that could be helpful for the reader. Footnotes primarily provide links to general websites for tools or projects. Web sources are regular citations listed in the references with all relevant information.

1.3.4 Online examples

The results of the implantation of multiscale map production workflows are available online. It shows examples of the micro diagram's visualization method and the discrete isolation selection approach. Some adjustments were made to serve them cost-efficiently and securely. Due to these changes and a small target group, the performance is slightly reduced. QR-Code and links are also provided in the particular section. The service will be maintained for a more extended time under the following URLs:

- Raster workflow example for the micro diagram approach: https://mathias-groebe.de/demo/raster-demo.html
- Vector workflow example for the discrete isolation: https://mathias-groebe.de/demo/vector-demo.html

2 Foundations of crowdsourced mapping on multiple scales

Visualization can facilitate understanding and analysis of data and present the results of an analysis. Geovisual analytics (Dransch et al., 2019) and the sensemaking loop (Thomas and Cook, 2005) of building a visualization can help explore user-generated data. Before doing this, it is necessary to understand where the data comes from and which visualization is suitable. The following chapter provides insights into the types of data and their visualization by going through its publication in the following selected journals: Journal of Location Based Services; The Cartographic Journal; and Cartography and Geographic Information Science. Somewhat separate from this topic, the technological foundations and history of multiscale maps are also described. This part aims to document the development of these technologies and clarify their current application before they are used in this thesis's practical parts.

2.1 Types and properties of crowdsourced data

Michael Goodchild introduced VGI as a case of user-generated content (M. F. Goodchild 2007). He names Wikimapia, Flickr, and OpenStreetMap as examples of this new kind of information generation. From today's perspective, VGI is seen in a more differentiated manner: Is Flickr data contributed by volunteers, or is it just a secondary product from people using a photo-sharing platform? Consider OpenStreetMap[1] contributors and the term "volunteered" information: What would be "non-volunteered geographic information" (Hormann 2015b)? Hormann proposes describing this as "crowdsourced geodata," which is a better description of the phenomenon. Goodchild formulates conflict from the concept of humans as sensors because humans do not want to work as machine-like sensors since they have their own opinions and motivations for why they are creating the data described as VGI, as the example from Hormann shows. Furthermore, the trend of location-based services and networks (Zheng 2011) has led to the creation of more geo-related and user-generated content.

2.2 Currents trends in cartography

A new challenge for cartographers is the amount of generated crowdsourced data that needs to be visualized for different purposes, from analysis to presentation. Previously, the data was collected and generalized for a map (Imhof 1972); today, there is so much data that new techniques must be developed to manage it. This is similar to the challenges mapping agencies face when producing topographic maps, which still require a great deal of manual work. For example, updates to data models at the German Federal Agency for Cartography and Geodesy (*Bundesamt für Kartographie und Geodäsie*) for different scales are still done manually (Mau 2018), while OpenStreetMap offers a worldwide, zoomable map that can be updated in a few minutes.

[1] https://openstreetmap.org/

There are currently efforts to bring those open source technologies and approaches to German national and federal mapping agencies. Starting with the TopPlusOpen project (Kunz 2018a; 2018b; Kunz and Bobrich 2019), the SmartMapping project aims to fully automize map production workflow within one day (Langhans and Wiemann 2021; Seifert 2021). This represents two interconnected aspects of this technological challenge—global data and data with several resolutions or scales. Users today expect to be able to use zoom on a digital map and that the map does not end at a national border. These reasons led to the TopPlusOpen project.

Moving back to the general idea, today, a map or a visualization of a dataset needs to be usable and explorable on several scales and not bound to a small area. This type of representation and its benefits are known and should be used to visualize and analyze user-generated geodata. Connecting classical map production for one scale, visualizing crowdsourced geodata, and applying multiscale mapping can help here.

2.3 Definitions

This work focuses on a special kind of user-generated content (Bauer 2010) with georeferencing. An understanding of the different data properties is essential for creating an excellent and meaningful visualization. An overview of the commonly used terms and variations in the user-generated data is provided. Overall, up to 60 percent of all information is geo-related (Hahmann and Burghardt 2013), which shows the importance of that information. In contrast for Twitter, with its billions of daily tweets, only a single-digit percentage of tweets have a coordinate or place model as an attached bounding box (Leetaru et al. 2013).

2.3.1 VGI

The term VGI was defined by Goodchild as a particular case of the "web phenomenon of user-generated content," enabled by Web 2.0 technology (M. F. Goodchild 2007). Different examples explain how users collaborate on the web to collect data, describing the motivation for and value of the contributions. It is always geographic data that is created, assembled, and provided voluntarily by individuals. This data can be images (Flickr), geographic data (OpenStreetMap), or entries in an encyclopedia (Wikipedia).

Volunteered geographic information (VGI)

Geographic data that is created, assembled, and provided voluntarily by individuals.

Over time, VGI broadened to include a wide range of applications and science (Sui, Elwood, and Goodchild 2013; Elwood, Goodchild, and Sui 2013). However, not all user contributions are valuable. Vandalism can occur in crowdsourced data for different reasons, such as frustration, politics, humor, or profit interest (Ballatore, 2014). Credibility was also a research topic (Flanagin and Metzger 2008).

2.3.2 LBSM

Data from social networking services are called LBSM, also described as location-based networks (LBSN). These are online services, platforms, or websites that allow users to communicate and share information with connected people (Lee and Ye 2017), such as Twitter, Facebook, Foursquare, and Flickr (N. Cao et al. 2012). These networks reflect the real social life of users through particular online platforms. Also, fitness and review/opinion applications, such as Tripadvisor,[2] Komoot,[3] and Strava,[4] are also considered social networks (Zheng 2011). After the social and network aspects are removed, these applications remain at their core location-based services, defined as "services that integrate a mobile device's location or position with other information to provide added value to a user" (Schiller and Voisard 2004, 10).

Location-based social media (LBSM)

Geographic-related content created by people using social media platforms.

Like VGI, LBSM is a data resource upon which research and applications are built (G. Cao et al. 2015; Chen, Lin, and Yuan 2017). A good shorthand for distinguishing between VGI and LBSM is the method of creation. VGI is more explicitly shared for another purpose, while LBSM content is created when a user shares or exchanges information using an internet platform. A more generic term would be user-generated content (UGC) or, more specifically, geo-related, user-generated content, which includes all new data sources.

2.3.3 Space, place, and location

The terms space, place, and location are often used to describe data related to geographic phenomena in the research context. It starts with the broadly used term space, becomes more defined with place, and reverts to abstraction as location. It is crucial to manage use of these terms by applying geographic information science tools to the collecting, modeling, managing, displaying, and interpreting of geographic data (Kemp 2008).

Space is a complex type of information, which can be considered the opposite of "time" (Elden 2009), but also as human achievements and human and social developments (Günzel and Kümmerling 2010). It is a fundamental concept of the science of geography. The concept of geographical space as space filled with relational and proportional qualities makes it possible to develop practical applications (Mazúr and Urbánek 1983; M. Goodchild and Li 2016). Coordinates and relations to other geographic features allow representation by points, lines, areas, and volumes (Gonçalves, Afonso, and Martins 2015).

A *place* represents a human or physical phenomenon that allows a location to be communicated (Scheider and Janowicz 2014; Tuan 1977; Winter and Freksa 2012). It can refer to a material or immaterial entity (McKenzie and

[2] https://www.tripadvisor.de/
[3] https://www.komoot.de/
[4] https://www.strava.com/

Adams 2017) and often combines different meanings (Cresswell 2009). Further research is being conducted to consider place-related modeling and analysis in geographic information science (Wagner, Zipf, and Westerholt 2020; Westerholt, Mocnik, and Comber 2020).

A geographic *location* refers to a spot on Earth's surface, defined with unlimited precision by coordinates within a reference system (M. F. Goodchild 1992; 2011) that can be represented in different types of data (Vector, Raster, TIN, Graph, etc.) and processed with a GIS (Church 2002).

2.4 Visualization approaches for crowdsourced geodata

The following section presents an overview of LBSM and VGI geodata and their visualization as a map. An overview of the data sources, the different kinds of data, and the connected visualization approaches are outlined to place this original research in context.

2.4.1 Review of publications and visualization approaches

Part of this review includes papers about geo-related, user-generated content with at least one map and naming the visualization data source. In presenting an overview of LBSM and VGI, there are no specific terms that can differentiate between the two without sounding like the same data is being described. Selected papers have used terms from two cartography journals and one LBS-related journal from 2007–2020, based on the definition of VGI by Goodchild:

- Journal of Location Based Services (JLBS)
- The Cartographic Journal (TCJ)
- Cartography and Geographic Information Science (CaGIS)

Table 2 summarizes the review by naming the publication year, journal, authors, data source, and visualization technique. In some cases, other data sources and visualization methods were also used but not included in the table in order to provide a better overview. A complete list of all papers can be found in Table in the appendix.

The visualization method categorization is based on the taxonomy of Kraak and Ormeling (2010). The heat map or heat line would be, in this case, a smooth statistical surface. A heat line means that the heat map approach to lines was applied; this was used in some publications (Cheng et al. 2010; Zhao 2015). Point maps represent a nominal point symbol map through the massive use of point symbols, and the micro diagrams represent area-related diagrams (Gröbe and Burghardt 2020). Another category is the space-time cube, which visualizes location and time in one graphic (Kraak and Ormeling 2010).

The research data sources are Twitter (61%), Flickr (24%), GPS tracks (18%), OpenStreetMap (10%), cell phone data (8%), taxi data (6%), floating car data (4%), Wikipedia (4%), and all listed papers. The LBSM definition

Heat map

A heat map shows point feature density with a yellow-orange-red color ramp that can be derived from a kernel density estimation, coloring the created raster or overlaying point symbols with suitable renderings. Aside from a geovisualization, a heat map can also be a type of statistical visualization, showing correlation through the color ramp.

Floating car data

Floating car data is typically time-stamped geolocalization and speed data collected from moving vehicles using the car's navigation app or the driver's smartphone. As a result, traffic jams can be detected or time-optimized routes predicted. It is also possible to derive this information from cell phone data or local sensors.

fits Twitter and Flickr. OpenStreetMap is VGI, as well as Wikipedia. GPS tracks are somewhere in between VGI, LBSM, and UGC. Some samples were explicitly created for research purposes (Krumm, Gruen, and Delling 2013; van Schaick 2010); in other cases, the samples were drawn from social or VGI platforms (Kessler 2011; John et al. 2017). The cell phone, taxi, and floating car data were user-generated and geo-related, with no direct social or volunteering aspect.

Most data is visualized as it is: point map (39%), nominal line symbol map (25%), and nominal point symbol map (6%). No aggregation or generalization is needed to generate such a map. The proportional symbol map (10%) is used chiefly in cartographic journals. A heat or line map (33 %) is sometimes combined with an isoline map (4%) and used mainly in cartographic journals. A particular case is a 2.5D relief (6%), which uses perspective to show the data in the third dimension, in combination with various color ramps.

Choropleth maps with administrative areas are usually more common (18%). However, for reviews, the grid choropleth map (25%) in particular, maybe a more accessible aggregation and avoid the modifiable areal unit problem effect (Madelin et al. 2009). Other particular cases are the grid-based micro diagram (8%) and the space-time cube (4%), which uses the third dimension to visualize movement over time.

2.4.2 Conclusions from the review

The geometry type of the data is, in most cases, a point. The position of a Flickr photo, a cell phone at a particular moment, a taxi drop, or a car's position are examples. It is possible to aggregate the points to lines using attributes such as usernames or, in the case of a GPS track, the point is already a line. User trips can be reconstructed using aggregation and some assumptions (Mamei and Colonna 2018; Rybarczyk et al. 2018). Aggregated visualizations can be derived from points and lines, such as the proportions symbol map or a surface interpolated to create a heat map that can be completed with contour lines. A specific example is OpenStreetMap or Wikipedia: contributors explicitly choose a specific geometry type to mode an object and create geodata. There is no general possibility of mapping the data because it can be a different geometry or thematic collection.

In most cases, the data in the reviewed publications are called VGI (25%), if there is any specification. UGC as an umbrella term is less used (16%). Overall, cartographic journals tend to use those terms, while publications like the Journal of Location Based Services rarely use VGI or UGC. One possible reason could be the different data sources. Another explanation could be the cartographic article author's intention to show the transferability of the presented approaches to UGC. In most publications, there is no differentiation between geometry types. The nature of the data is described as social media (Zhou and Zhang 2016; Rybarczyk et al. 2018; Wang and Ye 2019) or social network site data (Ghosh and Guha 2013), which is geotagged (Fujita 2013; Krumm, Gruen, and Delling 2013) or geolocated (Kounadi et al. 2018).

Table 1: Selected papers analyzing geo-related, user-generated content with map visualization. The background colors indicate the following journal abbreviations: Journal of Location Based Services (JLBS), The Cartographic Journal (TCJ), and Cartography and Geographic Information Science (CaGIS).

Publication			Data source								Visualization											Data	
Year	Journal	Paper	Flickr	Twitter	Wikipedia	OpenStreetMap	GPS tracks	Taxi data	Cell phone data	Floating car data	Point map	Nominal point symbol map	Proportional symbol map	Heat map or line	Isoline map	Choropleth map	Grid choropleth map	Micro diagrams	Nominal line symbol map	2.5D relief	Space time cube	UGC	VGI
2008	JLBS	Girardin et al.	x											x					x	x			
2010	JLBS	Andrienko et al.							x							x							
2010	TCJ	Cheng et al.								x				x									
2010	JLBS	Guo, Liu, and Jin					x				x								x				
2010	TCJ	Haklay et al.				x											x						x
2010	JLBS	van Schaick					x				x								x				
2011	CaGIS	Kessler					x										x		x				x
2013	CaGIS	Crampton et al.		x													x		x				
2013	CaGIS	Fujita		x													x	x					
2013	CaGIS	Ghosh and Guha		x							x			x		x							
2013	CaGIS	Kent and Capello	x	x							x		x			x						x	x
2013	JLBS	Krumm, Gruen, and Delling					x				x								x				
2013	CaGIS	Li, Goodchild, and Xu	x	x							x			x									
2013	JLBS	Mooney, Rehrl, and Hochmair				x					x												x
2013	JLBS	Patel					x												x				
2013	JLBS	Pippig, Burghardt, and Prechtel	x		x	x								x						x			
2013	CaGIS	Stefanidis et al.		x							x		x	x									
2013	CaGIS	Tsou et al.		x										x									
2013	CaGIS	Xu, Wong, and Yang		x							x		x										x
2014	CaGIS	Camponovo and Freundschuh		x									x										x
2014	JLBS	Gröchenig, Brunauer, and Rehrl				x											x						x
2014	CaGIS	Hawelka et al.		x												x							
2014	CaGIS	Sagl, Delmelle, and Delmelle							x								x	x			x	x	x
2014	JLBS	You and Krumm		x															x				
2015	JLBS	Gonçalves, Afonso, and Martins					x												x		x		
2015	CaGIS	Malleson and Andresen		x										x	x								
2015	JLBS	Tessem et al.			x						x												
2015	JLBS	Zhao					x							x					x				
2016	CaGIS	Gong et al.						x						x					x				
2016	TCJ	Hauthal and Burghardt	x											x			x					x	x
2016	JLBS	Oliveira et al.		x								x											x
2016	JLBS	Qiu and Wang						x			x												
2016	CaGIS	Robertson and Feick	x								x			x		x						x	x
2016	CaGIS	Shook and Turner		x							x			x									
2016	CaGIS	Zhou and Zhang		x								x					x						
2017	CaGIS	Baker et al.					x												x			x	
2017	CaGIS	John et al.				x	x				x						x		x	x		x	x
2018	JLBS	Almaslukh, Magdy, and Rey		x												x						x	
2018	CaGIS	Kounadi et al.		x							x						x						
2018	JLBS	Mamei and Colonna							x					x	x								
2018	CaGIS	Resch, Usländer, and Havas		x										x			x						
2018	JLBS	Rybarczyk et al.		x							x			x									
2019	JLBS	Çay, Nagel, and Yantaç										x											
2019	CaGIS	Jiang, Li, and Ye		x												x	x					x	
2019	JLBS	Keler and Krisp						x									x						
2019	CaGIS	Koylu et al.		x														x					
2019	JLBS	Viel et al.							x		x												
2019	CaGIS	Wang and Ye		x										x		x							
2020	CaGIS	Gröbe and Burghardt		x							x		x					x					x
2020	CaGIS	Hu, Li, and Ye		x							x					x							
2020	JLBS	Kveladze and Agerholm								x	x												
		∑	6	24	2	5	9	3	4	2	20	3	5	17	2	9	13	4	13	3	2	8	13
		%	12	47	4	10	18	6	8	4	39	6	10	33	4	18	25	8	25	6	4	16	25

2.4.3 Challenges mapping crowdsourced data

Overall, the new user-generated data is uncertain due to user errors and GPS device inaccuracies that are often incorporated into data creation or bounding boxes of abstracted places, e.g., Twitter. The long tail effect is typical (Elwood, Goodchild, and Sui 2013). Few users contribute considerable data, while many users contribute far less than the power users. This phenomenon also happens spatially; there is a great deal of data in urban areas, while rural areas contain fewer data.

To deal with this circumstance, new visualization techniques are being used more frequently, such as point maps, heat maps, micro diagrams, and space-time cubes. More traditional approaches are usually designed to work with fewer but higher-quality data.

The new crowdsourced geodata properties are similar to the properties of big data: variety, veracity, volume, and velocity (Schroeck et al. 2012; Tsou 2015). The new data, with challenging properties for cartographers, create the need for new or adjusted methods and tools for analysis and visualization (Burghardt, Duchêne, and Mackaness 2014, 128; Tsou 2015).

2.5 Technologies for serving multiscale maps

The next section explains the foundation of today's multiscale maps, technology, historical development, and research in the field. It provides a broad impression of how a community can develop and adopt the technology and standards. Innovations were driven mainly by big companies, named in the section for specific technologies. For this reason, this portion of the theoretical basis is more technical than scientific. The advantages and disadvantages of the described methods and their applications are detailed to better understand their use. The focus is on the technological evolution of multiscale web maps for a wide audience, not on the general term of web mapping, its platforms, applications, eras, or history (Tsou 2009; Veenendaal, Brovelli, and Li 2017).

2.5.1 Research about multiscale maps

Cartographers should be familiar with multiscale mapping. Its series of scales is associated with topographic mapping and is established in most countries worldwide. The current development was driven by big tech companies, as shown in Figure 1. Google introduced free multiscale web maps, and Mapbox initiated the current tectological trend toward vector tiles. Cartographers identified the need for research and to apply already developed concepts in this field (Jones and Mark Ware 2005). In parallel, the open source community developed solutions for OpenStreetMap (Ramm and Topf 2011).

A decision tool called ScaleMaster can guide multiscale map designers through a range of data resolutions, display scales, and map purposes. The resulting conceptional framework can be demonstrated in case studies and later extended with a set of operators to maintain legibility (Brewer and Buttenfield 2007; 2010; Roth, Brewer, and Stryker 2011). A ScaleMaster is a diagram that describes the level of detail and the visualization of features depending on the scale. It was applied to multiscale hypsometric mapping and a prototype system for designing multiscale maps (Samsonov 2011; Samsonov, Podolsky, and Yurova 2013). Somewhat similar is a multiscale legend, which uses a matrix concept to symbolize features depending on the scale. Possible use cases are developing and evaluating multiscale maps (Gröbe 2019).

Another research topic was correlating content or map load with zoom levels and whether additional zoom levels between individual scales would be helpful. It led to further implications for the generalization process. (Dumont, Touya, and Duchêne 2015; Dumont, Touya, and Duchene 2016; Dumont, Touya, and Duchêne 2016)

2.5.2 Web Mercator projection

The map projection of today's web and multiscale maps is called Web or Pseudo Mercator projection (Battersby et al. 2014; Wikipedia 2021b; OpenStreetMap Wiki 2022a), a variant of the well-known Mercator projection.

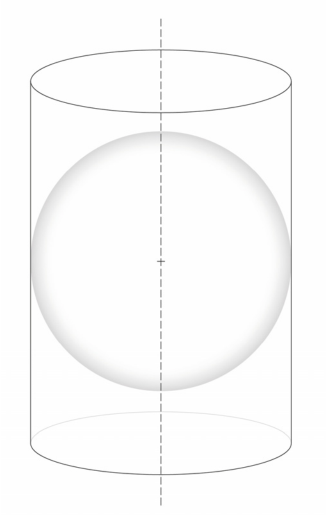


Figure 8: Principal of the Mercator projection in its normal position. It is similar to a cylindrical map projection, depicting the world on a plane, but is not a perspective.

In 1569, Gerhard Mercator presented a conformal projection, similar to projecting on a cylinder but not a perspective, leaving the meridians as equally spaced straight lines. The parallels are unequally spaced straight lines, closest to the Equator, cutting the meridians at right angles. Given that the Poles are at infinity, this leaves a significant distortion of the polar regions. Only along the Equator or parallels equidistant from the Equator are true to scale. The projection always preserves local directions and shapes, representing north as up and south as down everywhere in its standard position when the cylinder axis is identical to the Earth's rotation axis, as Figure 8 demonstrates. Loxodromes are straight lines on a map, making the projection handy for navigation.

Figure 9: First and only tile from OpenStreetMap on zoom level 0, showing the whole world on a square tile of 256 pixels. (Image: © OpenStreetMap contributors)

The properties of the Pseudo Mercator are nearly the same, but there is a difference: The following PROJ code shows that the projection does not use an ellipsoidal model for the Earth. Instead, the Earth is a sphere defined by the parameters "a" equal "b," which usually describes the semi-major and minor axis. For WGS84, "b" should be 6,356,752, but the value for "a" is still correct. This results in distortion, meaning a difference of up to 50 kilometers in the northing values in polar regions and that the projection is no longer conformal. (Battersby et al. 2014).

```
+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0
+x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext
+no_defs
```

After Google acquired the technology company *Where 2 Technolgies* in 2004 from Lars and Jens Rasmussen, Lars went on to co-found Google Maps, which developed the new Pseudo Mercator projection; through its usage in Google Maps, the new projection reached a broad audience (Battersby et al. 2014). Later, the application was reverse-engineered (cfis 2006), which was simple to do because the JavaScript code was publicly available, implemented in OpenSource software projects, and started spreading.

In the following years, various organizations started to introduce standardization. An old and unofficial reference was the EPGS code 900913, a transliteration of Google to the numbers: 9-G, 0-o, 0-o, 9-g, 1-l, and 3-e (Schmidt 2007). Other codes from OSGeo and ESRI were also used to describe the new map projection. In 2008, the EPSG was standardized, first as EPSG code 3785 (Popular Visualisation CRS/Mercator), then later adjusted to the final number and current code EPSG:3857 (WGS 84/Pseudo Mercator) (Wikipedia 2021b).

In 2018, Google changed its projection again to show the Earth as a globe when zooming out, using an orthographic map projection (heise online 2018). Since the projection looks like a globe, it eliminates the adverse distortions of the Web Mercator projection.

From a historical viewpoint, Mercator never intended to use this projection for a world map (Snyder 1987); its objective purpose was for navigation. For this use case, it does not matter that the map is not true to the area, even with the oversized pole regions and underestimated equatorial regions. Further, simplifying the use of a sphere is no more than a conformal projection (Battersby et al. 2014). Measuring distance is only minimally possible. Nevertheless, those things should not be done with a web map because digital tools can do this correctly and accurately. Using a globe would always be the better option, but this is technically complicated.

Mercator projection is quite a good solution if the projection is only used for web mapping, its original purpose. It always preserves shapes, is relatively simple to implement and calculate, and is not restricted to a specific scale (Battersby et al. 2014; cfis 2006). The fact that it is impossible to visualize the pole regions does not matter since those areas are nearly uninhabited, with fewer people interested in visiting.

2.5.3 Tiles and zoom levels

The new projection is not the only characteristic of multiscale maps; so is substituting the principal scale with zoom levels and splitting images into small images called "tiles." While the tiling is a known methodology for organizing raster data, the zoom levels are new, similar to the scale and connected to Web Mercator projection.

The worldwide map starts with a single tile in Figure 9. The extent of the Web Mercator and the scale is adjusted to fit on this first tile of 256 pixels (cfis 2006), resulting in an approximate scale of 1:500,000,000 ("Zoom Levels – OpenStreetMap Wiki" 2019). Often users do not recognize the first tile because it is too tiny for regular screens and is not shown. Google Maps started with eighteen zoom levels; the lowest zoom level showing the whole world is zero.

OSGeo

OSGeo is the abbreviation for the Open Source Geospatial Foundation. A non-profit, non-governmental organization founded in 2006 that supports and promotes the collaborative development of open geospatial technologies and data. It is a community-driven organization that organizes the annual FOSS4G conference. Well-known projects include QGIS, GDAL/OGR, and PostGIS.

EPSG codes

EPSG codes are numbered entries in a public registry of geodetic datums, spatial reference systems, Earth ellipsoids, coordinate transformations, and related measurement units that address a specific entry. It straightforwardly identifies all spatial reference systems. Initially, the data set was created by the European Petroleum Survey Group (EPSG) and was made public in 1993.

PROJ

PROJ, formerly PROJ4, is a software library for coordinate transformation. It transforms geospatial coordinates from one coordinate reference system to other projections. It supports many cartographic projections and geodetic transformations, and the software is an essential part of all open source GIS software.

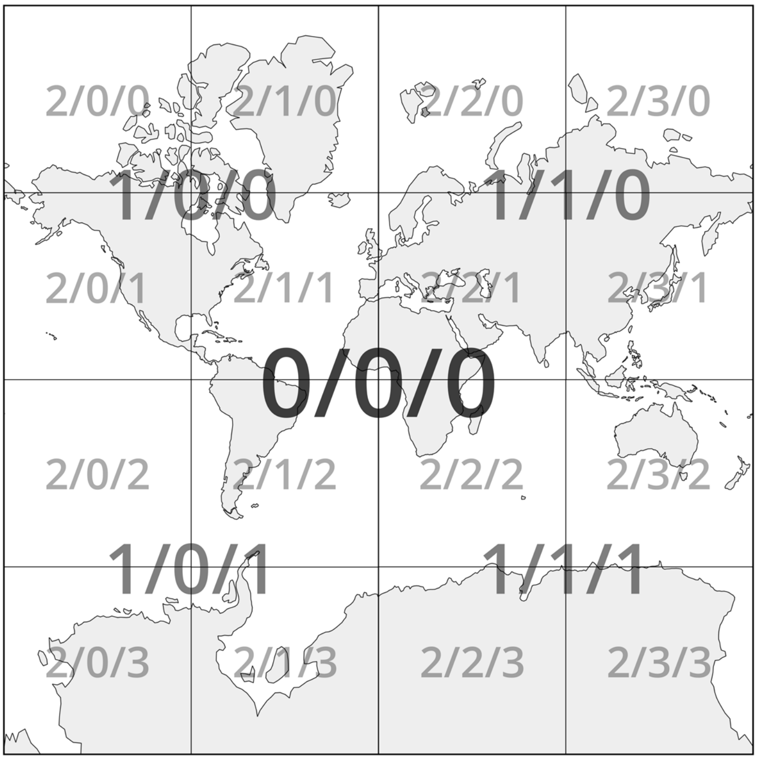


Figure 10: Visualization of the first tile (0/0/0) using Pseudo Mercator projection, showing further slicing for zoom levels 0, 1, and 2. The tile numbering follows the OpenStreetMap schema (zoom level/column/row or short z/x/y) with the origin in the upper left corner. Please note that the southern border is not shown accurately for correct tiling.

In zoom level one, similar to a quadtree (Samet 1984), the world is shown in four tiles, with each tile 256 pixels. This pattern continues until the last and most detailed zoom level 17 (cfis 2006). Figure 10 visualizes the approach for the first three levels, using the Google Maps and OpenStreetMap schemata, with the origin in the northwest corner. The tiles are named after the zoom level, column, and row. One convention is to use the variables z, x, and y as short names for zoom level, column, and row ("Slippy Map Tilenames – OpenStreetMap Wiki" n.d.). Bing Maps uses a slightly different one-dimensional approach, numbering the tiles starting in the upper left corner with a "quadkey" (Schwartz 2022).

Table A9 provides an overview of the zoom levels and their approximate scale at the Equator. In addition, a reference measure of one pixel to meters is given, but again, this is only valid for the Equator. This behavior is characteristic of the Web Mercator projection, with the scale changing when going north or south. It makes sense not to use the term "scale" because a numerical scale is not applicable.

Splitting the map into fixed small images offers the possibility of pre-rendering the tiles and merging them visually with the client's viewer. This approach makes it easier for many users to host a map with less computation because there is no need to render a map for each user (cfis 2006).

2.5.4 Raster tiles

The following ordering of standards for accessing map data through an API follows the technical development shown in Figure 1. It is always no geodata. Instead, all those interfaces deliver raster images live or pre-rendered from various data sources with no georeferences included. Nevertheless, using the APIs described here with other file formats would be possible, but it is not common.

Raster tiles

Raster tiles are visualized geodata cut into tiles in a standard raster image format.

2.5.4.1 WMS

The first version of web mapping service or server, part of OpenGIS web services (OWS), was standardized by the OGC in 2000. It provides an HTTP interface for a server when a client requests images with specified properties (OGC 2000). It is a part of a web service for geodata with a similar structure, allowing geodata to be handled in different ways (Andrae et al. 2011; OGC 2014).

OGC

The Open Geospatial Consortium (OGC) is a standards organization that was founded in 1994. It aims to develop and implement open standards for geospatial content and services, such as WMS, WMTS, and the GeoPackage.

A WMS must be able to handle at least the first two of the following requests to be meaningfully used by a client. Version 1.3.0 defines the support of GetCapabilites, GetMap, and GeoFeatureInfo operations as mandatory. The order stems from the logic that it is impossible to formulate any other request like the GetCapabilites request without knowing the GetCapabilites response. The most used is the GetMap request, answered with an image of rendered data. The client always sends a request via an URL (OGC 2006).

GetCapabilities: The capabilities request is an XML file containing information about metadata, available layers, supported projections, and image formats.

GetMap: The client requests a map as an image via URL values. The response contains a map extent in a specific format and in an offered coordinate system; what is available can be obtained from GetCapabilities. The server then sends—if possible—the map section in the requested format to the client, which displays the result.

GetLegendGraphic: A legend for the requested layer and its data is sent to the client.

GetStyles: A list of available styles is sent.

GetFeatureInfo: The server can send geodata with attributes for a specific map section to the client on request.

The benefit of WMS is flexibility: the adjustment of resolution, scale, style, and extent is possible. In addition, the service supports metadata and time—a map can be rendered at different requested points in time. All of those features are possible through complex requests, which are possible through operations and documents. Nevertheless, all of those steps need computing time, and different requests result in rendering the same data repeatedly, which takes time. Technically, the setup needs the following components: an HTTP server for managing communication and GIS software to handle the essential request and render the data as raster images.

2.5.4.2 TMS or XYZ

Both services behind the abbreviations of TMS and XYZ are not standardized. However, conventions exist for how to deal with these APIs, which are the basis for many web maps, following the convention of Google Maps (Wikipedia 2021a). It is a kind of representational state transfer (REST) API, which is easy to use.

XYZ stands for the row, column, and zoom level of a map tile, as in Figure 10, usually using a PNG image or another image format, with the origin in the upper left corner. XYZ completely follows the implementation of Google, and is bound to the Web Mercator projection and to the predefined zoom levels. It is often used, but it is not adjustable beyond changing the image format (OpenStreetMap Wiki 2021a; Wikipedia 2021a). The approach is part of a patent submitted in 2004 titled "Digital Mapping System" (Rasmussen et al. 2004).

TMS is very similar to XYZ—compared with the XYZ API, the TMS Y value is flipped. The origin of the tile numbering is in the lower left corner. As a cheat, some software supports a minus bevor Y-variable to deal with this circumstance (OpenStreetMap Wiki 2019). The service has been de facto standardized with OSGeo since 2006 (OSGeo 2012a) and offers some variables. It supports different projections, zoom levels, and tile sizes, and it also offers a file with available resources called "tilemapservice.xml." The two approaches work with caching and do not expect a live rendering of tiles. A simple script is enough to generate the needed folder structure to set up the service (OpenStreetMap Wiki 2021b).

The cache and its simplicity are advantages of TMS and XYZ. While a TMS is flexible at some point, XYZ is just as it is. It was designed for and applied to highly used maps that are not subject to change. Implementing a TMS or XYZ service only needs an HTTP server.

2.5.4.3 WMTS

In 2010, a WMTS had a similar architecture as a WMS, but only needed an HTTP server to fulfill the minimum requirements. Inside the standard, it's clear that its development was motivated by the success of Google Maps, Yahoo! Maps, and others (OGC 2010). WMTS mixes the features of WMS and TMS and is similar to a client, not officially standardized TMS or XYZ. It is the result of development started by extending the WMS, with an additional profile for tiled caching called WMS tile caching or WMS-C (OSGeo 2012b).

A WMTS offers the possibilities of GetCapabilities and GetFeatureInfo like a WMS, but not the GetMap operation. Instead, there is a GetTile request that delivers the rendered geodata. Service metadata must be available, and the origin of the row and columns is again in the upper left. It is possible to use different profiles to adjust to varying needs and purposes (OGC 2010):

- WMTS answers requests by using lists of parameters and their values, defined as lists of key-value pairs similar to the WMS.

- WMTS can support the SOAP encoding for each operation by offering the needed metadata for the operation.
- WMTS acts like a TMS or XYZ server that supports only RESTful access for each resource, with no further operations.

The advantage of the WMTS is the optional GetFeatureInfo option, combined with the flexibility of defining its own tile sets for each projection. Furthermore, it is a standard supported by different software applications. Hosting can be done with a simple HTTP server or an additional GIS component in order to offer the data behind the features.

2.5.5 Vector tiles

The following section explains the use of vector-based formats in multiscale mapping. To access the data, the same APIs are used for raster tiles. There are more differences than just the file format, but the structure of this section focuses on this point because it was essential for development. There was a wide range of formats in use, starting with vector graphic formats, geodata formats for tiles, and a specialized format for Mapbox vector tiles in 2014 (Mapbox [2014] 2021). The development progression culminates in the OGC API tiles (OGC [2019] 2021), which will be finalized soon.

Vector tiles

Vector tiles are not vector graphic, which is cut into tiles. They are a visualization-optimized geodata format with limited geometry precision and attributes.

2.5.5.1 SVG and other graphic formats

Scalable vector graphics (SVG) is an XML-based vector image format. It has been standardized since 2001 (W3C 2001) and is an open standard developed by the World Wide Web Consortium (W3C) in 1999.

SVG is technical, a possible output format of a WMS (OGC 2006). There is some evidence of trying to use it with the CloudMade vector stream server (GDAL 2022a), but it seems to not be used often. Reasons for this are a long rendering time for the client compared to raster images or the possibility of unauthorized usage. Other technological approaches, such as Flash, Silverlight, or JavaFX, were explored and compared (Lienert et al. 2012). No comprehensive application was achieved.

2.5.5.2 GeoJSON and TopoJSON

Another possibility is using geo-vector data formats on the web. GeoJSON (Butler et al. 2016) and TopoJSON (Bostock and Calvin Metcalf [2012] 2013) are essential formats that represent the attributes of geographic features based on the JSON format. The JavaScript Object Notation (JSON) is an open standard file format that is human-readable and derived from JavaScript. While GeoJSON represents each geometry discreetly in TopoJSON, line segments are stored only once to reduce the file size.

Simply overlaying single objects was (Lienert et al. 2012), and still is, possible in web mapping. Combining tiling technology and the JSON formats was the first approach toward vector tiling (Gaffuri 2012; Antoniou, Morley, and Haklay 2009). Many web mapping libraries, such as Leaflet[5] or OpenLayers,[6] can handle and visualize it according to its attributes.

The drawbacks of this method are slow rendering (Lienert et al. 2012; Farkas 2020) and relatively big file sizes. TopoJSON attempted to reduce the file size by zipping the files for transmission over HTTP. Nonetheless, the JSON formats never reached a small file size comparable with the data density of raster files. A particular challenge for all of those formats is the merging of cut lines and polygons during the tiling process; merging is necessary for rendering (Nordan 2012; Lin Li et al. 2017).

2.5.5.3 Mapbox Vector Tiles

The most used standard in vector-based web mapping for end users is Mapbox vector tiles (MVT). The format's first specification was published in 2014, describing the encoding of geometries and their attributes in the Google protocol buffers as the container format. To reduce tile size, screen coordinates inside the tiles are used, and the attributes are stored in a dictionary (Mapbox [2014] 2021).

Mapbox

Mapbox is an American provider of customizable online maps for various applications. Founded in 2010, the company offers a comparable service to Google Maps, and is a creator or contributor to open source mapping libraries and applications.

It would make sense if Google Maps started using a similar approach again, but there is no news of such a shift. Interestingly, specifications were created by Mapbox, but Apple holds a patent for a very similar technique (Pahwa, Lenoski, and Courtemanche 2015), in addition to some patents later obtained from Mapbox (Saman Bemel-Benrud et al. 2017; Thompson and Clark 2018).

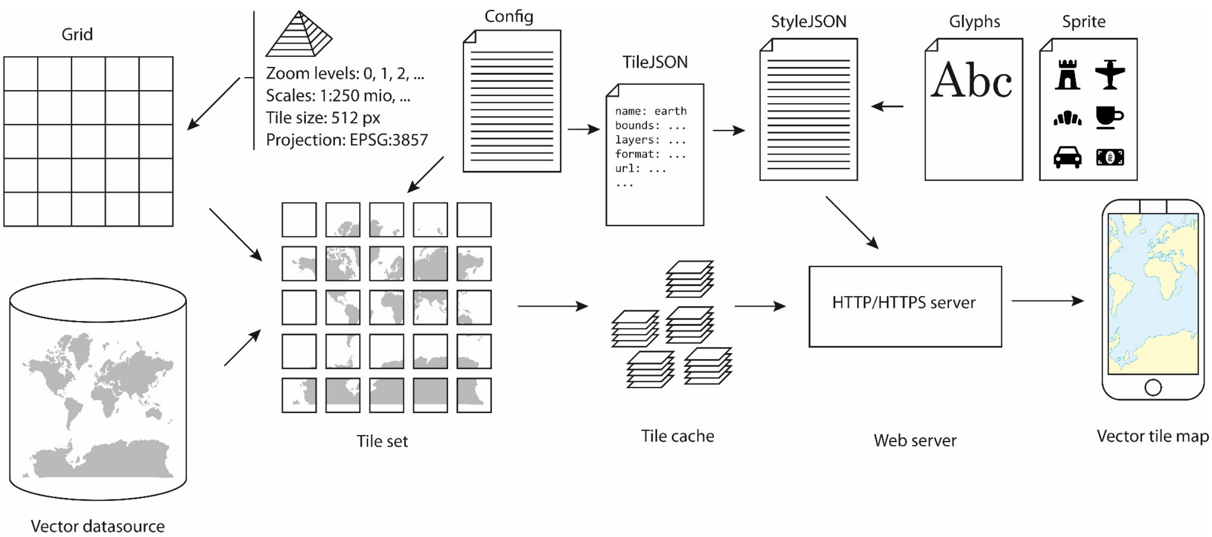


Figure 11: Overview of how Mapbox vector tiles are created and delivered to the end user, including the necessary fonts and symbols for rendering the map for the client in specialized formats.

[5] https://leafletjs.com/

[6] https://openlayers.org/

Besides using a new, specialized file format with strongly compressed and simplified geodata, many technical improvements made Mapbox vector tiles production-ready. The tiles contain no geocoordinates, only screen coordinates, but they do have attributes. The hardware acceleration for rendering using WebGL is another crucial feature. Figure 11 shows an example workflow for creating the data visualization needed for Mapbox vector tiles. Similar to raster tiles, the map is delivered in tiles, but now in 512 pixels. From these results, the vector tile zoom levels differ by one number because the first tile on zoom level zero has double the size. Table A9 makes this behavior clearer.

A file called TileJSON (Anon, 2022a) describes the properties of a collection of tiles and their attributes, and a StyleJSON ("Sprite | Style Specification | Mapbox GL JS" n.d.) brings the tiles and map style together. For optimized rendering, the font is stored in a particular format called glyphs ("Glyphs | Style Specification | Mapbox GL JS" n.d.), and all the icons are stored in an image called a sprite, an additional JSON file ("Sprite | Style Specification | Mapbox GL JS" n.d.).

As a result, it is now possible to consistently deliver the same tiles and adjust the style as applied by the client. It allows the client to change the map style, rotate the fonts if the map is rotated, and adjust feature names to the client's preferred language. Another point is seamless zooming; it is possible to interpolate the data between zoom levels.

2.5.5.4 OGC API tile

Currently, the OGC API tile standard is under development. The API should provide tile geospatial information. This is offered in different forms, such as map tiles, coverage tiles, styled vector tiles, styled map tiles, and the style itself by combing the principle of tiles with other OGC standards (OGC [2019] 2021; Cavazzi 2018). This would replace the currently known approaches of WFS, WCS, and WMS with a RESTful API that always provides the data as tiles.

2.5.6 Tiling as a principle

Besides the approaches mentioned here, raster datasets are often stored in tiles to allow effective access to smaller pieces of big data sets and to enable less complicated handling. In addition, processing geodata in a tiled way offers advantages for the generalization process (Touya et al. 2017). Further research could explore the possibilities of parallel data processing and handling the reemergence of cut-up features. Formats and technologies have changed over time, but the principle has stayed the same. Data is cut into rectangles, whether it is vector or raster data, for transmission, visualization, and processing.

3 Point collection visualization with categorized attributes

Point collection sources can be varied. One source is volunteered geographic platforms with a defined data structure and metadata for sharing knowledge, such as OpenStreetMap or eBird.[7] Those mapping platforms are specialized databases for spatial knowledge, while Wikipedia and Wikidata are more generic collections of data with text, pictures, and many more types of stored information. All of this data can also be seen as POI because a bird observation on eBird can be registered with a number to track that bird, and detailed information about a house on Wikipedia can distinguish it as very special. So, it is a question of scale and the aim of the analysis whether a point is part of a point collection or a POI. Possible example categories from these sources include bird species from eBird, a type of restaurant or cuisine from OpenStreetMap, or different language versions of Wikipedia entries.

For LBSM, the overall properties are more typical for point collection (Bereuter 2015): The user creates content to share with others; it is likely for photo sharing platforms, e.g., Flickr, for social networks, e.g., Twitter and Facebook, or for recommendation platforms, e.g., Foursquare or TripAdvisor. There are no clear boundaries between these service providers—it is possible to share photos on Facebook or Twitter and have a discussion on Flickr. The value of this information becomes relevant for a general audience in terms of volume, not single contributions. Possible categories for applying this data are the language or time of day of a tweet or the mobile operating system or camera model used to upload a photo to Flickr.

Another source could be statistical data from a census or a research institution. Those data sources can be very reliable but hard to obtain due to underlying privacy concerns stemming from point locations. Practical examples of census data can be found in the German Federal Statistical Office (*Statistische Bundesamt*) or the Robert Koch Institute for public health. Possible categories could include the religious denomination, the country of origin of non-nationals, and disease variants.

3.1 Target users and possible tasks

The most likely application would be analyzing the point data. Advanced users or visualization experts, such as data scientists, cartographers, and analysts, should be able to interpret the point patterns and be interested in going into more detail. With the help of various visualization techniques and their parameters, it should be possible to explore the data in detail and adjust the visualization to meet specific needs.

A possible first step for users could be checking the spatial extent and data distribution. Next, they might determine the data amount and compare

[7] https://ebird.org/home

category distribution. Third, reading specific values containing information about the exact numerical proportion of categories to reveal hotspots could be the next step. Eventually, the analyst could explore combinations of categories.

3.2 Example data

Twitter data from January 1, 2019, will be used. Each tweet is single point geometry and contains information about the tweet language provided by Twitter. This dataset will be utilized to show possible visualizations of a point collection with categorized attributes. It will demonstrate the benefits and shortcomings of different approaches to point collection visualization and discussion bases.

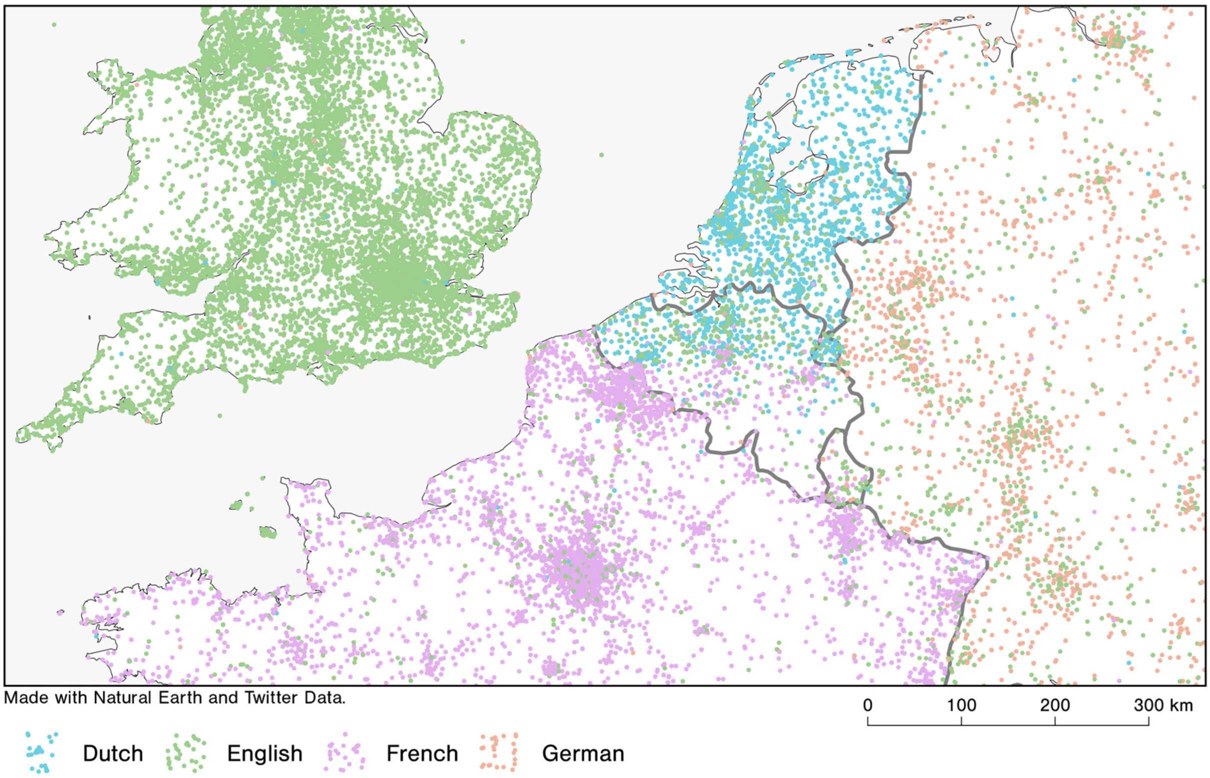


Figure 12: Point visualization of English, French, Dutch, and German tweets for January 1, 2019.

Belgium and the Netherlands is an exciting areas for high numbers of tweets due to the clear French/Dutch language border inside Belgium, as well as the clear language boundary along national borders. PostgreSQL's spatial PostGIS and QGIS for visualization were used for data storage and map computations. To compare, Figure 12 uses the standard point method for visualizing the language distribution, while Figure 15 shows the same data utilizing micro diagrams. For better map interpretation, the languages are limited to the four that are visible within the countries: Dutch, English, French, and German.

3.3 Visualization approaches

The following section explains different visualization techniques for point collection and categorical attributes, from straightforward to more sophisticated methods for providing an overview of the possibilities.

3.3.1 Common techniques

There are standard techniques in use for point data. For each category, an example map gives an impression of how a result can look, referencing cartographic textbooks and literature.

3.3.1.1 Point maps

Currently, visualization as points colored according to attributes of interest is the most used approach, as in the tweet language example. Many examples use data from Twitter, Flickr, or OpenStreetMap (Fischer 2011; Linna Li, Goodchild, and Xu 2013; Leetaru et al. 2013; Hauthal, Dunkel, and Burghardt 2021; Gugulica, Hauthal, and Burghardt 2020; Dunkel 2015; Ferreira et al. 2013), which requires no calculation and the data is shown as is by plotting simply all points.

There are different cartographic approaches for visualizing points according to the literature (Kraak and Ormeling 2010); this can include point and dot maps. Dot mapping means that “one dot is set equal to a certain phenomenon, and dots are placed where that phenomenon is most likely to occur” (Slocum et al. 2009, 90). As a result, several points can be combined and symbolized by varying the dot sizes. The nominal point symbol map (Kraak and Ormeling 2010, 137, 144) contains “symbols that are different in shape, orientation or color.” In this case, each point stands for one object and is not aggregated. An occurrence can be shown through geometric and figurative symbols depending on the data and use case. For point collection, geometric shapes are better because they are more generic and need less space on the map.

The dot map and the nominal point symbol map do not appropriately describe the approach used by Fischer and others because there is no dot aggregation, and the amount of data is so large that the points are not distinguishable, similar to a nominal point map. Due to this circumstance, the visualization would be better called a point map, if a simple geometric shape represents one occurrence of a phenomenon’s true position and uses colors to distinguish categories. This technique works well for one category shown in one color. The resulting visualization can be improved by alpha blending techniques (Ellis and Dix 2007).

A disadvantage is that in regions with high data density, many points overlap and cause clutter, while others look empty. Using the visual color variable (Bertin 1974) to show categories reinforces the cluttering effect and makes the order of point drawing essential, so that there is no systematic overlap. In addition, it is hard to quantify the visualized phenomenon; sometimes the points are countable but in dense areas, only is only a rough estimation possible.

3.3.1.2 Point map example

On account of the importance of this approach, Figure 12 demonstrates the point map method. The tweet points on the map are plotted in the order drawn to ensure that the languages and their colors do not systematically overlap each other. Color selection was applied using hclwizard[8] to ensure that the colors have the same brightness, avoiding any bias from one color attracting more attention than another.

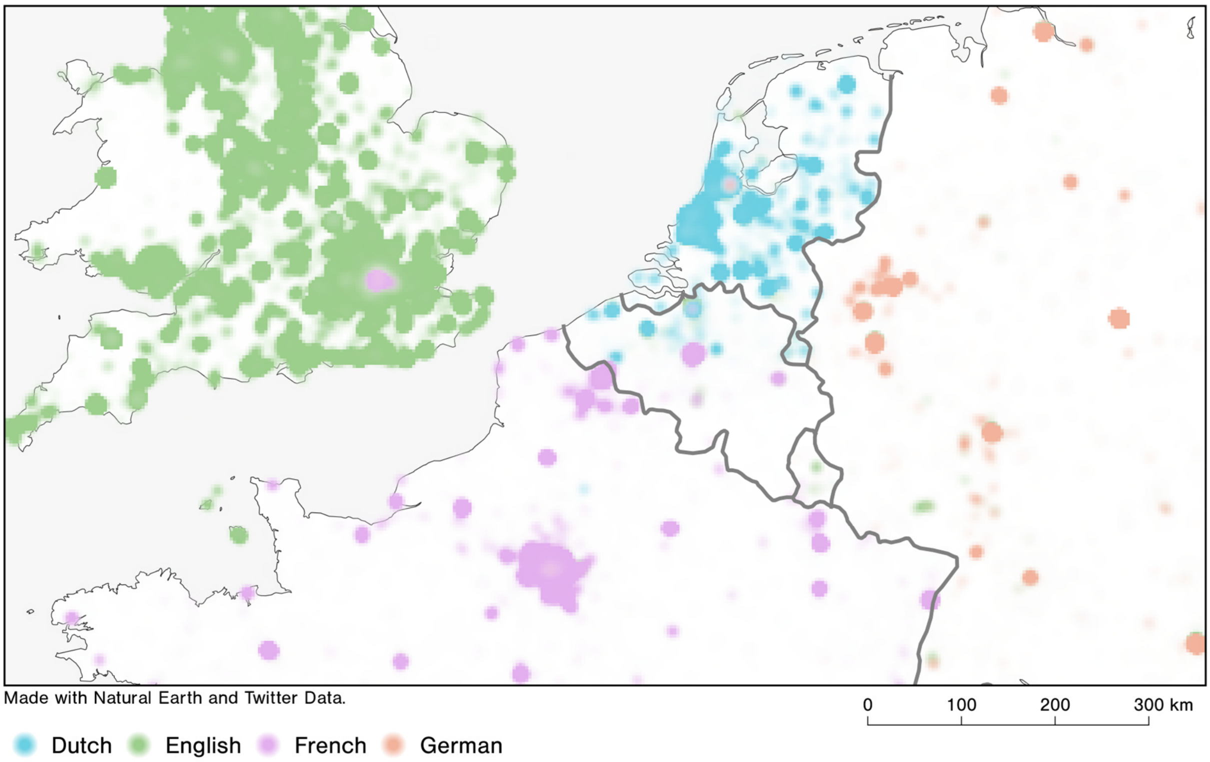


Figure 13: Kernel density estimation visualization of English, French, Dutch, and German tweets for January 1, 2019. Other layers overlie some areas. The drawing order is English, Dutch, French, and German.

3.3.1.3 Interpolated surfaces

The kernel density estimation (KDE) is a statistical tool for smoothing and interpolating values. A chosen kernel function interpolates point data into a defined raster containing specific estimated values. Usually, the result is visualized as a color gradient. This qualitative visualization can deal with the fuzziness of high-volume data sets and avoid cluttering effects. The result is mainly limited to showing larger or smaller data availability in a region because reading values is difficult for the map user. Another use case could compare two selected categories (Linna Li, Goodchild, and Xu 2013), and KDE is commonly applied (Hollenstein and Purves 2010; Tsou et al. 2013; Ghosh and Guha 2013; Huck, Whyatt, and Coulton 2015; Resch, Usländer, and Havas 2018).

[8] http://hclwizard.org:3000/hclwizard/

Figure 13 shows one weakness, which arises when using more than two categories: The different raster cells, with their interpolated values, are overlayed. It is possible to deal with this limitation for two categories by showing the dominating one, but not more. As a result, some values are hidden in the example map, e.g., in the areas of London and Brussels, where other languages are expected. Especially when compared to Figure 12, the bias of this approach becomes apparent.

3.3.1.4 Area-related approaches

Summarizing area values is a common technique in cartography and statistics. Administrative units are often used, as well as grid-based independent units. More recent approaches utilize the geohash data structure or hexagon-shaped areas. The results can be shown in diagrams of the area or by coloring areas to highlight the dominant category.

An example of using geohash data structure for aggregation is the Kibana[9] visual analytics tool. Administrative areas are commonly used if the analysis is somehow related to it, such as using, for example, census data and comparing the analysis with election results or with regions (Linna Li, Goodchild, and Xu 2013; Graham, Hale, and Gaffney 2014; Hauthal, Dunkel, and Burghardt 2021; Hauthal, Burghardt, and Dunkel 2019). Grid approaches are more independent and can show more details because the grid can be tailored to specific needs, avoiding random effects related to administrative borders (Wider, Palacio, and Purves 2013; Hollenstein and Purves 2010; MacEachren et al. 2011; Crampton et al. 2013).

This type of use case is shown in Figure 14. Displaying the dominant language by administrative area would only show a country's borders. Inside the countries, however, a grid can better reveal differences in Twitter language usage. This visualization approach displays areas in Germany where more English and Dutch tweets are created. In addition, the language border inside Belgium is more noticeable. An approach that uses independent reference units combined with well-known cartographic standards, such as minimal dimension, adds more information to the visualization compared to the previously discussed techniques, even though the initial dataset is still the same one. It underlines the impact visualization means can have on supported messages.

3.3.2 The micro diagram approach

3.3.2.1 The definition of micro diagram

Micro diagrams are a visualization method that can deal with a large amount of point data while simply resolving cluttering effects. It is a method for analyzing location-based social media data and according to Tsou, 2015, it may be a tool that can help manage the challenges of big data in cartography. The result of the visualization process is a well-organized map, similar to a cartogram but not distorted, such as a tile map (McNeill

[9] https://www.elastic.co/de/kibana/

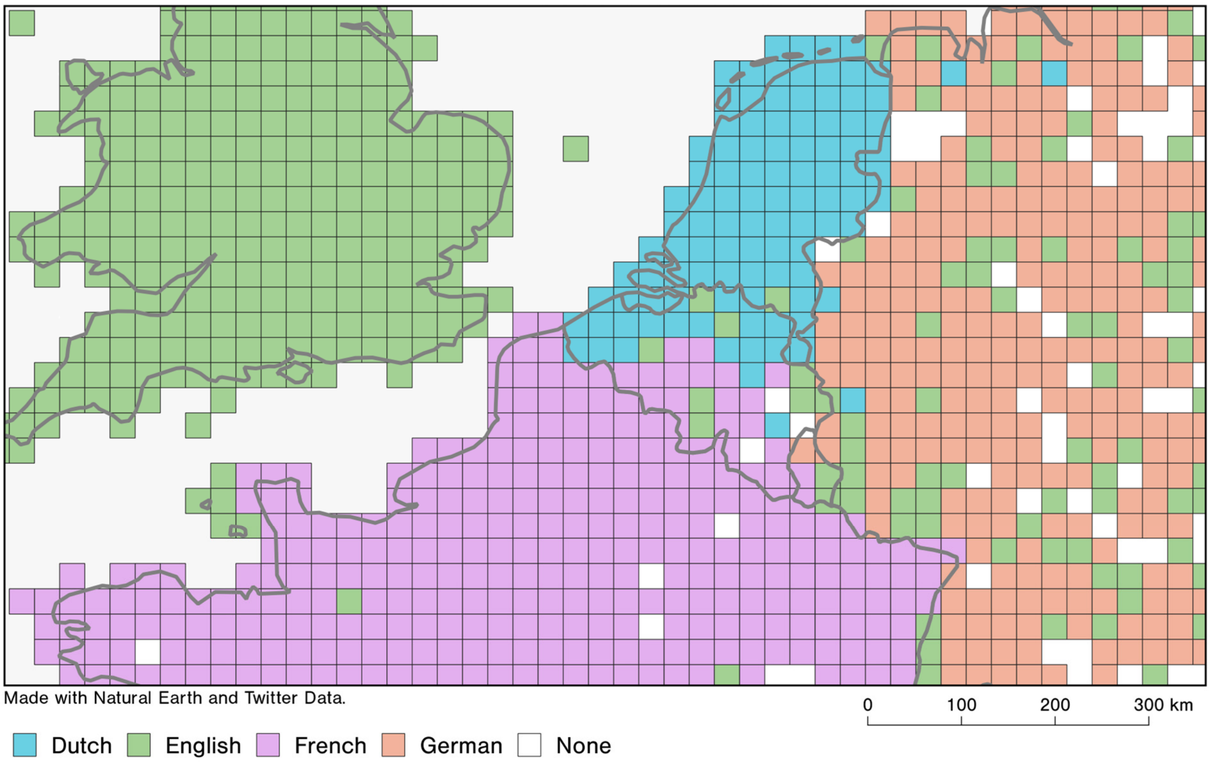


Figure 14: Grid visualization of English, French, Dutch, and German tweets for January 1, 2019. A grid cell shows the color of the language with the highest number of tweets or no color.

& Hale, 2017), and it can be combined with a reference map. Similar cartographic visualization approaches for lines (Bleisch & Hollenstein, 2017) and areas (Heimerl et al., 2018) are also known.

Like grid maps, micro diagrams are derived by aggregation within a regular grid. When visualizing area-based phenomena to create the look of a grid, a better-suited name would be micro grid diagrams. Small-sized diagrams, just above the minimum dimension, represent the proportion of the categorical attribute in a grid cell. In contrast, diagram size variation visualizes the overall number of values. The result is a detailed map with no cluttering effects, showing the exact categorical value distribution and not just the dominant category.

What distinguishes micro diagrams from conventional diagrams when combined with maps? So far, diagrams visualize the data related to a point, a point diagram map, or an administrative unit as an area diagram (Kraak and Ormeling 2010, 137). Combining a grid with small diagrams, or aggregating point data into small units to visualize multiple categories, makes this method suitable for point data. Using grids to visualize one variable with proportional symbols is known as a proportional symbol grid map (Kraak and Ormeling 2010, 137), a grid net map (Bollmann 2001, sec. Gitternetzkarte), or "symbols of variable size in a grid" (Bertin 1974, 137).

3.3.2.2 Micro diagram map example

Similar to the example in Figure 14, a grid is the basis of the micro diagram visualization in Figure 15. The points inside each unit were summarized and organized into five classes based on the minimum value of one tweet and the maximum value of 24,643 tweets in one grid cell. Due to the long tail effect, manual classification was applied. The size of the symbols corresponds with the grid. The areas of the smallest circle, with a 0.6 millimeter diameter, and the largest circle, with a 3.3 millimeter diameter, are derived this way. For the example map, a cell size of 40 kilometers in the Web Mercator fulfills this requirement. For maximizing distinguishability, five classes show the amount of data. The highest class fills a grid cell with its diameter and leaves some space at the edges due to the shape of the pie chart.

Indifference to the area-based visualizations of Figure 13 and Figure 14 is indicated by the visibility of more than just the dominant language. The pie charts show the language distribution in each grid cell. The point method in Figure 7 provides similar information but here, the information is quantitative and cluttered. Numerical proportions are much clearer, and the number of points is distinguishable.

Again, the language border in Belgium is noticeable; however, many more details stand out in other countries. English tweets dominate the United Kingdom, but in all other countries, English only comprises a fraction of tweets, with particular national languages dominating. The amount of data differs between high density in the Netherlands and Great Britain and less density in the other countries, which is also visible on the point map.

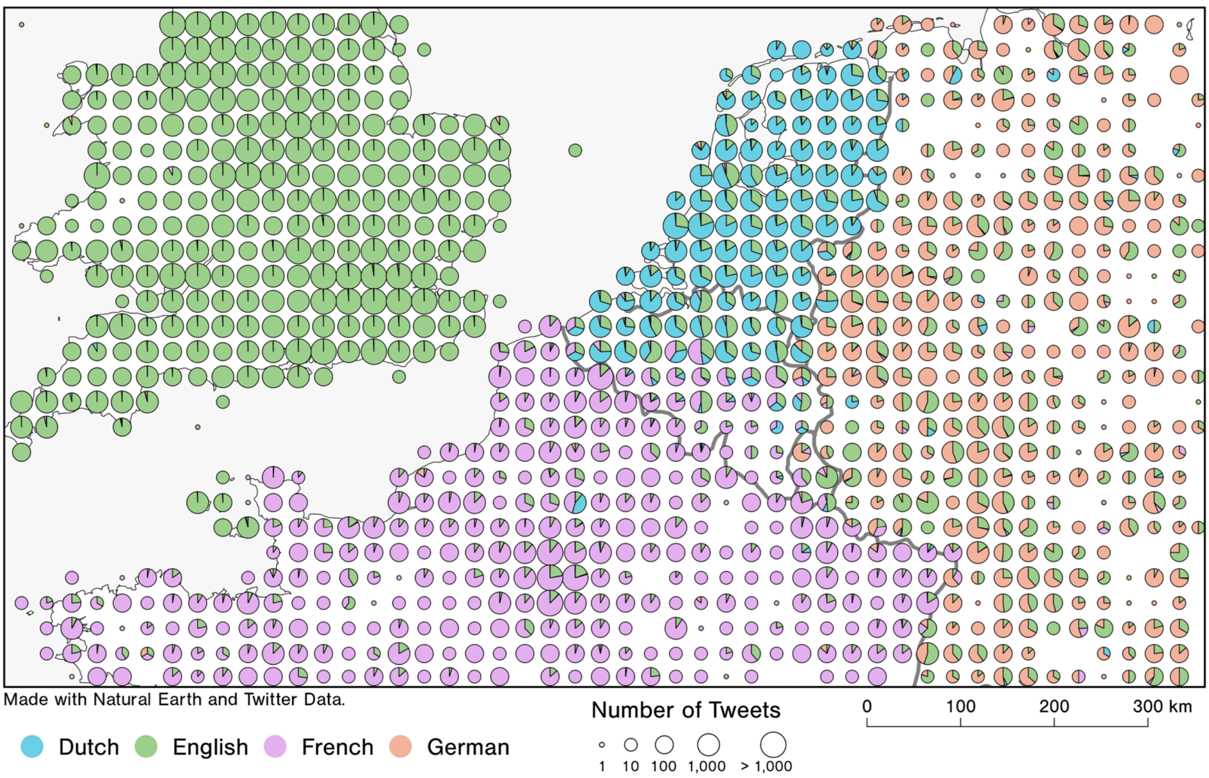


Figure 15: Micro diagram visualization of English, French, Dutch, and German tweets for January 1, 2019.

3.4 The micro diagram and its parameters

The micro diagram, or micro grid diagram, aims to provide an overview and details simultaneously, using small diagrams arranged in a regular grid. The size of those diagrams depends on the size of the regular units and the number of accumulated values in one cell. In addition, the diagram visualizes the numerical proportion of a categorical attribute. The following paragraphs introduce the micro diagram creation process with its parameters.

To achieve this visualization, some processing steps are necessary. Figure 16 illustrates the entire visualization workflow for creating micro diagrams with the processing steps, applied parameters, and intermediate results. The first task is aggregating the primary point data into reference units. The parameters at this step are types of units (spatial data structure or regular units/areas), shapes of units (hexagons, rectangles, squares), and spatial resolution. In the next processing step, appropriate colors are chosen for the number of categories and a diagram type that fits the units for aggregation is identified. Next to diagram type, diagram size represents the amount of data per cell, which can be executed in discrete steps or continuously. This choice results in either a more or less complex visualization. Finally, combining the thematic layer and a reference map results in a complete micro diagram map.

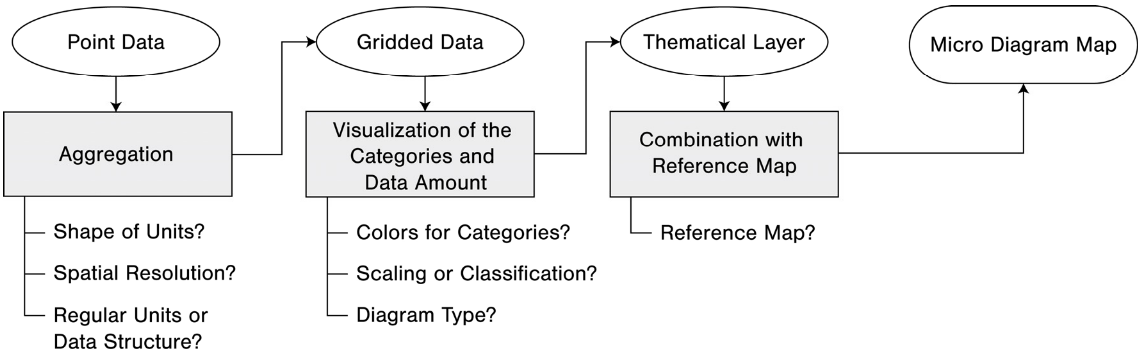


Figure 16: Workflow for micro diagram creation. The processing steps are in gray boxes, the parameters ae below, and the intermediate results are in the white ovals.

3.4.1 Aggregating points into a regular structure

The processing chain starts with the non-uniformly distributed points with categorical attributes, as in Figure 12. Overall, the data contain regions with low and high data densities, aggregated to summarize the data in comparable units. This approach often utilizes a grid, or sometimes a graticule, to construct regular units (Bollmann 2001, sec. Felderprinzip). The resulting units can be squares, rectangles, or trapezoids, but triangles or hexagons are also possible, which look more native and less artificial (Arnberger 1993, 145). Finally, the resulting areas with the contained points can be summed up. This method is suitable for various applications due to its adjustable size, shape, and independence from administrative or functional units. The creation process for these units is restricted to one map projection and, consequently, is not applicable worldwide. The global grid approach, with nearly constant cell size, could overcome this restriction (Ben, Tong, and Chen 2010; Lin et al. 2018).

The sample maps in Figure 17 show examples of the different grid types for comparison. Figure 17a visualizes the raw point data as a starting point. Figure 17b shows a simple grid, while the maps in Figure 17c and Figure 17f demonstrate examples of constructed grids as hexagons and rhombuses, respectively. In contrast to the always present grids in the most examples, in the maps in Figure 17d and Figure 17e, the units are only visible where data are available due to their construction using the particular spatial data structures.

Spatial data structures can also achieve similar results as grids by dividing the space into regular pieces to generate an overview, which can be used in databases to achieve faster data access. Furthermore, spatial data structures can represent hierarchies with rougher and finer resolutions and can be used as indexes. The quadtree in Figure 17f is an example of such a hierarchical data structure (Samet 1984). In general, the quadtree decomposes a space into equal parts on each level and stops dividing the space if a defined number of features exist in each tile. This property groups the data into small regular clusters with defined areas or numbers of points. Another index structure is the geohash ("Geohash" 2017), which uses a string representation for each coordinate tuple. Removing characters from the end of the string reduces accuracy. The geohash summarizes nearby points by dropping the last characters, reducing only coordinate precision and allowing rectangles to be built that approximate the potential bounding box of these points (Figure 17d). A restriction of the geohash is that this approach only works with geographic WGS84 coordinates, and the resulting rectangles differ in size and shape depending on the map projection, scale, and region on the earth.

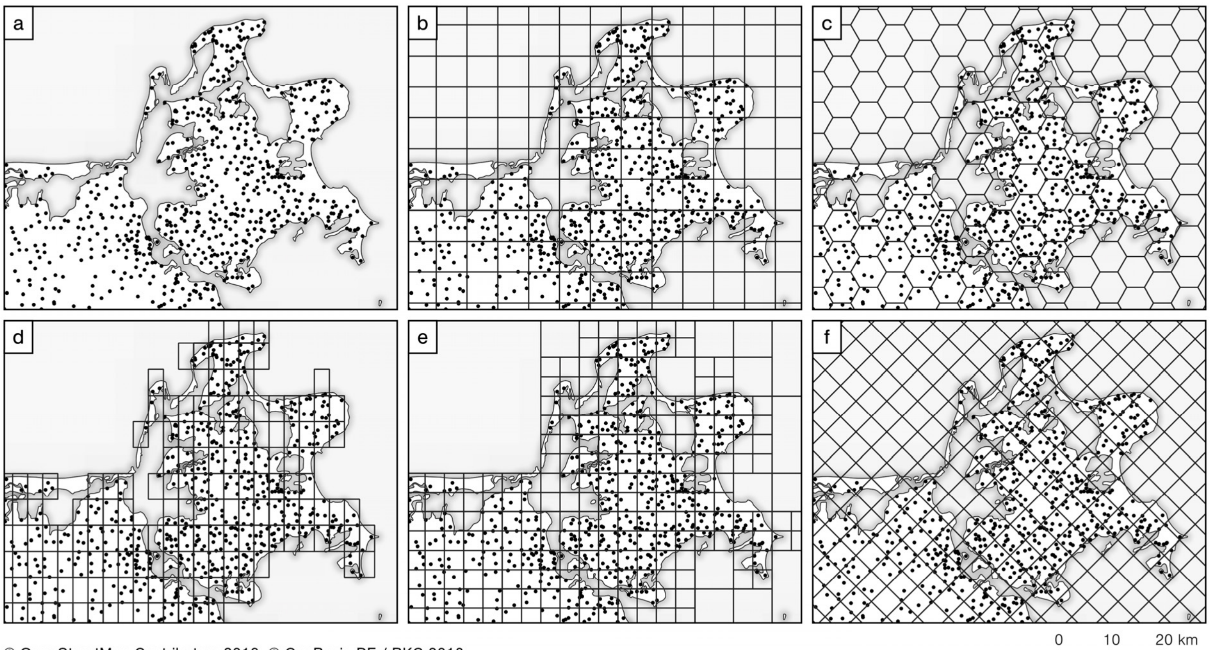


Figure 17: Examples of the different grids. Map (a) shows the raw point data, (b) shows a regular grid, (c) shows hexagons, (d) shows the bounding boxes pattern of the geohash, (e) shows the patterns from a quadtree, and (f) shows rhombuses,

3.4.2 Visualizing the number of data points

After the points have been summarized using grids or spatial data structures, the number of aggregated values can be visualized. For the visualization of these ratio-scaled numbers, it is common to use the visual variable size (Bertin 1974). An appropriate approach is to scale the symbol area according to the number of aggregated points. The result is a proportional symbol grid map, already known as the regular order of the single symbols of geometrically-shaped symbols (Kraak and Ormeling 2010, 144). The area of a symbol scales with the number of aggregated points based on a defined function. For a circle, the diameter can be calculated as follows (Töpfer 1974, 97):

$$d = \sqrt{\frac{4}{\pi} * A_0 * N_i}$$

The d is the diameter of the circle, the A_0 area for one point, and N_i the number of points.

The minimum area for one point is a circle with an area of 0.3 square millimeters, which results in a diameter of approximately 0.6 millimeters, according to the minimum dimensions for a single-colored point from the literature (Hake, Grünreich, and Meng 2002), because the color should be used to distinguish between categories that are combined with diagrams. For smaller diameters, it would not be possible to identify the color to distinguish between categories. A circle should have a maximum diameter of 3.3 millimeters and, thus, an area of approximately 8.5 square millimeters. This assumption ensures Bertin's recommendation of using, on average, not more than ten features per 100 square millimeters, as he proposes in his examples (Bertin 1974, 135, 145, 198). In the case of a square, the side length is based on the maximum area of the circle. The result is a side length of 2.9 millimeters, ensuring enough space for the reference map to remain visible.

The formula for calculating the diameter assumes a linear relationship between the number of points and the symbol area. In the case of a large number of points, a high diameter will not fit within the defined maximum limit of a 3.3 millimeter diameter. A solution can be using a nonlinear function or extending the formula above by adaptively scaling the proportional symbols (Slocum et al. 2009, 305; Töpfer 1974, 94). On the one hand, this is a very versatile solution but on the other hand, it is more difficult to read and interpret the values. A better solution that might fulfill the described requirement could be using graduated symbols with three to five classes to categorize the number of points. Graduated symbols are easier to read, although they do not offer information about the exact data amounts.

3.4.3 Grid and micro diagrams

The maximum size of the diagrams should correspond to the spatial resolution of the grid for aggregation and map scale. In the ideal scenario, the spatial resolution of the grid cells is adapted to the threshold value of the maximum symbol size. Due to the structure and functionality of the spatial data structures, it is not always possible to follow this rule. For example, in the case of the geohash spatial data structure, only a discrete number of aggregation levels and grid resolutions can be selected. In these cases, the grid cells should be larger, up to 5 millimeters in diameter, rather than smaller for better results.

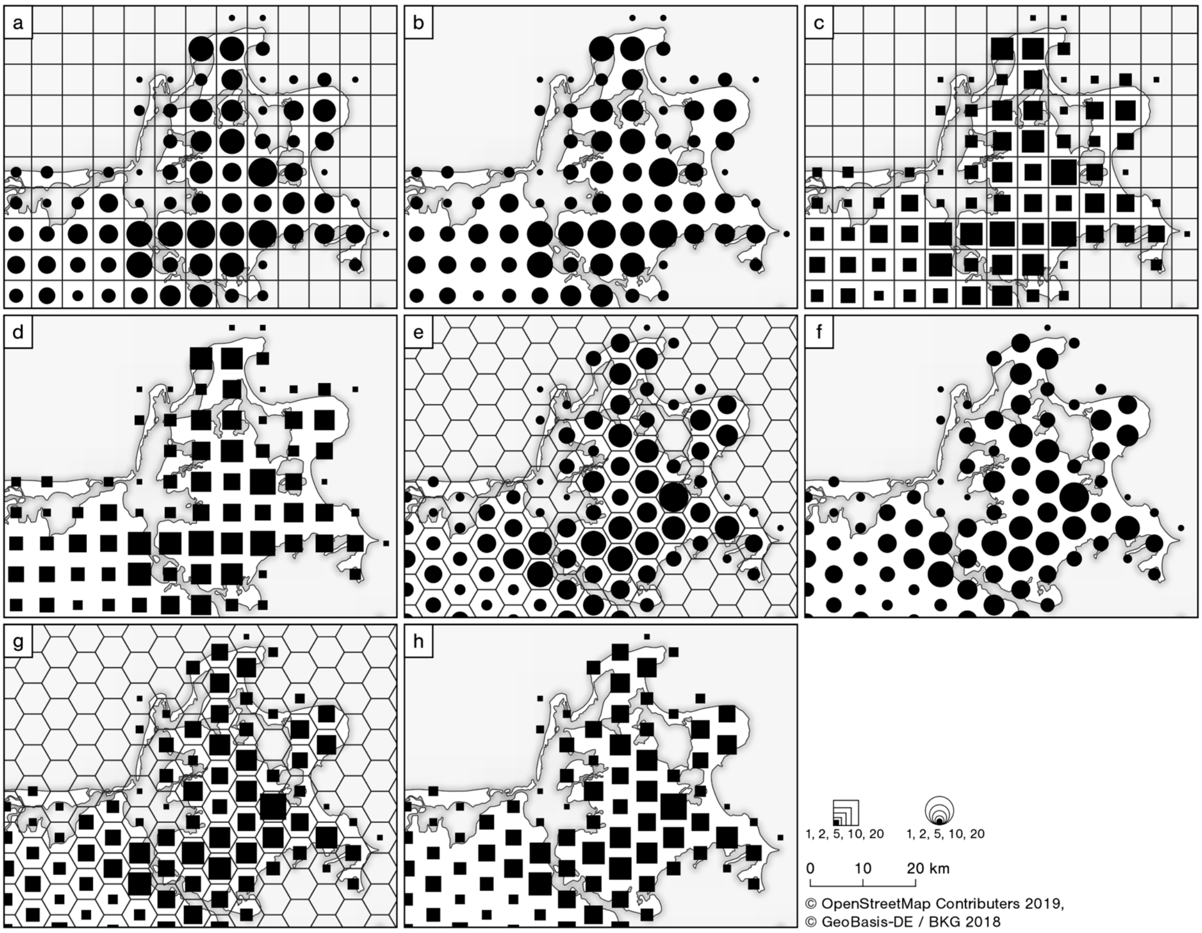


Figure 18: The maps without the visible grid illustrate how a micro diagram map can look using this grid type and diagram shapes. Samples (a) to (d) show how the circles and squares fit into a square grid. The examples in (b) and (d) display the results without the grid lines. Maps (e) to (h) use hexagon cells, again in combination with circles and squares. Subfigure (f) and (h) demonstrate the results without the grid.

Other parameters are the shapes of the grid units and the diagrams. Figure 18 shows grid unit example variations and simple geometric shapes, which can be the outlines of a diagram. Circles fill the hexagons better (Figure 18e), while squares fill the rectangular grid cells better (Figure 18c). The circles in a visible grid look good (Figure 18a), while the squares in hexagons (Figure 18g) seem ill-suited. The micro diagrams usually work without showing the grid (see examples in Figure 18b, d, f, and h) and allow for the use of all shapes and grids.

Nevertheless, how much space is between the invisible grid and the diagram is essential for the resulting map. If the shape fits nicely into the grid, there is less space for the reference map. Conversely, incompletely filled grid cells mean more information comes from the reference map.

In addition, there is a difference in diagram arrangements. While the rectangular grid creates an even and rigid grid look, the hexagon cells imitate a natural pattern, e.g., honeycombs, basalt columns, or snowflakes, leading to a more harmonious look. The result is a somewhat natural-looking pattern. The combination of square grid units and circles in the example visualizations is preferred because it emphasizes the grid character. It shows that the data are aggregated, look harmonious, and offers space to the reference map. Hexagon grinding produces excellent results but hides more of the grid character, which should be shown in our examples to demonstrate the idea of the approach.

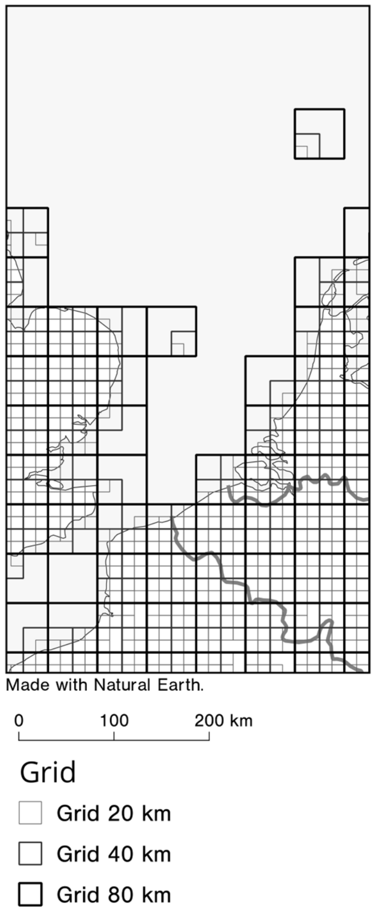


Figure 19: Grids for micro diagram creation. Only grid cells with data are shown.

3.4.4 Visualizing numerical proportions with diagrams

The diagrams are used to show the proportions of the individual categories. The selection of usable diagram types is limited to compact area proportional diagrams, which are appropriate for medium- to high-value margins. This requirement excludes all length-proportional diagrams and the used grid structure in which the diagrams should fit.

It is well known that the frequently used pie charts are typically not cartographers' first choice (Imhof 1972, 77), but their pleasing shape offers advantages for map design. The judgment of areas is less accurate than the judgments of lengths, but pie charts are more appropriate for comparing combinations than bar charts (Spence and Lewandowsky 1991). Nevertheless, rectangles are more suitable for reading values than circles (Arnberger 1993), but they are rarely used. A systematic overview of the cartography used in diagrams was provided by Schnabel (2007) and, combined with evaluations of various data usability (Tsorlini et al. 2017), helped us select the appropriate types of diagrams.

Figure 20 shows examples of different diagram types used as micro diagrams. The polar area chart and the wing chart can deal with just four categories, while the divided area chart and the pie chart do not have this limitation. In exchange, the polar area and wing charts can show evident category null values. The divided area and pie chart are very compact and fit better with the grid. In particular, the divided area diagram is weak visualizing small values because the areas have small stripes. Due to their frequent implementation and harmonious shapes, pie charts are primarily used for examples and research.

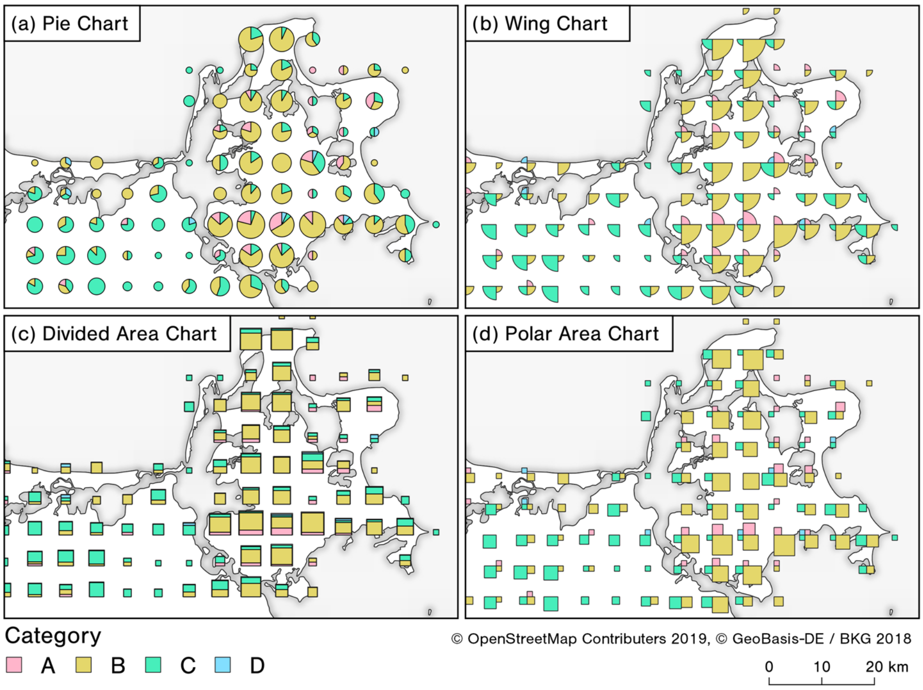


Figure 20: Example maps for the usable micro diagram chart types: (a) pie chart, (b) wing chart, (c) divided area chart, and (d) polar area chart.

The number of possible micro diagram categories is limited due to the diagram type, the number of visually distinguishable colors, and the minimum visible area for a category in a diagram. In particular, Figure 20 displays examples that clearly show that the diagram type limits the number of categories since the visibility of a category in a divided area diagram is lower than in a pie chart. The exact number of visualizable categories depends on the diagram's size and the spatial distribution of the categories. According to ColorBrewer (Brewer 2005), twelve distinguishable colors are possible. That is much more than is possible in a micro diagram and is a minor limitation of the approach.

3.4.5 Influence of color and color brightness

The selection of colors to differentiate the categories is an important task that can easily manipulate a map's appearance. Using a prominent color, such as red, for a category with a high percentage can lead to the impression that this is the most critical category, steering the interpretation of the map in a specific direction. Comparatively, using such a prominent color for a category with a relatively low percentage can help keep this category visible. Keeping this in mind, colors should be chosen with care.

Designed for choropleth mapping, the ColorBrewer (Cynthia Brewer 2017; Brewer 2005) can help select colors. Another option is hclwizard ("HCL Wizard - Somewhere over the Rainbow" n.d.; Zeileis et al. 2019; Zeileis, Hornik, and Murrell 2009), which produces a perceptually uniform color palette for different application fields that are easy to distinguish and have the same level of brightness. This tool is often used for selecting colors in statistical graphs and uses a color model adapted to human perception. Based on the assumption that colors with the same level of brightness attract an equal amount of attention (Bollmann 2001, para. Farbhelligkeit), this tool ensures that the selected colors do not overemphasize any particular category.

Figure 21 compares colors selected from ColorBrewer (a) and colors that were generated with hclwizard (b). While the yellow on map (a) attracts the most attention because of its brightness, the colors on map (b) have nearly the same level of brightness, with no color attracting noticeably more attention than another, which may prevent overweighting a category. ColorBrewer also offers palettes with similar and often varying brightness values, which is more beneficial for choropleth maps. Both tools help choose the colors, but the results can differ and bias the map reader toward different conclusions for the same visualization.

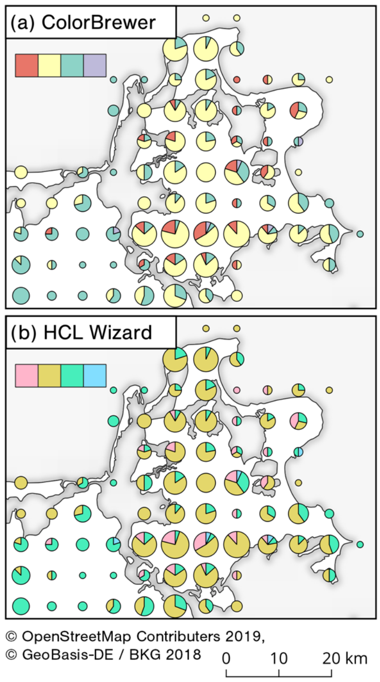


Figure 21: Map (a) uses "Set3" from ColorBrewer and map (b) uses self-defined colors from hclwizard.

3.4.6 Interaction options with micro diagrams

Micro diagrams are not only a static mapping method; they also promise to add interactive options to the visualization. These options are essential for conveying detailed information. Possible visualization interactions include showing the exact numerical distribution of the category values when the pointer hovers over a diagram; offering a magnifying glass if a user wants to see the details of a region; changing the category colors to regroup the information; and zooming in and changing the map.

In addition, it is possible to give the user control of the parameters when creating the micro diagrams. The tools can include adjusting the grid resolution, changing the classification, scaling the diagram size, selecting the amount of free space around the diagram, turning the aggregation grid on or off, choosing the diagram type, reclassifying the data with different methods that affect the categories and their distribution, and selecting the reference map. These options can turn micro diagrams into powerful tools for geospatial analysis.

3.5 Application and user-based evaluation

This section presents use cases of micro diagrams visualizing point collection and their perception. The new visualization approach is discussed from three selected viewpoints: mapmakers, analysts, and map readers.

3.5.1 Micro diagrams in a multiscale environment

Abstraction and scale are crucial factors in map creation that require a closer look. For discussion, the example map from this chapter is revisited to facilitate the comparison.

3.5.1.1 Multiscale use case

Figure 22 visualizes the Twitter data from the introduction example (Figure 12, Figure 13, Figure 14, and Figure 15) on three different scales. The map in the middle has the same scale as the other samples, while the map on the left has more visible details due to the lower scale number. The right map offers more abstraction due to the higher scale number. Figure 19 shows the grids behind the micro diagrams in Figure 22. The tweets are situated inside the 80 km grid, subdivided by the 40 km and 20 km grids, because a grid cell is only drawn if there's at least one tweet inside.

Across all three maps, the micro diagram dimension remains constant. There is no change in grid size, colors, classifications, or diagram sizes. The grid aggregation is matched with the chosen scale to recreate similar looks. Otherwise, the maps would not be comparable since the diagrams would grow too big and no longer be micro diagrams.

Generally, a greater scale leads to more substantial abstraction, as the right map in Figure 22 shows. The left map contains more details, but for a smaller area—regions with sparse data become more visible and more details are noticeable. For example, in Great Britain, a small number of German tweets is only noticeable on this scale. With each step toward the right, the aggregation level rises, leading to a more evenly filled map at the cost of visualization precision.

Compared to the previous maps (Figure 12, Figure 13, Figure 14, and Figure 15) on this matter, the micro diagram visualization offers a clutter-free and quantitative estimation of the Twitter data, which works on different scales and for four variables within the same graphic.

3.5.1.2 Micro diagram map perspectives

Different readers perceive a map differently, but they are all guided by their existing knowledge and map-reading skills. This fact needs to be considered when creating a complex graphic, e.g., a micro diagram visualization.

For a non-expert map reader, the visualization should be as simple as possible, offering a good overview without too many details and only necessary variables. An analyst, however, wants to explore data using the map and is interested in details and different views of the same dataset. Complex visualization and many variables are acceptable as long as they are purposeful and legible. A micro diagram map targets the analyst as a user because it is a complex visualization that can lead to different conclusions using a set of different visual variables.

There is a workflow showing all the options for a cartographer who wants to create a micro diagram. The parameters discussed in the workflow on page 33 should be chosen with map reading skills and interests in mind. All of these parameters are part of the design process and should be coordinated with the map reader's skills and interests. In addition, age and possible disabilities can influence the size and color of the micro diagram map. In contrast, the number of classes and grid size depends more on the general audience—experts appreciate more details, while ordinary map users should not be overwhelmed.

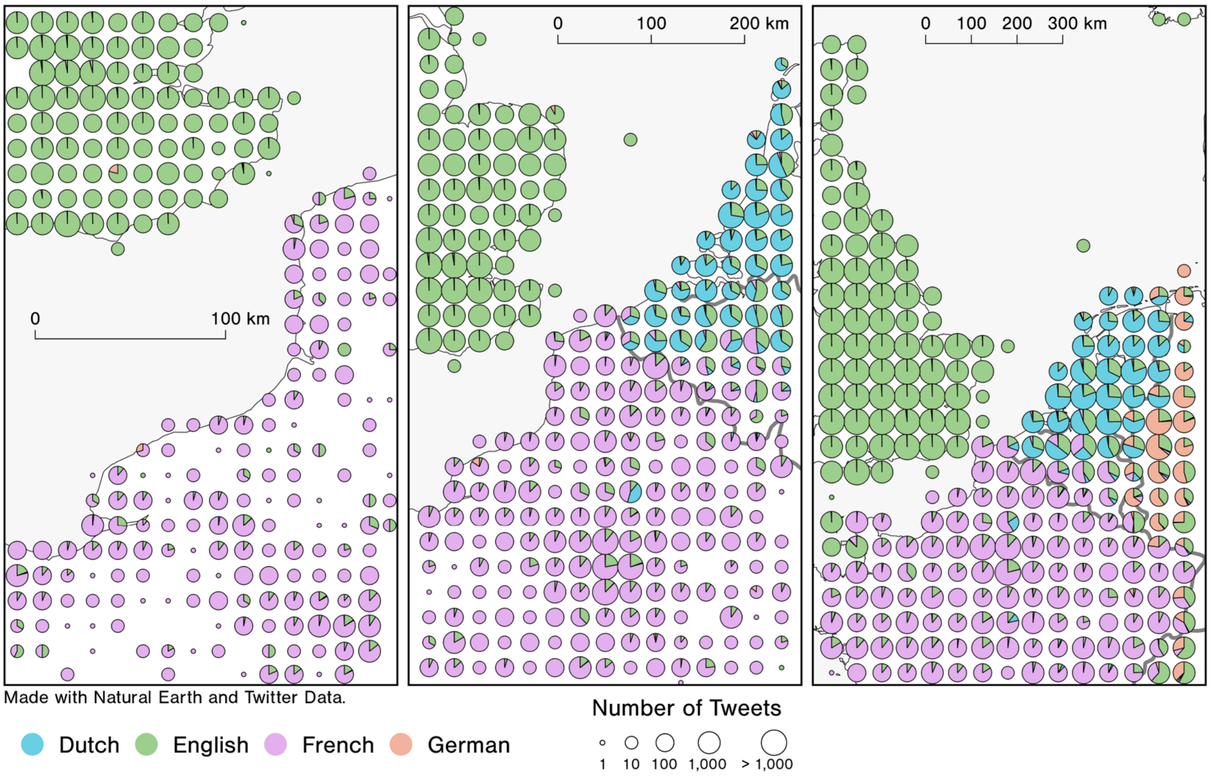


Figure 22: Micro diagram visualization of tweet languages from one day in three different scales. The scale is doubled for each map from left to right.

3.5.2 The micro diagram user study

The study aims to evaluate the new micro diagram visualization method. The advantages and possible problems of this approach for visualizing point collection data should be identified.

3.5.2.1 Objective

A user study was conducted with the help of a master's thesis (Friedel, 2018), verifying that the users understood how to interpret micro diagrams. The study also supports the advantages of the new visualization method, as well as identifies the users' preferences regarding the point map and the micro diagrams, according to the estimating numerical proportions and quantities tasks.

3.5.2.2 Structure

At the beginning of the online survey, information about the participants' education, gender, age, use of visual aids, and cartographic background knowledge was collected. The survey started with an explanation about micro diagrams, followed by a question to verify that the users understood the visualizations and had not guessed.

Each participant had to solve four tasks of increasing difficulty related to the point method and the micro diagrams. Everyone was asked to solve the four tasks twice for a marked grid cell or in a functional region. The functional region, a district of Dresden, as shown in Figure 23, is an example of an administrative unit. A functional unit, such as a district, is often used in the application, e.g., administration and urban planning. Ideally, the map reader could summarize the information for this region, draw conclusions, and make decisions based on the visualized data. For the survey tasks, thirty different maps were produced, similar to the previous examples, to prevent the learning effect from always using the same examples. Finally, the participants were asked whether they preferred to use the point map or the micro diagrams to solve the tasks.

3.5.2.3 User tasks

The four defined tasks are listed below. Two micro diagram examples and two point map examples were provided for each task, requiring a total of four sub-questions per task to be answered.

- (A) Identify the dominant category in a grid cell and a functional region.
- (B) Estimate the category percentage for a grid cell and a functional region.
- (C) Estimate the overall amount for all categories in a grid cell and a functional region.
- (D) Estimate the category value for a grid cell and a functional region.

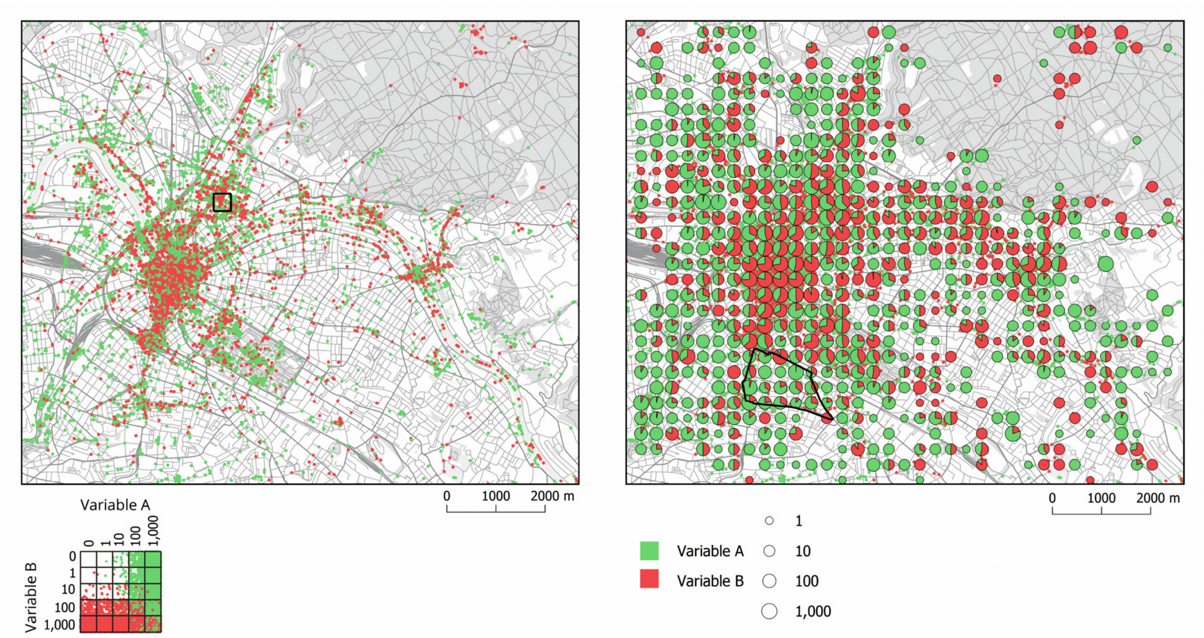


Figure 23: Example maps from the user test. The left map shows an example of the tasks related to a grid cell on a point map, while the right map shows an example of the tasks related to a functional region on a micro diagram map.

Task (A) is a simple decision question—is a region redder or greener? It is more difficult to estimate the number proportion between the two categories in task B. The answer to task C is the estimated amount of data in the marked grid cell or functional region. For this task, the map reader had to use and understand the legend to answer this question. Finally, task D asked for the value of a category. To answer this question, combining the knowledge of tasks B and C is necessary.

3.5.2.4 Participants

Seventeen people participated in the survey, of which fourteen held a university degree. As Figure 25 shows, the majority of participants had some experience with point symbols.

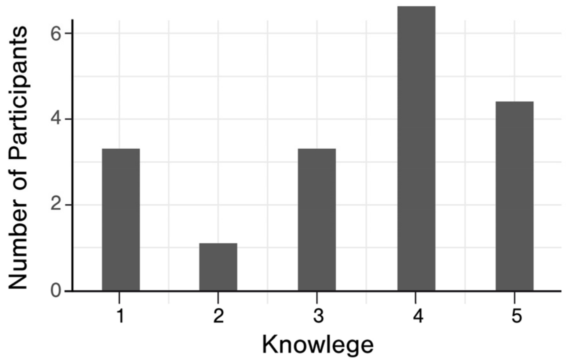


Figure 25: Participant self-assessment of their knowledge of cartographic point symbols. One means no knowledge, while five means expert knowledge.

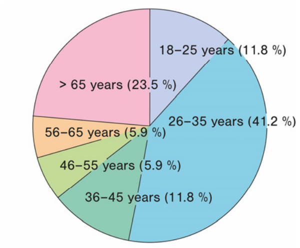


Figure 24: Ages of the survey participants.

According to Figure 24, most study participants were between 25 and 35 years old, with the second largest group over 65 years old (23.5%). In the user test evaluation, the results of two participants were removed due to rough mistakes in the answers. Seven of the remaining participants indicated that they used visual aids. Everyone answered the micro diagram visualization question correctly, proving that they read the explanation about the micro diagram method.

3.5.2.5 Results

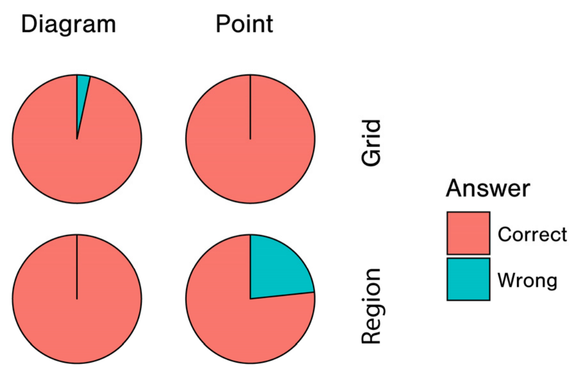


Figure 26: Evaluation of task A: Estimation of the dominant category. Each diagram corresponds to one question in the survey.

Task A was to identify the dominant category in the marked region; an example map is shown in Figure 23. The corresponding answers in Figure 26 show that the micro diagrams allowed more participants to estimate the dominant category correctly.

In task B, which asked participants to estimate the percentage of variables A and B, the standard deviation for the micro diagrams is notably lower than for the point method. The mean standard deviation for the micro diagrams is 6.0, while the mean standard deviation is much higher for the point method at 15.1. In addition, Figure 27 shows that the standard deviation tends to be higher if the question refers to a functional region. The reason for this behavior could be unclear regional borders in the micro diagram grids. The grid cells do not match the border of a function, leading to different estimations.

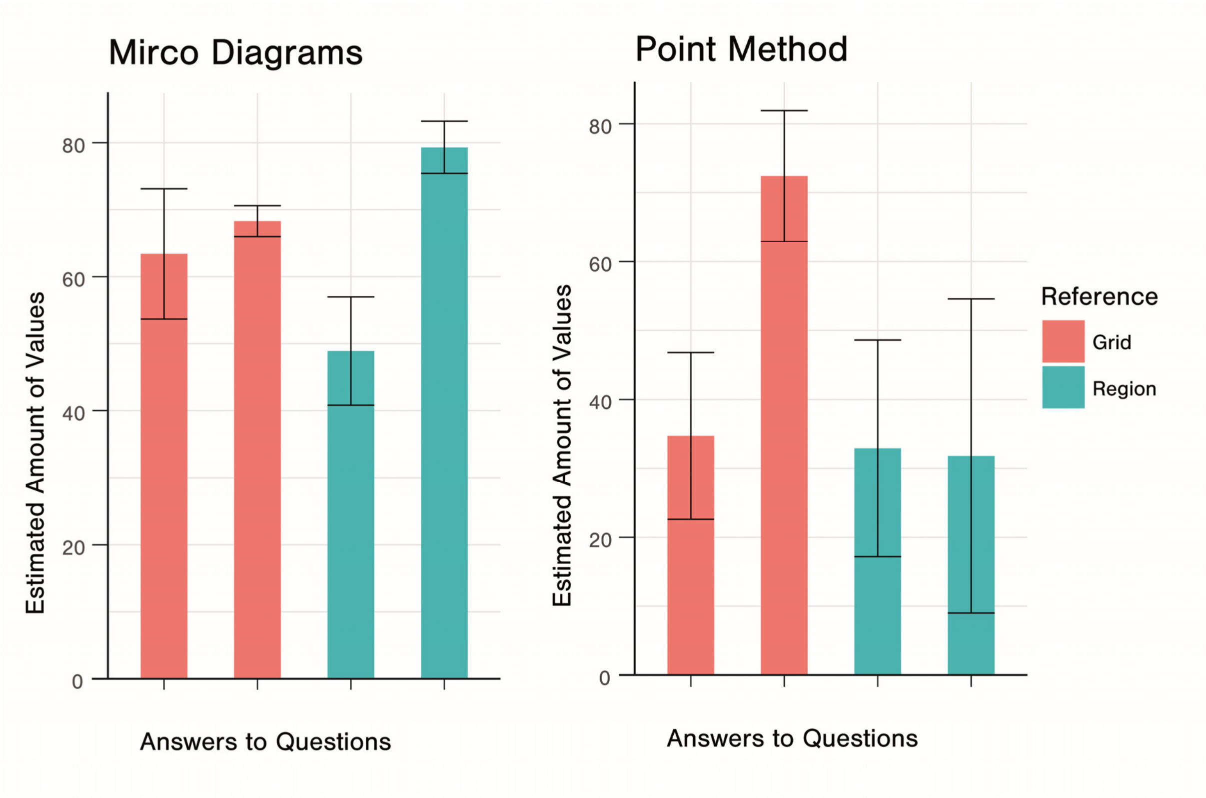


Figure 27: Evaluation of task B: Estimation of the percentage for a category. Each bar of the diagram corresponds to one question in the survey.

Results for tasks C and D are similar, shown in Figure 28 and Figure 29, respectively. The standard deviations are very high for reading the amount of data or values for one category. In the end, eleven participants stated that they preferred the micro diagrams for solving tasks, four gave no preferences, and no one preferred the point map.

3.5.2.6 Discussion and conclusions

A challenge in creating the user study map was constructing a legend for the point method, which enables the map reader to estimate the number of values. Figure 23 shows a possible solution, which provides a matrix for the categories and the number of points. The result of the study should be seen more as a trend because of the limited expressiveness due to the sample size of 17 participants.

In general, the high standard deviations indicate, it seemed very hard for even the most experienced participants to read the values. Estimating the values using micro diagrams works better than using a point map. However, this method only allows for a rough determination of the values, as reflected by the standard deviations in the evaluations of tasks C and D. For the functional regions, the estimation resulted in higher standard deviations, which corresponds with the user comments that it was harder to estimate the values in this case. The micro diagrams offer noticeable advantages over the point method for tasks A and B, such as being able to read the numerical proportions in grid cells and regions.

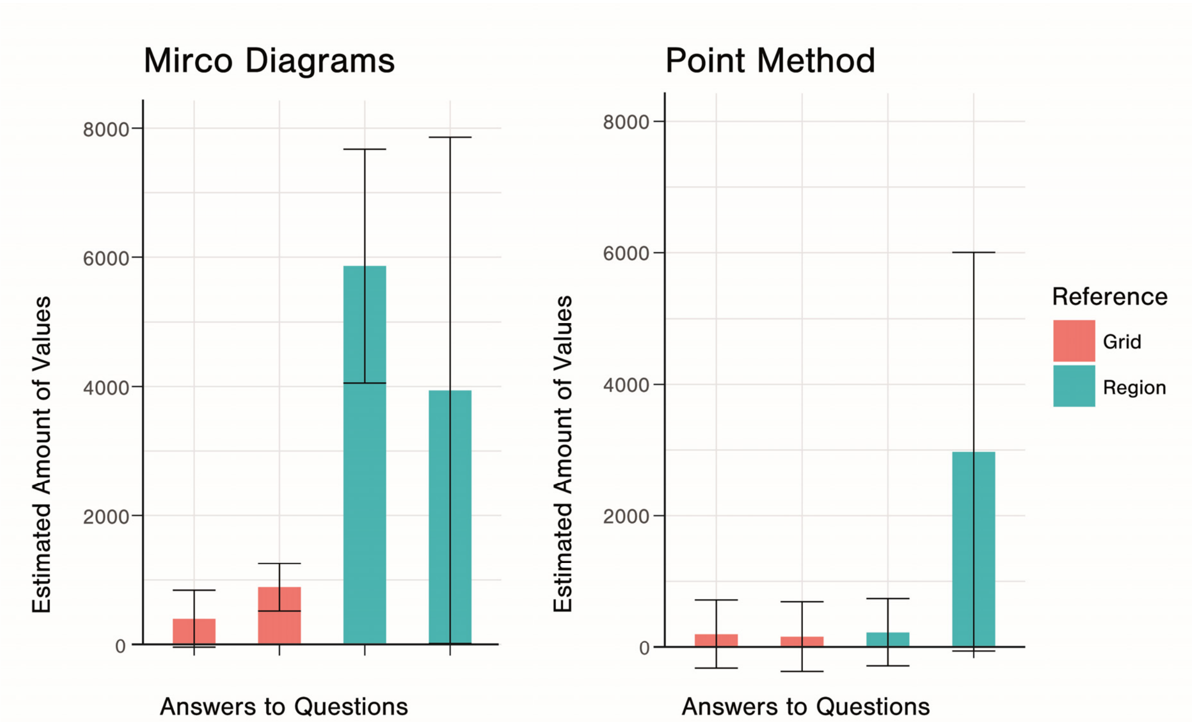


Figure 28: Evaluation of task C: Estimation of the overall amount for all categories. Each bar of the diagram corresponds to one question in the survey.

According to the results, it can be concluded that the micro diagram visualization method offers some advantages over the point method. Micro diagrams make it easier to determine the dominant category and the percentage in a category. Reading specific values also works better in the micro diagram than in the point map. Estimating values with a point map is complex, and the method works more qualitatively than quantitatively. The participants understood the visualization and obtained rough information about the amount of data and the numbers for one category when using the micro diagram method.

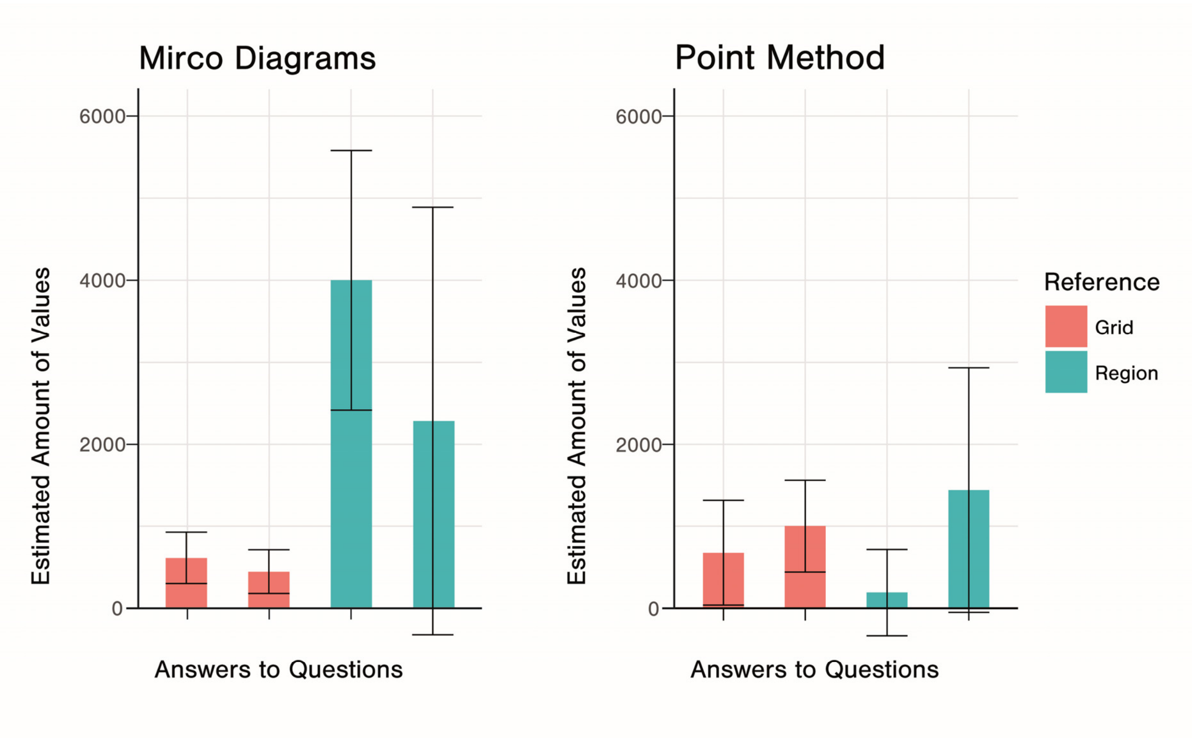


Figure 29: Evaluation of task D: Estimation of the overall amount for all categories. Each bar of the diagram corresponds to one question in the survey.

3.5.3 Point collection visualization discussion

The use case and the user test demonstrated that micro diagrams suit specific tasks. The use case revealed the advantages of the point method for showing detailed distributions of point data. Figure 8 illustrates this example in contrast to the more abstract visualization with micro diagrams in Figure 9. On the one hand, the details in the point map can be fascinating and relevant; on the other hand, the micro diagram map offers a better overview of the numerical proportions of the diagrams and more accurate information about the amount of data beyond the general statement "there are many points".

Table 2 offers an overview of possible target user tasks and their fulfillment according to the case study and user test. Overall, the point map allows a better look at spatial distribution with more details. In contrast, the micro diagrams offer better quantitative and more detailed information. The user test also validates that it is possible to estimate the amount of data using the point map and micro diagrams. Point maps might be the better choice for an analyst to form the first visualization since they provide detailed representations of spatial distributions and are easy-to-produce visualizations for an average GIS user. The result is an overview of the spatial data distribution that allows an estimation of the relative distribution of data and categories. However, if a more accurate analysis, comparison of categories, or detection of hotspots is needed, then the micro diagrams are more appropriate. The choice between these two visualization methods depends on the objective, the topic of the visualization, and the user group.

Table 2: Comparison of visualization approaches. One star means less appropriate, and three stars are the most suitable.

Task	Point maps	Micro diagrams
Determine the spatial distribution	***	**
Determine the amount of data	*	**
Determine the dominant category	**	***
Determine the proportion of categories	*	***
Determine the combination of categories	**	***
Determine the hot spots	*	***
Compare the amounts of data	**	***

3.5.3.1 Applications

Micro diagrams are suitable for use cases requiring deeper analyses. The complex and time-consuming creation process is not suitable for quick visualization, and the resulting graphic needs time to be understood and read. The user test participants were all, to some degree, visualization experts and were able to solve the tasks. A comparison with the average map reader was not possible due to the small number of participants, and it was also not the aim of the study. The most likely applications could be analyzing location-based social media data, comparing the values for different indexes, or exploring census data (e.g., race, region, and education). These application cases are performed by experts such as data scientists, analysts, and decision-makers, who are similar to the user test participants and can benefit from the wide range of parameters that can be used to create micro diagrams. Using different parameters also allows the exploration of the data in detail and adjustments to the visualization for specific tasks.

3.5.3.2 Parameters

The choice of aggregation method depends on the data source, the objective, and the amount of data. If the data are already stored in a form similar to a grid structure, aggregation is unnecessary. Aggregation could also be performed to anonymize data (e.g., census data). If deeper analysis is needed, the spatial data structures offer more options for generating different aggregation levels. The use of grid units is primarily suitable for visualization, mainly because they can be adjusted to meet specific needs resulting from a fixed scale or a desired abstract representation. More data helps obtain better analysis results; in this instance, the micro diagrams offer benefits over the point maps. The underlying grid data structure provides an index for fast processing and supports aggregation.

Diagram size is an important parameter. If the diagram is too small, it gets difficult to read the category colors or determine the amount of data via size. There is a fine line between too large and too small and unreadable, which is also influenced by the aggregation parameters. The user test showed that the micro diagrams allow value estimations. Some interaction techniques can simplify this for users and provide better readability.

In the example, mostly pie charts were used to show numerical proportions because of their flexible usage and broad implementation. All other diagrams are usually unavailable out of the box and must be implemented. For example, Figure 20 shows that other types of diagrams can also fit the requirements of the micro diagram method.

Nevertheless, the design of the reference map influences the readability of point or micro diagram maps. The reference map should contain enough information to allow the meaning to be inferred with enough space, but not too much, to keep the focus on the thematic information (Arnberger, 1977). The current examples always included a grayscale reference map with only the most essential features, e.g., sea, land, and borders. Depending on the scale, large cities, water bodies, and hill shading are certainly conceivable. The colors must always be correctly assigned to the categories or objects; otherwise, the readability and the expressiveness of the map will presumably be strongly impaired. These requirements are not fulfilled by any standard map style and require the creation of a unique style.

3.5.3.3 Further questions

Several subjects for further investigation would be interesting in addition to the presented theoretical considerations and case studies: What happens if the grid size changes or the origin moves? How can changes in the map scale and the degree of aggregation be combined or decoupled? On the one hand, the aggregation should always remain adjusted to the scale for optimal readability. On the other hand, changing the aggregation can open new perspectives and lead to new findings. Similarly, this is also possible by moving the grid origin. This may strongly impact the number of objects in each grid cell, which is reinforced by the classified visualization of the amount. Therefore, ideally, interactive methods offering options for deeper analyses should be made available to analysts.

3.5.3.4 Conclusion

Micro diagrams are a new visualization method that can handle a high amount of point data, which is typical for LBSM data. It resolves cluttering effects through grid aggregation. The result of aggregation and visualization by the diagram is a well-organized map, similar to a cartogram but not distorted like a tile map (McNeill and Hale 2017), that can be combined with a reference map. Similar cartographic visualization approaches (Bleisch and Hollenstein 2017; Heimerl et al. 2018) exist in the literature for other geometry types.

Micro diagrams

Micro diagrams are a suitable method for visualizing generic point data sets, allowing the map user to estimate the number of points and numerical proportions of the categories.

Reading and understanding can take some time due to the high information density in a micro diagram map. Advanced map readers, such as data scientists, analysts, and decision-makers, can benefit from it. The methods creation process offers a wide range of parameters that can modify the visualization for specific cases and personal needs by adjusting the aggregation, the diagram, or the data classification. In addition, aggregation allows data anonymization and protects the privacy of sensitive data, which can be LBSM or even statistical data from a census.

4 Selection of POIs for visualization

The following chapter deals with different selection approaches for POIs or similar objects using space and one numerical attribute. Available solutions from the literature are explained, following a methods discussion covering less known approaches and new solutions not found in published cartographic literature. All methods are documented with a comparable example for better understanding. This is followed by an evaluation of the performance of the different approaches, use cases for the method of discrete isolation, and considerations of usage in multiscale mapping. The examples with different data sources should illustrate diverse applications. A critical review of the different use cases and a comparison of the approaches complete the discussion.

4.1 Approaches for point selection

Selection is a method for preserving significant features in the generalization process while decreasing the scale. Based on the feature's attributes, a decision has to be made to remove or maintain them depending on their spatial distribution. In generalization operator terminology, selection (Kraak and Ormeling 2010; Hake, Grünreich, and Meng 2002) is also abstracted as "refinement" (Slocum et al. 2009, 102; Stanislawski et al. 2014) or "selective omission" (Z. Li 2007).

How selection was performed previously is a matter of knowledge and experience for the conducting cartographer. Some recommendations are given in the literature, such as Arnberger's recommendation to grade settlements according to the number of inhabitants, function, and importance (Arnberger 1993). A cartographer training book (Laubert, Woska, and Habel 1988, 90) mentions additional criteria, including the area's character, settlement density, area of the settlement, importance of the settlement, and settlement type.

Today, new machine-learning techniques attempt to reproduce this complex selection process (Karsznia and Sielicka 2020; Karsznia and Weibel 2018). In the two example cases, the authors explore previous cartographers' decision criteria for settlements using machine-learning methods and eventually reach, from their point of view, equally satisfying results for their chosen sample. In order to achieve this goal, a high number of criteria, such as administrative, cultural, or economic factors, were collected, prepared, and used.

Overall, the settlement selection use case is a typical example, perhaps because it is the most common application or an obvious and understandable example. There were early attempts at algorithmic selection approaches that considered the frequency and distribution of the features (Srnka 1970). Additionally, different measures for selecting settlements, made with the help of a radius derived from the population or an R-tree (Shea 1988, 50), are also known.

Li offers an overview of selection methods and also explains some algorithms that are part of Sheas's report (Z. Li 2007, 81–84). The circle growth algorithm (Kreveld, Oostrum, and Snoeyink 1997) and settlement-spacing ratio algorithm (Langran 1986) are explained and illustrated. Both approaches rank point features but in different ways, which fits nicely in the model of automated settlement selection by Flewelling and Eggenhofer, who tried to formalize its importance (Flewelling and Egenhofer 1993). The settlement-spacing ratio algorithms start with the essential point, which is checked to see if a circle inversely proportional to the numerical importance measure contains an already selected point. If there is no point, the point gets selected; otherwise, the point is omitted. The circle-growth algorithm uses a circle, which is proportional to its importance. While processing, the circles grow until they overlap with a neighboring circle, which leads to a ranking.

4.2 Methods for point selection

In the following sections, three point selection methods are described and illustrated for the area shown in Figure 30. The map visualizes all populated places tagged as "place=town" or "place=city" in the same way to show the spatial distribution, with reference. It is well understood that a selection is necessary to achieve good representation without cluttering. All described selection approaches will be explained with a schematic profile and a map example. Settlements may not be the typical example for points of interest, which is understandable because the relationships are very vivid and data collection is simple. The numerical population attribute can be replaced by the number of photos, comments, visitors, mentions, or recommendations for a restaurant or tourist destination. Implementing the methods as a QGIS plugin[10] allows a comparison of the three methods and makes them available to more GIS users.

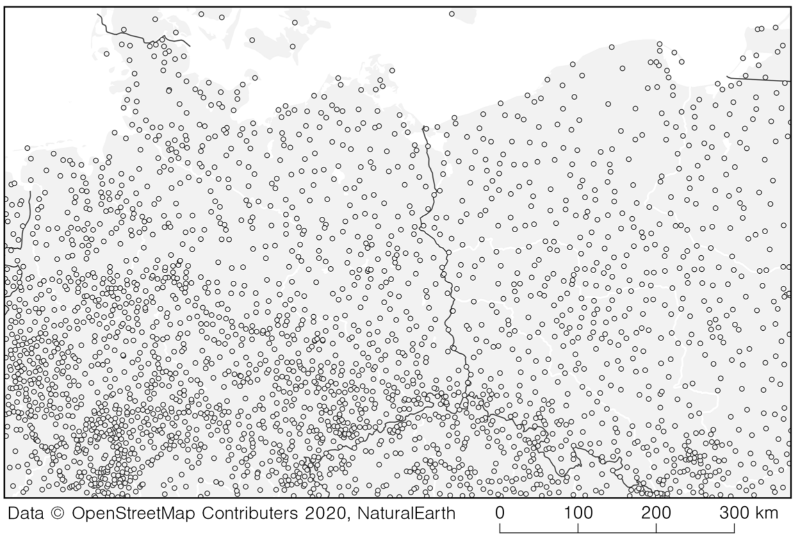


Figure 30: All places and towns from OpenStreetMap in the example extent.

[10] https://plugins.qgis.org/plugins/point_selection/ (accessed April 2021)

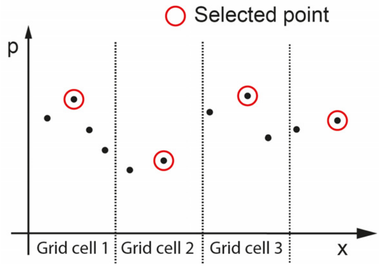


Figure 31: Schematic profile for the label grid approach: Selection of the point with the highest p-value in each grid cell. For example, p can be the population, while x describes the space.

4.2.1 Label grid approach

The first selection approach uses a grid—the point with the highest value is selected for an area defined by a rectangular grid cell, introduced and implemented as an SQL function for PostgreSQL/PostGIS from Mapbox.[11] This solution works well for populated places and can also be transferred to other point data. The grid width can be changed to adjust the density of the selected points. The SQL implementation uses squares, but it is also possible to work with a grid of diamonds, hexagons, or any other polygon in the QGIS plugin. The result is a ranking of the points inside the grid, offering the flexibility to vary the selected points number without re-computation by choosing the first six points, for example.

The method requires assigning the points to a grid cell in an implementation. Next, the ordering of all points a grid cell contains is followed according to a numerical value. The result can be stored as a boolean or an integer (as ranking) with the point. Figure 32 shows an example of the label grid method applied to the populated places from Figure 30. Only the place with the highest population in each normally invisible grid cell, with a height and width of 156 kilometers, is shown. In the case of a typical web map, this would be the size of one tile at zoom level eight, a scale of 1:2,000,000 at the equator. The result is mainly influenced by grid origin and grid cell size, as illustrated in Figure 31. The grid is a strict border and may show points with relatively low values, e.g., in grid cell (2). Moving the grid leads to a different result, so it is essential to use the same grid if a reproducible selection is desired. A loop is needed to find which point is contained by which grid cell, following a sorting of the values in each cell. The result is a complexity of $O(n * \log n)$ due to the label grid approach sorting algorithm.

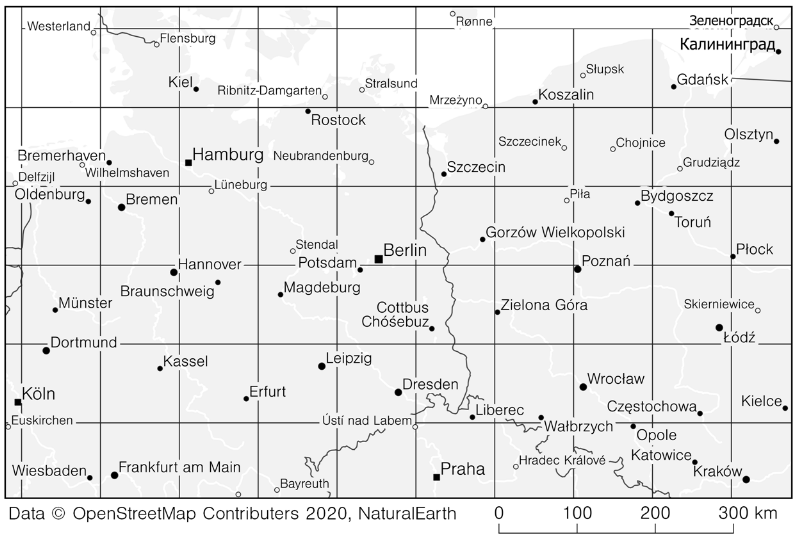


Figure 32: Example map for the label grid approach; only the place with the highest population in each grid cell is shown.

[11] https://github.com/mapbox/postgis-vt-util (accessed March 2021)

4.2.2 Functional importance approach

Functional importance models the significance of a place with a bell-shaped curve, lowering values by distance and numerical value, as Figure 33 shows. The difference between the function value of different points can be stored and used to select the points. This way of modeling interactions between objects is less common in generalization and was used by Hormann (Hormann 2021). According to documentation on the website, the following formula is used in the QGIS plugin (Hormann 2020; 2015a):

$$f(x) = p * e^{-\frac{d^2}{\beta}}$$

d… distance between the point geometries
p… numerical attribute value, e.g., population
β… variable to adjust how quickly the distance value is decreasing and to maintain distance between nearby points

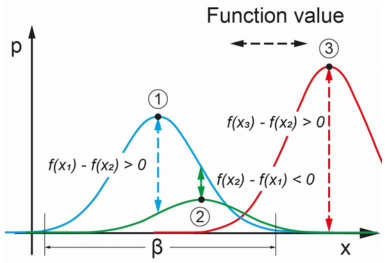


Figure 33: Visualization of the functional importance method. The function of each place is formed by the parameter population (p) and β. In cases of a positive difference between function values, the place should be shown on the map. In this example, points (1) and (3) would be selected. The x-axis shows the spatial distance.

Figure 33 illustrates the formula and shows the influence of the parameters. By adjusting variable β, it is possible to control the minimal distance between the points shown on the map. The numerical value, e.g., the population, also influences the selection or non-selection of the point because the difference between the function values must be greater than zero. For point (2), the value is too low for selection. It would be possible to use another function similar to the function $f(x) = -p^2$, to model functional importance with a reduced set of parameters.

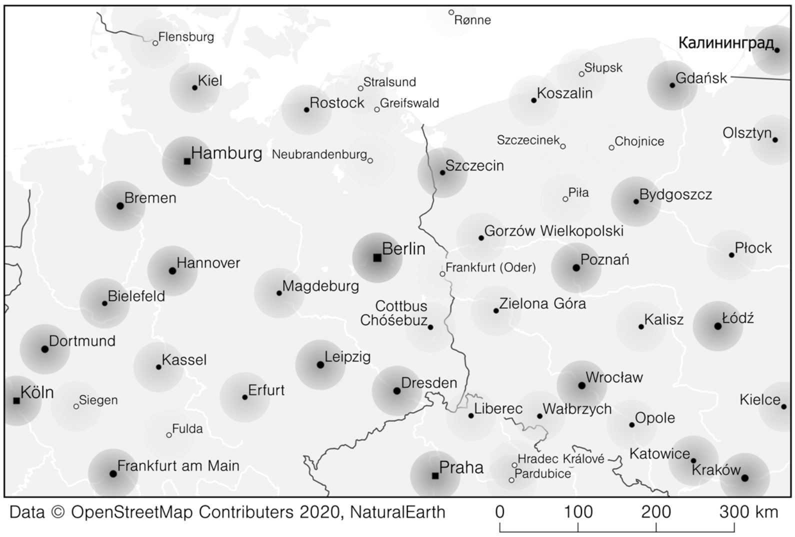


Figure 34: Map example for functional importance. The brightness of the circles represents functional importance based on population values.

A loop over all points has to be applied to determine the highest functional importance value, followed by a second loop that calculates the distance and function values. The result is a quadratic complexity of $O(n^2)$. Figure 34 shows an example map using functional importance for cities with a β-value of 78 kilometers, which is the radius of the function when it nearly reaches zero. The radius is comparable with the tile at zoom level eight, with a size of 156 kilometers in the Web Mercator projection. Through this approach, two cities with nearly the same population as Pardubice and Hradec Králové may get selected.

4.2.3 Discrete isolation approach

The discrete isolation approach transfers the principle of topographic isolation to discrete points. Topographic isolation is the distance to the nearest higher point from a peak, often a point on a slope and not a peak. Discrete isolation considers only peaks; there is no continuous surface, such as a relief, just discrete values from points. Per definition, discrete isolation is the distance from one point to the closest point with a higher attribute value, as Figure 35 illustrates. When computing this value, it is necessary to calculate the distance to all other points with a higher p-value and order the results to get the point with the lowest distance and the next highest value. Using this approach, points (1) to (5) receive a distance, which is storable as an attribute. Point (6) is the highest point; thus, no distance can be derived from points with a higher value. For practical reasons, a default value, such as the earth's perimeter, can be applied.

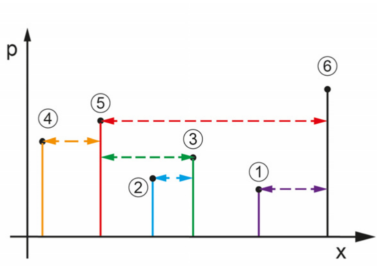


Figure 35: Visualization of the discrete isolation value computation. The points numbered 1 to 6 receive different isolation values because of the p-value. The discrete isolation is the distance (d) to the point with the next highest value for (p).

Implementing discrete isolation requires calculating the distance from a selected point, e.g., (3) to every point with a higher value—(4), (5), and (6)—and then saving the lowest distance to a point with a higher value as point (3). Overall, two loops are needed, as Figure 36 shows: one to go through all the points and a second to compare one point with all the other points, calculating the distance, if necessary. The result is a quadratic complexity of $O(n^2)$.

```
points = (...) # Set of points with numerical attribute
max_isolation = 100000

# Go over all points
for point1 in points:

    # If attribute not null
    if point1.attribute:
        point1.isolation = max_isolation
        # Loop over all points again for comparing the attribute value
        for points2 in points:
            # Compare attribute value
            if point2.attribute > point1.attribute:
                # If attribute value great calculate distance
                isolation = distance(point1, point2)
                # Compare with already calculated isolation, if smaller store it
                if isolation < point1.isolation:
                    point1.isolation = isolation

    # If no attribute aviable, then isolation have to be null
    else:
        point1.isolation = None
```

Figure 36: Python pseudo-code to implement discrete isolation, similar to implementation in the QGIS plugin, "Point selection algorithms."

Figure 37 presents a spatial example of Dresden. Leipzig is the nearest place with a higher population, and the distance between the cities—around 100 kilometers—is the discrete isolation for Dresden. To reach this value, it is necessary to calculate the distance between Dresden and all other cities with more inhabitants, and the minimum distance is stored. A loop over every point, as shown in Figure 36, is needed for all points.

The discrete isolation value can be used later to select points by defining a minimum distance. This technique makes it possible to refine the selection, if required. In the GIS, a graduated renderer can be combined with various classification methods to find suitable isolation values.

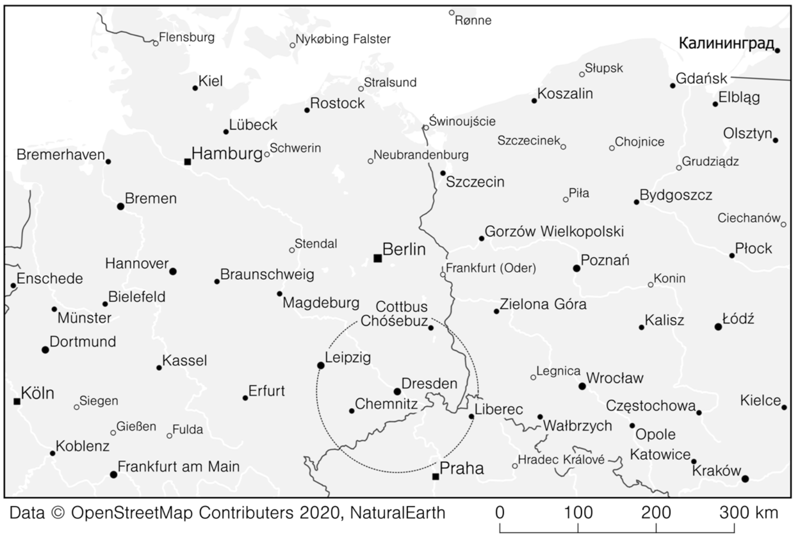


Figure 37: Visualizing discrete isolation for Dresden. Leipzig is the nearest town with a higher population.

As a result, selecting a lower or higher distance value for the isolation increases or decreases the number of points shown on the map. Figure 38 visualizes a selection of cities determined by the isolation method. Only places with a distance greater than 78 kilometers to the next place with a higher population are shown. Visualizing the isolation value is possible by visualizing the circle's radius. The minimum isolation distance is comparable to a circle with a diameter of 156 kilometers, which is comparable to the tile size at zoom level eight.

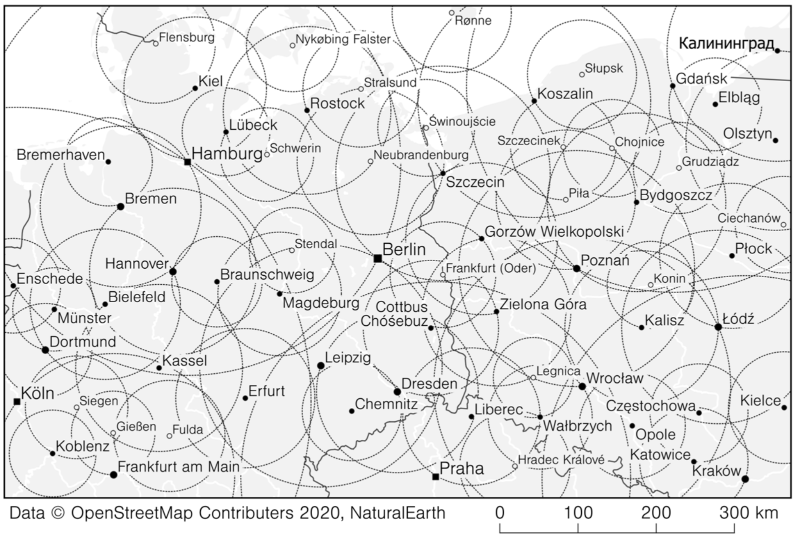


Figure 38: Example map for discrete isolation. The circles show the distance to the nearest city with a higher population.

4.3 Functional evaluation of selection methods

To evaluate the different approaches, a runtime test was conducted to assess the performance of the approaches and to make further statements about possible usage. Additionally, different use cases for discrete isolation are illustrated to show possible applications.

4.3.1 Runtime comparison

To evaluate the runtime of the three described approaches, a series of test runs was conducted with the default settings of the different QGIS plugin tools. Data from randomly distributed points were generated inside a predefined bounding box—100 points were increased to 2,000 points, with 500 and 1,000 points as steps in between. A numerical value attribute between 1 and 6,000 was generated for each point. The selection tool was applied six times to get average runtime values and to remove background effects. A Microsoft Surface with 16GB of RAM, SSD hard drive, Intel Core i7 8650U, and QGIS 3.16.3 was used to run the test.

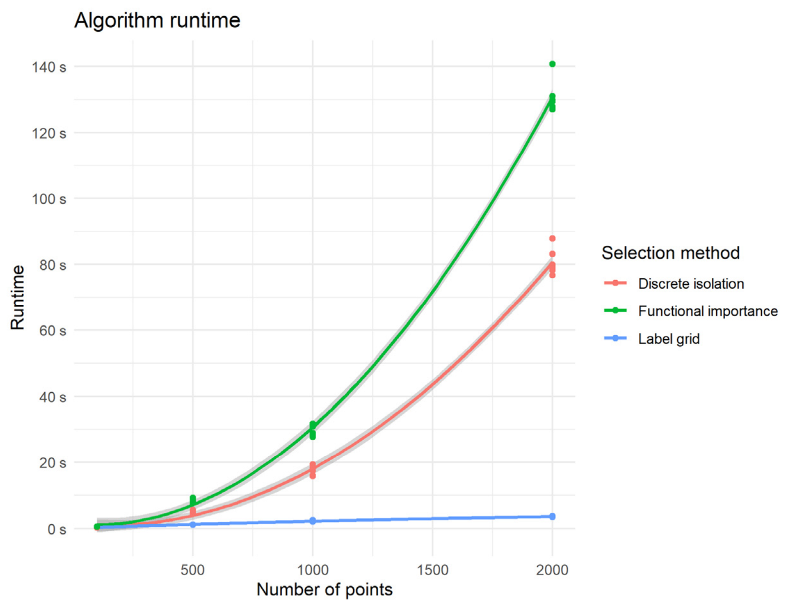


Figure 39: Runtime of the three different selection methods, depending on the number of points.

Figure 39 shows the performance test results—the computing time is nearly identical for 100 points. After a five-fold increase in the number of points, a clear differentiation in runtime and computing time variance becomes visible. At 2,000 points, the label grid needs only 5 seconds, while discrete isolation needs 80 seconds and functional importance 130 seconds. For a better runtime estimation, polynomial curves were worked into the visualization. According to the results, the computation time for the label grid increases somewhat linearly, consistent with the estimated complexity. In contrast, discrete isolation and functional importance show a quadratic runtime development.

The long runtimes of functional importance and discrete isolation increase the number of distances, which must be calculated by increasing points. In the case of discrete isolation, the number of calculated distances is reduced by only considering points that have a higher attribute value. In contrast, the label grid approach runtime depended more on the number of grid cells that were kept constant during the test. Increasing the number of grid cells would also increase the label grid's computation time. In the case of discrete isolation, the parameters did not influence runtime. In the event of unsatisfactory results, it can become necessary to re-run the label grid or functional importance with changed parameters; this is not necessary for the discrete isolation approach.

Data © OpenStreetMap Contributers 2020 and NaturalEarth

Figure 40: Example using discrete isolation for the selection of populated places around Berlin: (a) uses a minimum isolation of about 150 km, (b) uses about 100 km, and (c) uses 100 km for towns and 50 km for cities.

4.3.2 Use cases for discrete isolation

The following examples illustrate how discrete isolation can be utilized in map-making. It is also possible to use for data sets and analysis, such as extracting local hotspots, as the LBSM for Dresden example shows.

4.3.2.1 Selecting populated places

The selection of settlements is a typical map production task, which is essential for the map user's orientation. The following example works with populated places from OpenStreetMap using the provided population number. For the computation, the data was imported into a PostgresSQL database with the PostGIS extension and visualized by QGIS.

Figure 40 shows different selections made with discrete isolation, demonstrating how flexible the measure can be for the generalization refinement operator. In comparison, Figure 40a uses a minimum isolation distance of 150 kilometers, which leads to a sparse distribution of places. If the distance is decreased to 100 kilometers, as in Figure 40b, the map becomes fuller and more places are shown. Figure 40c demonstrates another possibility for using discrete isolation: a combination of different attributes, with the discrete isolation measure refining the selection. In this case, the OpenStreetMap classification of populated places was used, distinguishing between a town,[12] an important urban center, and a city,[13] the largest settlement for a region. Applying different minimum discrete isolation values reproduces a better map than the overall distribution of places. To compare to all existing places, refer to Figure 30.

4.3.2.2 Selecting places from LBSM

Another possible use case is the selection of places from LBSM for analysis or presentation purposes. The outcome is Twitter hotspots around Dresden. This example is purely for demonstration; for better results, LBSM should be combined with topic modeling approaches to split the places into different categories, creating more meaningful results.

[12] https://wiki.openstreetmap.org/w/index.php?title=Tag:place%3Dtown&oldid=1893997
[13] https://wiki.openstreetmap.org/w/index.php?title=Tag:place%3Dcity&oldid=2057875

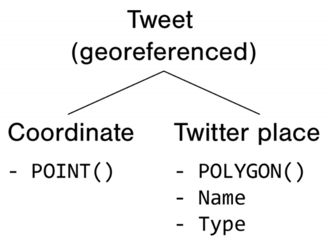


Figure 41: Possibilities for georeferencing a tweet with a coordinate tuple and a Twitter place. Both properties can exist at the same time.

For research purposes, the georeferenced tweets for Europe have been collected. How often a user mentioned Twitter places was summarized for the region around Dresden. Each tweet can contain two possible geolocations, as Figure 41 shows: one location is defined by coordinates, and the other by a predefined place offered by Twitter. Each Twitter place has a name and is classified and described by a rectangle. To simplify processing, the bounding box of the place is reduced to the centroid and the discrete isolation was calculated in QGIS.

With those measures for each Twitter place, it is also possible to derive a locally more critical place according to the number of usages. Table 3 offers a preview of places around Dresden, shown in Figure 42 and ordered by usage count. The map only shows places with discrete isolation greater than 3,000 meters. A closer look at the table shows how many places are hidden despite having a high number, but they are not as essential as in other areas, with many fewer points.

Table 3: Places, mentions count, and isolation around Dresden.

Name	Count	Isolation
Frauenkirche	158	55,470.7
Semperoper Dresden	146	510.2
Stadion Dresden	132	1,311.3
Messe Dresden	132	2,273.0
Dresden Hauptbahnhof	110	1,129.3
Zwinger	101	239.4
Flughafen Dresden	98	6,895.6
Glaserne Manufaktur	79	690.7
Deutsches Hygiene-Museum	78	389.1
Gasthof Bärwalde	73	11,149.3
Sächsischer Landtag	64	225.1
Striezelmarkt Dresden	60	361.0
Großer Garten Dresden	54	836.1
...	...	...

Through discrete isolation, it is possible to identify local hotspots because the values of the nearby places are considered. Indeed, the difference between the values used for ranking can be minimal but still significantly affect the selection result. Another detail that should be mentioned is that places in rural areas are scarce. However, rural areas are locally essential and sometimes exciting, and discrete isolation can handle this effect and provide some results, as Figure 42 visualizes.

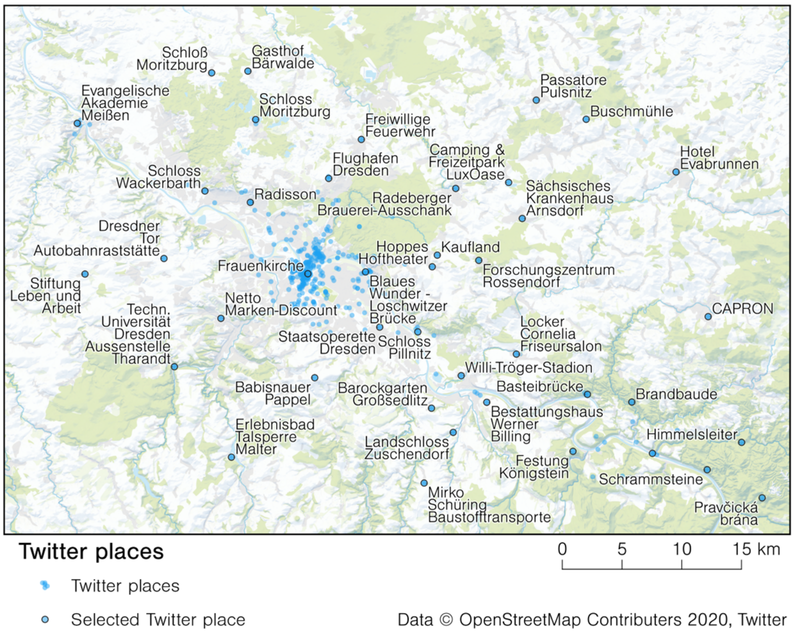


Figure 42: Twitter place selection according to the number of uses.

4.3.2.3 Selecting features depending on the scale

Finally, the discrete isolation selection measure should be demonstrated in a multiscale context using the same data as the "Selecting populated places" section. A new selection should be initiated, starting from one scale to reach a similar information density.

The distance between the point or the example populated places should be increased to select the following scale. A linear distance can be transferred into the scale by applying radical law (Töpfer 1974, 25, 31) as follows:

$$d_T = d_S * \sqrt{\frac{S_T}{S_S}}$$

d_T … distance at target scale $\qquad$ S_T … target scale

d_S … distance at source scale $\qquad$ S_S … source scale

$$d_T = d_S * \frac{S_T}{S_S}$$

Formular for transferring the selection formular, directly proportional to the scale.

A short calculation, starting with the scale at zoom level 7 and 20 kilometers, shows how the minimum isolation distance, according to Töpfer, should be increased in Table 4. This would lead to a map that is overcrowded with points, as in Figure 43d and Figure 43e. Also, counting the number of points shows that there are many more points: in Figure 43d, 49 points and in Figure 43e, 123 points are shown. That is four times the number of the source scale. The usage of the Web Mercator projection might be a reason for this result, as it is not true to length and area. Instead, a proportional formula is used, following leads to a more comparable result in this use case, as Figure 43b and Figure 43c shows. The number of points varies between 18 and 42 but does not change significantly.

Radical law and multiscale mapping

As the examples show, there are problems with applying radical law here. Further investigations should be conducted to evaluate whether this is a specific or general problem.

Table 4: Calculating the minimum isolation distance for the following scale, based on radical law and an adjusted formula.

Zoom level	Scale	Min. isolation distance [km]		Number of points		Figure 43
		Radical Law	Proportional	Radical Law	Proportional	
7	1:4,000,000	20.0	20	31	31	a
6	1:10,000,000	31.6	50	49	42	d, b
5	1:15,000,000	38.7	75	123	18	e, c

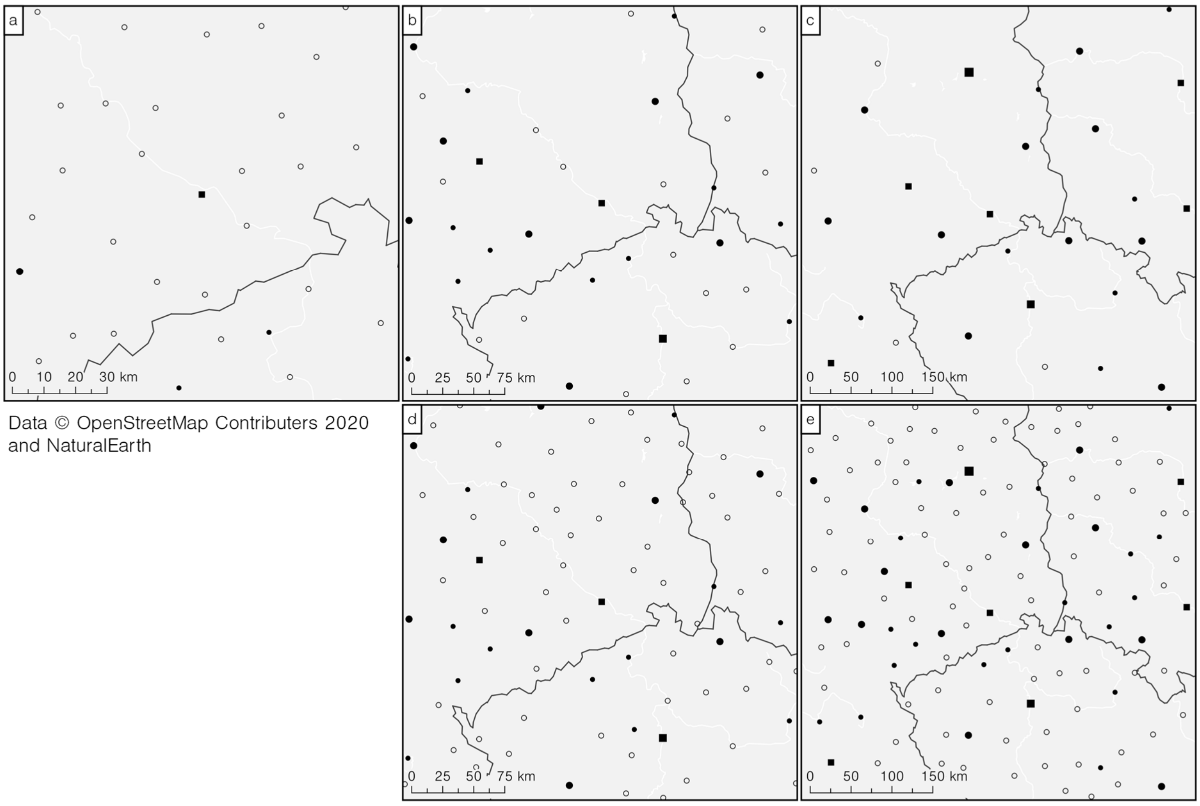


Figure 43: Example for using discrete isolation for the selection of populated places around Dresden on different scales: (a) uses a minimum isolation about 20 km, (b) uses about 50 km, (c) uses about 75 km, (d) uses about 31.6 km, and (e) uses about 38.7 km in the Web Mercator projection. No labels are shown to focus on the selection result. For the derivation of (d) and (e), radical law was applied, while (b) and (c) use a proportional approach.

The resulting selection in Figure 43b and Figure 43c, with the simplified formula, would enable labeling the points because there is enough space. Furthermore, it has a similar information density. Nevertheless, more research needs to be done on how selection measures can be transferred from one zoom level to another, taking into account the unique properties of Web Mercator projection. The given example is limited to transferring the formula to more general problems.

4.4 Discussion of the selection approaches

A critical review of the presented selection methods and demonstrated application possibilities follow now. After discussing the applications, the different approaches will be compared according to their properties and application possibilities for further usage.

4.4.1 A critical view of the use cases

The discrete isolation measure seems to be useable for visualization selection purposes. It can deal with the classic cartographic task of selecting settlements and new LBSM data on different scales. An advantage is that once computed, the isolation values can be used several times and for several scales. Refinement can be done with other attributes, as in the use case with populated places. Additionally, the degree of generalization abstraction and the information load can be adjusted with discrete isolation. This circumstance can also solve labeling problems with the point data, and a minimum distance between the points can be ensured.

Unfavorable is the high complexity of discrete isolation and the long computation time that undoubtedly results from it. It is possible to reduce this a bit by only considering points within a defined radius, such as 500 kilometers, because points further away have no local effects. Nevertheless, it is still a disadvantage, which can make long computations necessary when updating a database. Here, the label grid approach would be very beneficial because only the points in a grid cell need to be considered.

However, the label grid method needs to be re-computed for each scale, zoom level, or change in the parameters. This is not very handy for a multiscale environment, such as "Selecting features depending on the scale" or "Selecting places from LBSM". Here, discrete isolation allows a flexible solution by changing just the value for selection, which can be helpful for finding a degree of abstraction or for identifying relevant places.

Depending on the map scale, the selection has similar demands that discrete isolation can fulfill. Nevertheless, keeping the selection consistent over many scales seems challenging. Traditional radical law failed in the use case, but there can be many reasons for this. Of course, the use approach with a proportional transfer of linear measurements can be generalized, but it should be tested more in multiscale setups and evaluated by an empirical study. Problems with transferring measures from one zoom level to another often happen during the production of web maps. A design must work at multiple zoom levels and at a single level, requiring a system that can ensure the consistency of the map. A selection measure can be an example, but symbol sizes or colors are also relevant. There is still room for cartographic research on this topic.

4.4.2 Comparing the approaches

The following section provides an overview of the described and applied methods for selecting POIs with a numerical attribute. Table 5 compares the described selection methods according to performance, efficiency, and usability in multiscale mapping.

The estimated complexity and the performance test with the QGIS plugin showed that the label grid approach has the best performance. The python implementation and the PostGIS function can handle large-point data sets very well. This is one reason for their use in web mapping production, but it comes with a disadvantage—selection ranking is based on a grid cell, which depends on one zoom level or scale.

The complexity of discrete isolation is slightly lower than functional importance, which was also proven by the runtime comparison. Figure 39 shows an average runtime of 1,000 points, which is significantly higher for those two approaches. The difference between the two approaches is that discrete isolation only considers points with higher values and not all points, like functional importance. In contrast to the label grid, a selection refinement is still possible because the numerical values are already calculated and used for the final selection.

The grid for the label grid is adjusted for one scale, as well as the functional importance formula. This results in re-computing the selection for every scale or changing the parameter again. Discrete isolation is only one calculation performed once and afterward, the value for selecting the points can be adjusted without further processing. That is a strong advantage of discrete isolation over other approaches, which require a re-run to adapt the selection to a new scale.

The label grid parameters are the grid shape (square, hexagon, diamond) and the grid cell size. In contrast, discrete isolation offers only the resulting isolation distance as a criterion for selection. The functional importance is entirely flexible—the formula can be changed completely; in our case, radius β is the parameter and the calculated importance values. The label grid's disadvantage is that combining a selected point or a ranking with other attributes is difficult. For example, it is impossible to consider administrative capital in the ranking process, but it can be shown with other points ranked by their numerical attribute. Also, a ranking inside one grid cell does not consider spatial distribution and may cause later labeling problems. Functional importance and discrete isolation, with their value-based selection, offer the possibility of considering spatial distribution and other attributes independent from a grid.

Table 5: Comparison of selection method performance and usability: label grid, functional importance, and discrete isolation.

	Label grid	Functional importance	Discrete isolation
Complexity	$O(n * \log n)$	$O(n^2)$	$O(n^2)$
Average run time for 1,000 points	2 s	29 s	17 s
Re-run with new parameter useful	Yes	No	Yes
Re-run for different zoom levels or scales	Yes	Yes	No
Parameters	Gird width, grid shape	Formula and their variables, function values	Isolation distance
Depending on the map projection	Yes, if a grid is used	No	No
Combined with other attributes	Limited	Yes	Yes
Implementations	QGIS plugin, PostgreSQL/PostGIS	QGIS plugin	QGIS plugin, PostgreSQL/PostGIS
Usage	Mapbox, TopPlusOpen, OpenMapTiles	Imagico.de	meinGrün

For the label grid approach, the grid shape and size depending on the map projection and the influence of the resulting selection; the other methods can be used depending on—or independently from—the map projection, utilizing ellipsoidal distances. According to a map projection's project settings and characteristics, one solution might be advantageous.

The label grid was already implemented for PostgreSQL/PostGIS, and is now available for QGIS. An implementation of discrete isolation also exists for both environments; the functional importance is only available in the QGIS plugin. The application of the three different methods can be reviewed through different web services. Only some examples are mentioned for the label grid, although more exist. Functional importance seems to be in use by the originator and is less known about it; discrete isolation is already in use at the meinGrün project.[14]

4.4.3 Conclusion

Discrete isolation

Discrete isolation delivers good results for analysis, and for visualization purposes if an even spatial selection is needed or critical local points are selected.

Overall, the label grid is the best choice if performance matters. It can also be used with regular or non-regular polygons. Discrete isolation is versatile and delivers good results but needs more computation time. Both methods work well in a multiscale environment and seem suitable for efficient map production. More complicated is functional importance—the results are good, but the computation time is long and the modeling is complex. An application in multiscale maps would require a long pre-computation for each scale, making the method unsuitable.

All of these methods generally combine ranking features according to a numerical attribute with a spatial component that considers the surrounding points and their attributes. This ensures that the most critical points are visible, followed by less critical points. The approach is transferable to data sources other than VGI and LBSM, such as measuring air pollution levels or selecting elevation or depth points. In addition, labeling conflicts can be prevented because all of those techniques can be utilized to ensure a minimum distance between selected features.

[14] https://app.meingruen.org/#/

5 Creating multiscale maps

The following chapter explains the production of multiscale maps using VGI by analyzing two broadly used open source examples. Section 2.5, "Technologies for serving multiscale maps," provides the theoretical foundation. The workflow of raster tile-based web maps with by-the-minute updates is the OpenStreetMap Carto[15] project, and the vector tile-based project with daily updates is the OpenMapTiles[16] project. Both projects are well-known, are prototype-like, use the same data sources, are available on GitHub, and can be set up easily using Docker, which makes them easy to use. More maintained open map styles are based on OpenStreetMap, such as the German[17] and French[18] variants of OpenStreetMap Carto or OpenTopoMap.[19] However, the two projects do not address a similarly broad audience. A published example that uses OpenStreetMap and official data is the creation of the TopPlusOpen[20] service (Kunz and Bobrich 2019).

The discussion follows an abstraction of the general workflow for multiscale map production. The different intermediate stages are explored, including various requirements and software solutions. In addition, the uprising stackless approach for map tile creation is documented. Finally, this thesis applies the technological approach to visualization and generalization techniques: micro diagrams and discrete isolation.

5.1 Examples of multiscale map production

To describe the production of web maps on multiple scales, OpenStreetMap Carto and OpenMapTiles were chosen. Both projects have existed for several years, serving different purposes with their techniques as raster and vector tile services. In addition to the workflow, a summary of the crucial cartographic topic generlization in this context is provided.

The default tile layer on the OpenStreetMap homepage (full name OpenStreetMap Carto or "OSM Carto" or "osm-carto") has been in use since 2013 (OpenStreetMap Wiki 2022c). Nevertheless, the map style is older, and the "Carto" in the name references the conversion software of the map style written in CartoCSS to an XML file, readable for the mapnik rendering software (Hormann 2018).

MapTiler Team currently maintains OpenMapTiles, which is based on the OSM2Vector Tiles project, created from a bachelor's thesis (Martinelli and Roth 2016). While it is a free project, there are still possibilities to set up self-hosted servers or to rent services from MapTiler with some additions,

CartoCSS

CartoCSS is a language that easily defines a map style for mapnik rendering software, similar to CSS. It exposes different functions as filters and styling from mapnik XML-style files but is much easier to handle. It is converted by the software Carto to an XML readable for mapnik.

Mapbox GL

Mapbox GL is a collection of tools for creating responsive, client-side maps in web, mobile, and desktop applications. The abbreviation "GL" stands for OpenGL, the industry-standard Open Graphics Library.

---

[15] https://github.com/gravitystorm/openstreetmap-carto
[16] https://github.com/openmaptiles/openmaptiles
[17] https://github.com/giggls/openstreetmap-carto-de
[18] https://github.com/cquest/osmfr-cartocss
[19] https://github.com/der-stefan/OpenTopoMap
[20] https://www.bkg.bund.de/SharedDocs/Produktinformationen/BKG/DE/P-2018/180608-TopPlusOpen.html

e.g., hill shading, contour lines, terrain model, or satellite imagery (OpenStreetMap Wiki 2022b). There are projects that are similar to OpenStreetMap Carto that can adjust a map's style, e.g., the OpenStreetMap Americana.[21] By using vector tile technology, it is possible to create a new style or tileset with just a few changes. The OpenMapTiles project is a collection of tools for creating vector tile "openmaptiles-tools"[22] and a configuration for the vector tile schema "openmaptiles."[23] There are three suitable styles for the created tiles called "osm-bright-gl-style," "maptiler-basic-gl-style," and "positron-gl-style." The "gl" stands for accelerated graphic rendering, similar to the Carto software in the OpenStreetMap Carto style.

5.1.1 OpenStreetMap Infrastructure

Since both mapping projects use OpenStreetMap as a primary data source, it is described before each workflow explanation. OpenStreetMap saves its fully versioned data in a PostgreSQL database. Through an API, it is possible to retrieve and update that database. Due to the need for a fast response time, the API request size is limited and mainly used to edit data with an OpenStreetMap editor (Ramm and Topf 2011). For accessing larger chunks of data, a dump produced by the Osmosis program is used ("Osmosis – OpenStreetMap Wiki" 2022), which stores all information for one point in time in a binary exchange format for the entire world called "planet.osm." This is done weekly and is available for download to every registered OpenStreetMap user ("Planet.Osm – OpenStreetMap Wiki" 2022). Various services offer different regional extracts as free downloads without requiring contributor information. The Geofabrik,[24] for example, cuts the data according to administrative boundaries into pieces daily and then converts it into various formats, such as the binary OSM-PBF-format, the OSM-XML-format, or for small enough regions as a GIS-ready Shapefile.

As shown in Figure 44 and Figure 51, those dumps are only one part of the database's exports; "diff" files are created by the minute, hour, and day with the changes from that particular time interval ("Planet.Osm/Diffs – OpenStreetMap Wiki" 2021). This data extract allows users to update their database. For the implementation workflow, the derived database design must consider that requirement and a tool such as osm2pgsql[25] or IMPOSM[26] to handle this behavior. In addition, it is possible to cut the diff file for a specific region or filter for changes by user, time, or tags. Possible tools for processing the raw OpenStreetMap data are Osmosis or osmconvert, to name a couple.

---

[21] https://github.com/ZeLonewolf/openstreetmap-americana/

[22] https://github.com/openmaptiles/openmaptiles-tools

[23] https://github.com/openmaptiles/openmaptiles

[24] https://download.geofabrik.de/

[25] https://osm2pgsql.org/

[26] https://imposm.org/

A particular dataset is OpenStreetMap's coastline. Due to its nature as a colossal polygon, which reaches continent size, a specialized workflow is used. In order to provide a valid coastline, it is necessary to run a separate daily process, just in case faulty edits break the coastline; a broken coastline has to be fixed manually, which can take several days ("Coastline Processing" 2022; "Coastline – OpenStreetMap Wiki" 2022). A website[27] offers the latest valid coastline, the inverted water areas, and the Antarctic ice shield as Shapefiles for download for all mapping projects.

5.1.2 OpenStreetMap Carto

It is volunteers who develop OpenStreetMap Carto using GitHub as a collaboration platform. This allows us to explore the technology's details but does not explain the context or the relationship between the tool and the reasoning behind some actions. This section aims to describe and visualize what needs to be done to create a map from OpenStreetMap data, and how minutely updates work.

5.1.2.1 Data source and data storage

Osm2pgsql imports the data into the PostgreSQL rendering database with PostGIS. The tool's task is schema transformation, making the key-value-tagged data available in a relational database. This is done by generating four tables named "planet_osm_line," "planet_osm_point," "planet_osm_polygon," and "planet_osm_roads" with the most common tags in a column, resulting in huge tables with many empty cells. Additional key-value combinations are stored using the hstore-extension in a column that is accessible by separate operators.

PostgreSQL/PostGIS

PostGIS is an open source spatial extension of the free relational database management system PostgreSQL, using Simple Features for SQL specification. It adds support for geographic data types and functions, allowing location queries and analyses to be run in SQL.

Afterward, indexes and clustering is created according to the geometry to allow minimal access time for the data. With the help of a lua-script[28] cleanup, operations are performed during the import, such as how to treat the geometry of some tags, deleting unnecessary tags, and calculating the rendering order of road features. The import of diff and summaries of what has changed are handled the same way, and specific tiles are added to the tile-expire-file. This gives the renderer the information needed to update the tile. ("Osm2pgsql Manual - Osm2pgsql" n.d.; ircama 2018).

Additional data is used for rendering lower zoom levels: borders from Natural Earth[29] and the coastline. Those datasets are stored as shapefiles in a data directory. In the case of the coastline, it makes sense to update those datasets from time to time.

---

[27] https://osmdata.openstreetmap.de/data/

[28] https://github.com/gravitystorm/openstreetmap-carto/blob/master/openstreetmap-carto.lua

[29] https://www.naturalearthdata.com/

5.1.2.2 Tile and style creation

The main tool for producing raster images from geodata is the mapnik renderer, which is configured by an XML file that contains rules on how to visualize the data request using SQL queries from the database or the shapefiles, according to the requested scale. Due to the complexity of handling big XML files, the map style is created in a styling language called CartoCSS, which is converted by the Carto tool in the mapnik configuration file. Splitting the styling into several files (one per layer) is possible, in order to maintain the overview while everything in the project is collected.

As a design tool, Kosmetik is used, which offers a text editor next to the preview map, making it possible to check style changes on the map. This gets rendered after the required conversion of the style for mapnik and the rendering of the tiles.

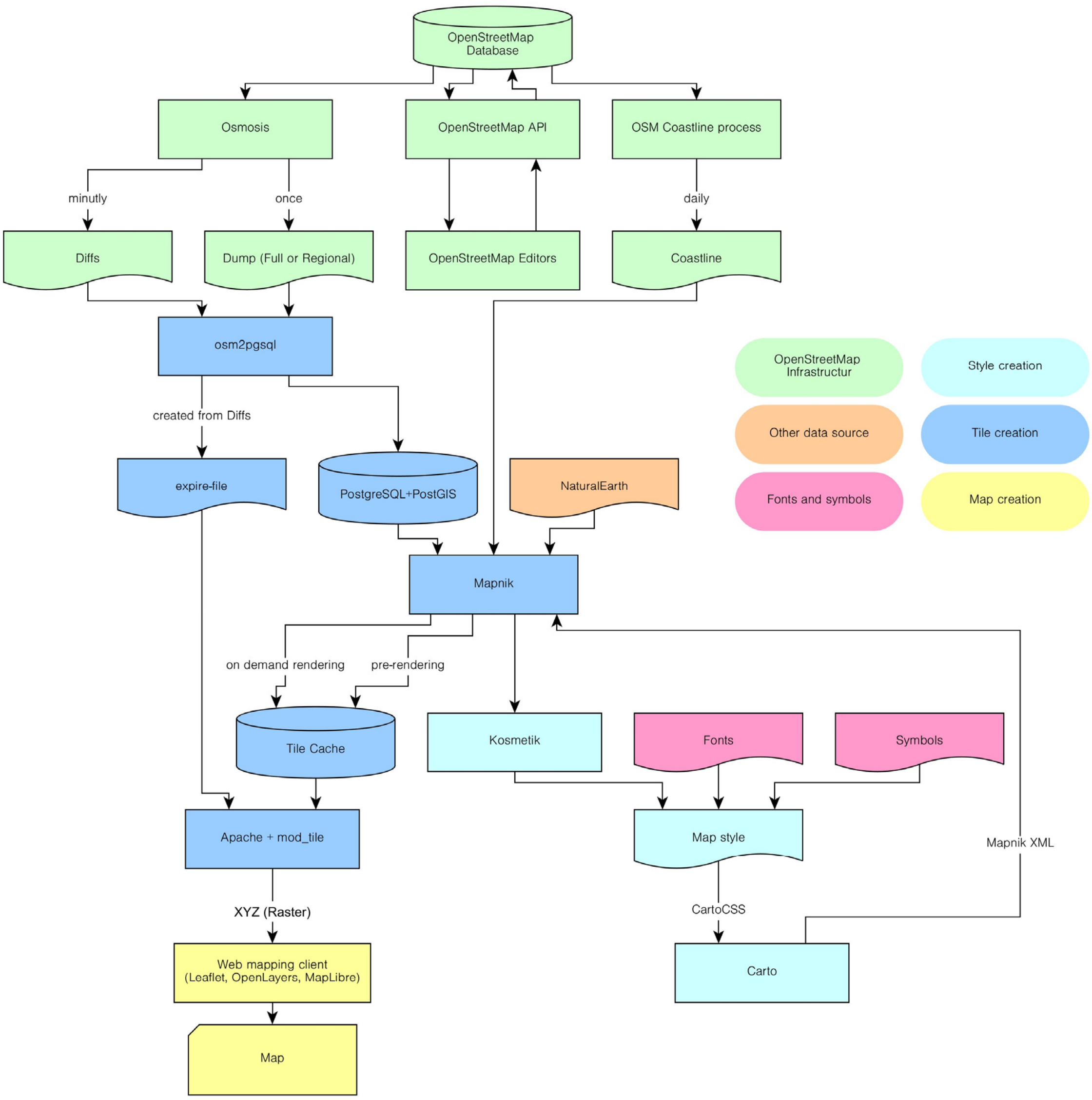


Figure 44: Visualization of the data flow to produce a multiscale map with OpenStreetMap Carto.

The generalization is produced by selecting, ordering, or aggregating features according to their attributes or size on the fly through SQL queries. Other generalization effects are achieved by using graphic effects that visually merge features in combination with a dedicated rendering order, as in Figure 45 and Figure 46. In addition, for low zoom levels, data from Natural Earth is used for the boundaries.

Icons for patterns or symbols are vector or raster images directly accessed by the mapnik renderer. The font of the map style is Noto Sans, which covers nearly all worldwide languages and alphabets. In addition, there are scripts that create road shields and coloring schemas for highways. The intercultural map style now follows a typical color schema and uses an automatically created and optimized color ramp to distinguish between different types of roads.

mapnik

mapnik is an open source mapping software for map rendering. An XML is used to reference and style the mapped objects. Possible data sources are usual vector and raster data formats.

5.1.2.3 Serving map tiles

The Apache Webserver mod_tile is a crucial feature for serving the tiles. It controls which tiles are pre-rendered and which must be rendered by managing the request and the cache. Lower zoom levels are frequently in use and always in the cache. Higher zoom levels are rendered on demand. If a user zooms in, the tiles are requested from the server. If the mod_tile notices that a particular tile is not in the cache, it requests the tile's rendering. The rendering process works like a queue: It contains tiles that must be prerendered by definition, tiles that the user requests, and tiles that need to be processed because of new data in the database. The source for this information is tile-expire-file.

There are still other techniques in use as well. Due to the potentially high memory usage (Ramm and Topf 2011) of the raster tiles, it is also necessary to delete tiles that are not requested. Another approach is dedicating similar tiles, such as the blank tile in oceans, which lacks supplementary information. In addition, technical restrictions, such as limited storage capacity, the limited number of file names on a device, and labeling issues, are resolved by the principle of meta tiles ("Meta Tiles – OpenStreetMap Wiki" 2015). A meta tile is an oversized tile that aggregates smaller tiles; it is cut into smaller tiles for delivery to the client.

5.1.2.4 Viewing map tiles

Finally, map tiles are visually merged by a viewer. In the case of a web page, a JavaScript-based web mapping client performs the task. Map tiles within the extent, according to the current extent and scale, are requested and shown. A GIS uses the same approach but may offer the user the possibility of using more parameters, such as authentication, attribution, and image resolution, as shown in Figure 50. Nevertheless, the client uses a cache to save time requesting and loading tiles.

5.1.2.5 Generalization

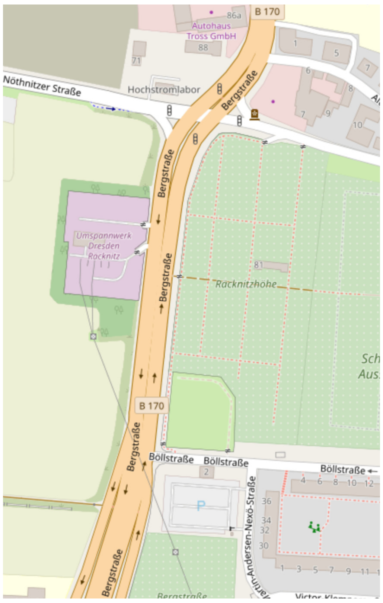


Figure 45: Example of merging lines from OpenStreetMap. (Visualization OpenStreetMap Contributors CC BY-SA 2.0 2021)

Figure 46: Example of ordered labeling from OpenStreetMap. (Visualization OpenStreetMap Contributors CC BY-SA 2.0 2021)

The workflow usually only contains the generalization operator selection that omits, simplifies, and combines the data sources (Kraak and Ormeling 2010; Slocum et al. 2009). Due to the use of Natural Earth for lower zoom levels and OpenStreetMap for higher levels, two data sources are combined; Natural Earth is generalized for a defined scale. A gap in between is filled by selecting features through SQL queries on the fly and their visualization.

Zoom-level adapted SQL commands exist in the project file, separated from the styling for each map layer. The queries collect the features for the visualization across three to five zoom levels, as Figure 47 shows for the land cover for zoom levels 5 to 9. A separate part follows styling, as Figure 49 demonstrates. In generalization theory, this is a separation into model generalization (queries) and cartographic generalization (styling) (Burghardt, Duchêne, and Mackaness 2014). The SQL is generated on the fly in a cartographic landscape mode suitable for limited rendering to the requested bounding box of the renderer. This procedure is similar to the star model of digital landscape model derivation (Kraak and Ormeling 2010, 99). In general, the "CASE"-statement combines feature classes, while the "WHERE"-clause applies filtering and the selection omits features. The selection works with the length and area of the feature in pixels, not in the natural area, by using a set of variables replaced at runtime through mapnik in the query:

```
!bbox!: Bounding box of tile
!zoom!: Zoom level of tile request
!scale_denominator!: Map scale of tile request
!pixel_width!: Width of a pixel in grid units
!pixel_height!: Height of a pixel in grid units
Note: The parameters are specific to the mapnik rendering,
but similar ones are used in other software.[30]
```

The bounding box variable is always part of the query and limits the result to requesting map extent, which is usually a meta tile. It works as a dataset partition during processing, and only the necessary data is processed. Later the result is cropped to the final smaller tile extent. Due to this approach, the result looks seamless in the end, but is processed in chunks.

Another notable feature is ordering. In Figure 47, "ORDER BY" enforces the rendering of more essential features before smaller ones to avoid overlapping labels. The labeling in Figure 46 is in descending order according to population number. Thus, in this method, the city of Nürnberg is labeled first, leaving no space for the name of the nearby city of Fürth. This principle of ordered plotting by attributes ensures a ranking that labels the more critical places (Allan [2012] 2019).

[30] https://github.com/mapnik/mapnik/wiki/PostGIS

```
Layer:
  - id: landcover-low-zoom
    geometry: polygon
    <<: *extents
    Datasource:
      <<: *osm2pgsql
      table: |-
        (SELECT
            way, way_pixels,
            COALESCE(wetland, landuse, "natural") AS feature
          FROM (SELECT
              way,
              ('landuse_' || (CASE WHEN landuse IN ('forest', 'farmland', 'residential', 'commercial', 'retail', 'industrial',
                                                    'meadow', 'grass', 'village_green', 'vineyard', 'orchard') THEN landuse END)) AS landuse,
              ('natural_' || (CASE WHEN "natural" IN ('wood', 'sand', 'scree', 'shingle', 'bare_rock', 'heath', 'grassland', 'scrub') THEN "natural" END)) 
              ('wetland_' || (CASE WHEN "natural" IN ('wetland', 'mud') THEN (CASE WHEN "natural" IN ('mud') THEN "natural" ELSE tags->'wetland' END) END))
              way_area/NULLIF(POW(!scale_denominator!*0.001*0.28,2),0) AS way_pixels,
              way_area
            FROM planet_osm_polygon
            WHERE (landuse IN ('forest', 'farmland', 'residential', 'commercial', 'retail', 'industrial', 'meadow', 'grass', 'village_green', 'vineyard', '
              OR "natural" IN ('wood', 'wetland', 'mud', 'sand', 'scree', 'shingle', 'bare_rock', 'heath', 'grassland', 'scrub'))
              AND way_area > 0.01*!pixel_width!::real*!pixel_height!::real
              AND building IS NULL
          ) AS features
          ORDER BY way_area DESC, feature
        ) AS landcover_low_zoom
    properties:
      cache-features: true
      minzoom: 5
      maxzoom: 9
```

Figure 47: Excerpt from project.mml showing the SQL query for land cover visualization in OpenStreetMap Carto.

Similarly, the highways are visualized to hide tunnels under all above-ground features or merge lines through overlaying. In OpenStreetMap, contributors digitize direction lanes separately. The lanes should be merged at lower zoom levels because the distance gets too small. Figure 49 shows that merging happens visually because the darker orange chasing is drawn first; afterward, the lighter orange is used for filling. Finally, if the distance is small enough, the lines visually merge into one line (Allan [2012] 2019; Hormann 2018).

Adding details on demand or the ability to zoom in is a principle of OpenStreetMapCarto for handling POIs because people are more motivated by features they can find on the map. Essential points are already shown at lower zoom levels with a suitable icon, while colored points indicate more POIs, as in Figure 48. This way, information is already indicated. The user can explore the details on demand by zooming in and getting the specific symbols and names, if there is enough space for labeling.

A geometry simplification is not directly part of the map style and is outsourced by selecting already simplified datasets, such as Natural Earth, and a simplified version of the coastline. In the case of Natural Earth, the generalization is made manually, while the coastline process is fully automatized and freely available on GitHub. [31]

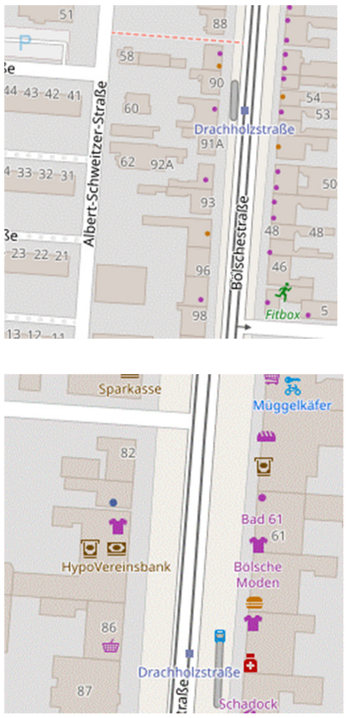


Figure 48: Example of details on demand—the point getting symbols in the next zoom level. For the remaining colored POIs, there are no symbols in the map style. Upper shows tile zoom level 17 and lower zoom level 18. (Visualization OpenStreetMap Contributors CC BY-SA 2.0 2022)

[31] https://github.com/osmcode/osmcoastline

```
[feature = 'wetland_swamp'][zoom >= 5] {
  polygon-fill: @forest;
  [way_pixels >= 4]  { polygon-gamma: 0.75; }
  [way_pixels >= 64] { polygon-gamma: 0.3;  }
}

[feature = 'wetland_mangrove'][zoom >= 5] {
  polygon-fill: @scrub;
  [way_pixels >= 4]  { polygon-gamma: 0.75; }
  [way_pixels >= 64] { polygon-gamma: 0.3;  }
}

[feature = 'wetland_reedbed'][zoom >= 5] {
  polygon-fill: @grass;
  [way_pixels >= 4]  { polygon-gamma: 0.75; }
  [way_pixels >= 64] { polygon-gamma: 0.3;  }
}

[feature = 'wetland_bog'],
[feature = 'wetland_string_bog'] {
  [zoom >= 5] {
    polygon-fill: @heath;
    [way_pixels >= 4]  { polygon-gamma: 0.75; }
    [way_pixels >= 64] { polygon-gamma: 0.3;  }
  }
}
```

Figure 49: Styling of the land cover. Text starting with "@" references a color defined at the beginning of the file landcover.mss.

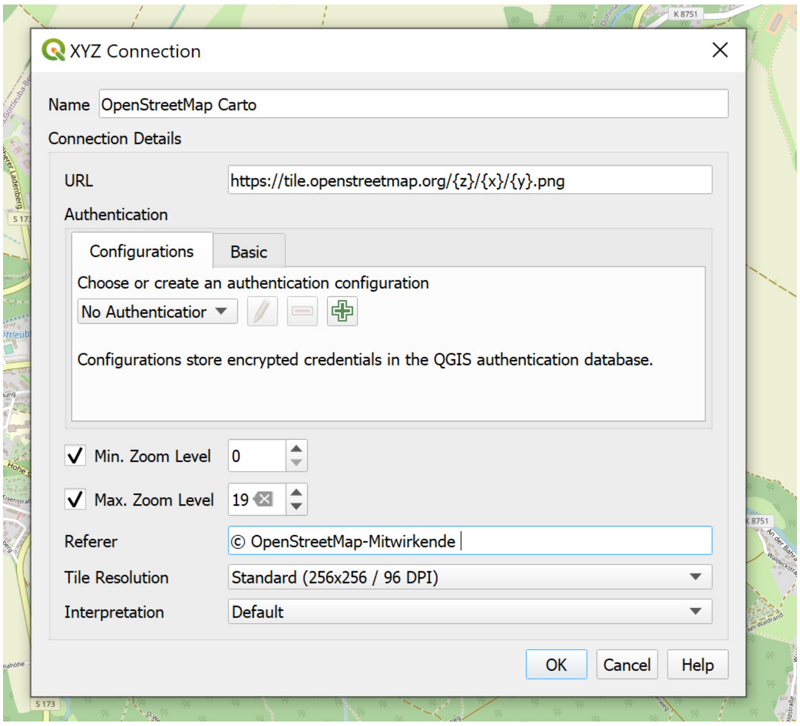


Figure 50: Configuration possibilities for an XYZ layer in QGIS.

5.1.3 OpenMapTiles

All OpenMapTiles code is open source and accessible via GitHub. The code consists of two parts: a collection of tools for creating the vector tiles, and a set of layer configurations that complete the specific tiles. It is possible to fork and complete the layer according to individual needs. Figure 51 offers an overview of the complete workflow with the tools and intermediate formats for the creation of the vector tiles.

5.1.3.1 Data sources and storage

All different data sources are imported into a PostgreSQL database with PostGIS tools to generate the tiles. For the non-OpenStreetMap data, tools based on GDAL are used, while IMPOSM handles the OpenStreetMap import. According to the configuration file, the software transforms the key-value-based data into standard geodata suitable for a relational database. This allows defining which tags should be considered in a particular geometry, collected in tables with manually defined columns. There are also functionalities for deriving more generalized layers based on imported data and functions, which handle the z-ordering of roads. The software clusters and indexes the data for faster access. While importing data into the database, a directory is created for caching, which also works with the diff files to identify which data need to be updated. Updates through the diffs-affected tiles are collected in a separate directory.

Wikidata

Wikidata is a collaboratively created and multilingual knowledge graph that is part of the Wikipedia project, available under the CC0 public domain license.

Data from the Natural Earth[32] project is used at lower zoom levels to show places and lakes along with coastline data from OpenStreetMap. As an additional data source for the OpenMapTiles project, similar to Natural Earth, it serves as a collection of center lines for lakes—, “osm-lakelines”[33], which are precalculated for labeling them correctly.

Wikidata[34] is an additional source for naming places. Within OpenStreetMap, data is often Wikidata ID saved. By joining both sources, it is possible to offer a broader range of names for cities. That is, so far, a point because the technology of the vector tiles allows changing labels on-demand already the names are present in the tiles, as Figure 54 demonstrates. It is a kind of data enrichment in map production and generalization (Stanislawski et al. 2014, 166) for labeling and selecting cities by the label grid method, as explained in section 4.2.1.

5.1.3.2 Tile and style creation

The creation of the tiles is PostgreSQL in combination with PostGIS in use. While the data for a specific layer and zoom level is collected in materialized SQL, the export follows in a separate step (Martinelli and Roth 2016). The tiles are exported in parallel processes from the different views for all layers by making a “UNION” and saving the result by the ST_AsMVT[35] function. It encodes it and saves the result zipped on the disc. Afterward, the tiles are

MBTiles

MBTiles is a file format for storing map tiles in a single file. Technically, it is an SQLite database, which serves as a container. It stores the tile with the information of a TilesJSON, and it can contain raster and vector tile sets.

[32] https://www.naturalearthdata.com/
[33] https://github.com/lukasmartinelli/osm-lakelines
[34] https://www.wikidata.org/wiki/Wikidata:Main_Page
[35] https://postgis.net/docs/ST_AsMVT.html

stored with the tileJSON in an MBTiles container to avoid the problem of handling many small files and having too few possible filenames on a disk. Updates of the tile are made daily according to the content of the tile expire directory, and the recent version replaces the tile.

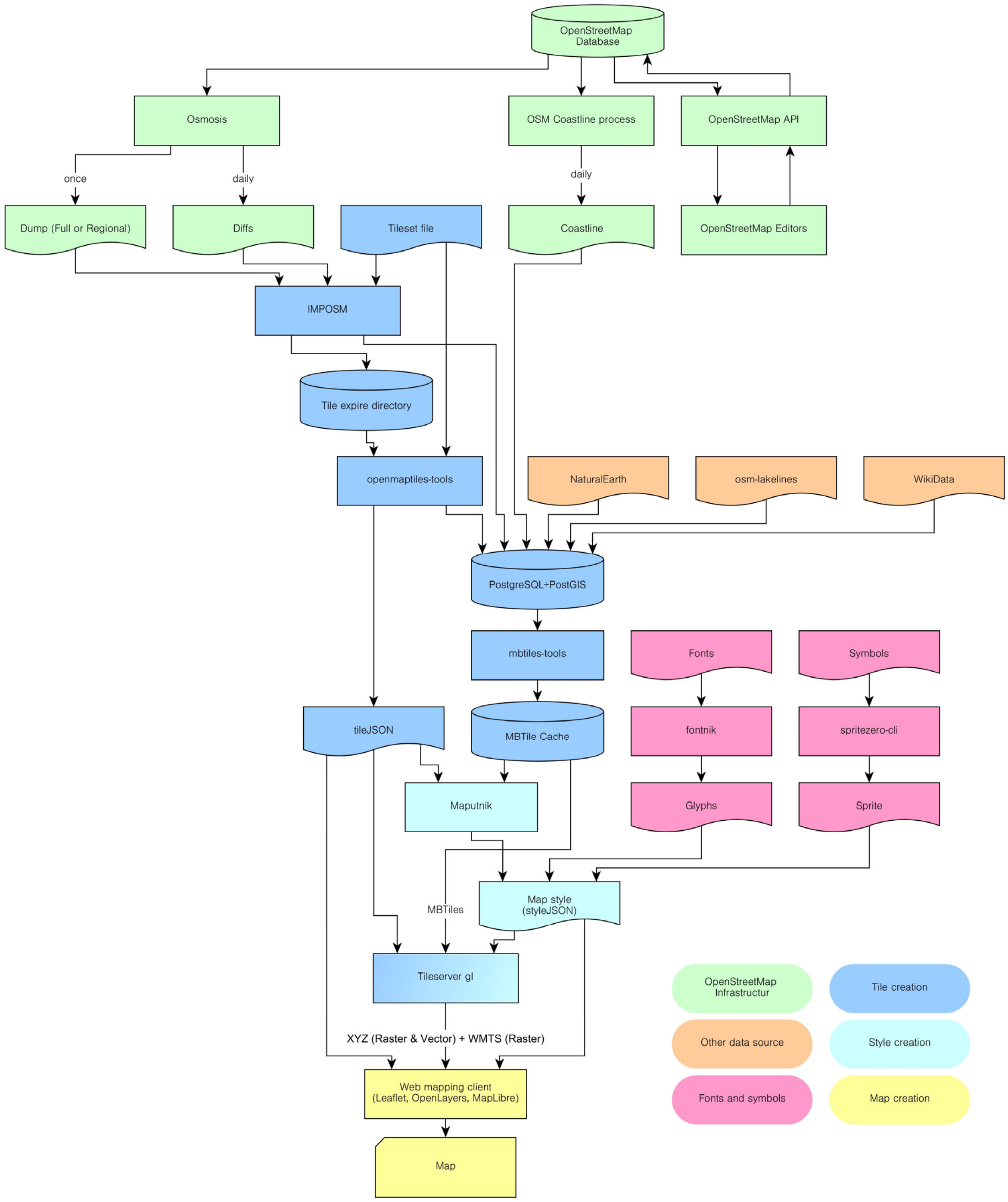


Figure 51: Workflow for creating a multiscale map with the OpenMapTiles toolset and all necessary additions.

The creation of a style is split due to the intermediate nature format of the vector tiles into two steps. It starts by determining a layer's contents by creating a view in the database, which results in a layer in the vector tiles. It has to be done first and influences the later styling by offering attributes and processing the geometry. Styling the tiles is done with the help of the Maputnik[36] editor in the Mapbox GL language. It collects the tileset's information, offers various preview options, and directly shows changes in the map style. It is possible to use the graphic interface, but some features need to be directly done in a text editor. The fonts and symbols are included in their vector-tile-specific formats called Glyphs and Sprites, which are explained under “Mapbox Vector Tiles” section.

5.1.3.3 Serving map tiles

The Tileserver gl accesses the cached tiles for delivering them and sends them as a vector tile to the user. Part of the cache is also the tileJSON file describing the tileset with its extent, zoom levels, layers, and attributes. In addition, it is possible that the server renders the vector tiles with map style to raster tiles and delivers them to the mapping client. There is the flexibility to deal with clients for which vector tiles cannot render or use the WMTS standard for serving the tiles allowing more flexibility.

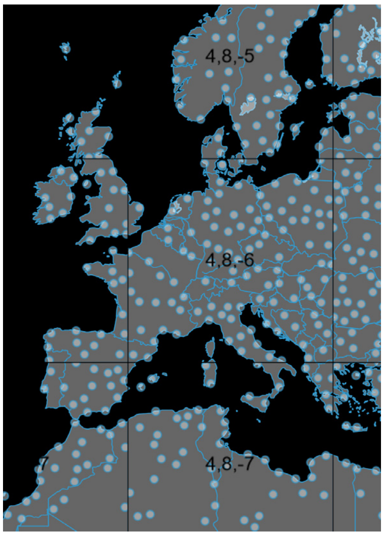


Figure 52: Vector tile examples, with tile numbering and data using the t-rex software viewer.

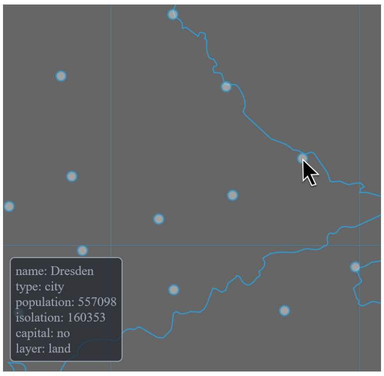


Figure 53: Vector tile examples showing the data in a general style and providing information about feature attributes using the t-rex software viewer.

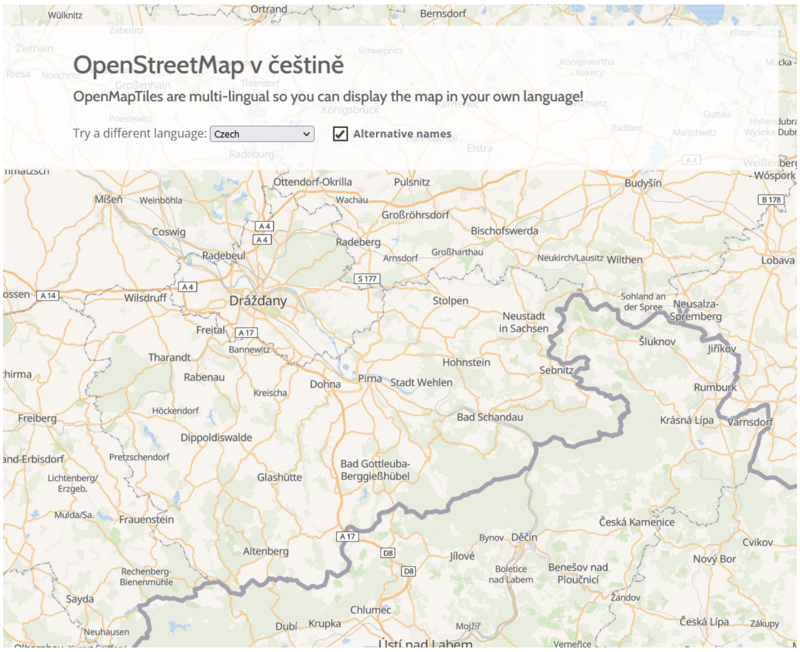


Figure 54: Demo[37] of adjustable names through the usage of vector tiles. All names are shown in Czech if available.

[36] https://maputnik.github.io/

[37] https://openmaptiles.org/languages/cs/#9.05/51.0907/14.0949

5.1.3.4 Viewing map tiles

How the tiles are displayed depends on the technology – raster tiles are usable with any web mapping client or GIS. In contrast, vector tiles are currently bound to some frameworks which allow a performant rendering, such as Mapbox GL JS or MapLibre[38]. They use hardware acceleration for rendering, which means that everything needs to split into discrete triangles according to the map style for using the graphic card of the device. Mobile devices have similar frameworks included in apps to speed up the rendering process. OpenLayers, Leaflet, and QGIS offer support for Vector tiles without acceleration, or the style is converted into its style format.

The tiles are merged visually for the raster tiles or with more computation for the vector tiles to complete cut polygons, for example, in Figure 52. Nevertheless, caching is used on the client's side, and the tiles are requested according to the viewed area and scale. Sophisticated adjustments on demand are possible with JavaScript. It is also possible to adjust the viewed style in points, e.g., changing the field used for naming features to adjust the map style to a language, as in Figure 54. Figure 53 shows, as an example, that it is possible to access the geometry's attributes in vector tiles.

5.1.3.5 Generalization

Due to the explicit distinction between cartographic model generation and visualization in the vector tiles, it is more independent because the model is not only made for one style (Burghardt, Duchêne, and Mackaness 2014). The vector tileset is a cartographic landscape made for each zoom level and ready for visualization. There is also a visualization of the generalization in the documentation, as the excerpt of Figure 55 shows. The later styling depends on the attributes and is no topic here. The generalization operator selection, omitting, and simplification are used in this workflow (Kraak and Ormeling 2010; Slocum et al. 2009).

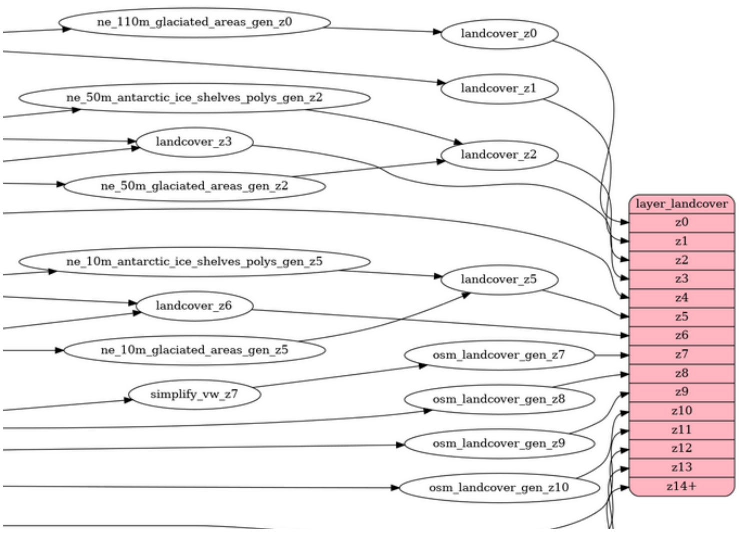


Figure 55: Part of an ETL model for the OpenMapTiles land cover layer.

---

[38] https://maplibre.org/

For each zoom level and feature class, an SQL-View is created, which is later exported and merged according to zoom level to vector tiles. The views use other tables, which are collected together. The derivation of the lower zoom levels sources the tables of the higher zoom levels, as Figure 58 demonstrates. This is done by clustering [ST_ClusterDBSCAN[39] (Ester et al. 1996)], the selection by area according to the resolution, and filtering according to attributes. In the examples, particular variables and functions are in use:[40]

```
bbox(): Variable limiting the query to the tile extent
ZRES(): A function returning the pixel resolution of that
zoom levels
Z: Returns the zoom level for the given scale
LabelGrid(): Label grid function for the selection of popu-
lated places
LineLabel(): Function which tries to estimate if a label would
fit on a line
```

Many of the functions were initially developed by Mapbox and are freely available SQL functions.[41] The functions adjust the selection to pixel size, not to the actual area size of the features.

The generation of the cartographic model contains elements that are similar to a ladder solution (Kraak and Ormeling 2010, 99), as Figure 55 and Figure 56 show. Data for the next zoom level is derived from the previous level. Nevertheless, the star solution appears in the workflow.

In general, the PostGIS functions ST_Simplify[42] or ST_SimplifyVW[43] reduce the vertex numbers, while ST_SnapToGrid[44] reduces the precision of the coordinates to create better compression. Another generalization approach of aggregation and simplification is clustering small pieces, merging them after expansion, and shrinking them back, as implemented in Figure 57. This is a vector version of the MapAlgebra expand and shrink approach (Jaakkola 1998; Z. Li 2007).

Figure 54 shows an example of simplification, which is always adjusted to the zoom levels and aims for maximum compression to reduce tile size. When the tile is generated, ordering occurs for later rendering. This is needed to render transportation features, such as railway lines and roads at crossings, bridges, and tunnels; therefore, the z_order attribute is present. Similarly, settlements are ordered according to their population to ensure that big cities are shown before smaller ones.

---

[39] https://postgis.net/docs/ST_ClusterDBSCAN.html
[40] https://github.com/openmaptiles/openmaptiles-tools/tree/master/sql
[41] https://github.com/mapbox/postgis-vt-util
[42] https://postgis.net/docs/ST_Simplify.html
[43] https://postgis.net/docs/ST_SimplifyVW.html
[44] https://postgis.net/docs/ST_SnapToGrid.html

```
-- etldoc: simplify_vw_z8 ->  simplify_vw_z7
CREATE TABLE simplify_vw_z7 AS
(
    SELECT subclass,
           ST_MakeValid(
            ST_SnapToGrid(
             ST_SimplifyVW(geometry, power(zres(7),2)),
             0.001)) AS geometry
    FROM simplify_vw_z8
    WHERE ST_Area(geometry) > power(zres(5),2)
);
CREATE INDEX ON simplify_vw_z7 USING GIST (geometry);
```

Figure 56: Table for intermediate data showing the derivation of further zoom levels in the generalized.sql for land cover.

The OpenMapTiles workflow also includes enrichment for map production as a part of the generalization (Burghardt, Duchêne, and Mackaness 2014, 166). Therefore, the data from OpenStreetMap and Wikidata are combined, which helps close data gaps in OpenStreetMap and collect all possible information. The massive use of the Natural Earth data set, already generalized for a scale, is also a kind of enrichment and simplifies the processing of the map tiles. Urban areas, borders, and places are used for the low to middle zoom levels to create overview maps.

```
DROP TABLE IF EXISTS cluster_zres14;
CREATE TABLE cluster_zres14 AS
(
WITH single_geom AS (
        SELECT (ST_Dump(geometry)).geom AS geometry
        FROM osm_landuse_polygon
        WHERE landuse='residential'
    )
    SELECT ST_ClusterDBSCAN(geometry, eps := zres(14), minpoints := 1) over () AS cid,
           geometry
    FROM single_geom
);
CREATE INDEX ON cluster_zres14 USING gist(geometry);


DROP TABLE IF EXISTS cluster_zres14_union;
CREATE TABLE cluster_zres14_union AS (
SELECT ST_Buffer(
            ST_Union(
                ST_Buffer(
                    ST_SnapToGrid(geometry, 0.01)
                    , zres(14), 'join=mitre'
                )
            ),-zres(14), 'join=mitre'
        ) AS geometry
FROM cluster_zres14
GROUP BY cid
);
CREATE INDEX ON cluster_zres14_union USING gist(geometry);
```

Figure 57: The clustering, expanding, merging, and shrinking of land use polygons in the prep_landuse.sql file.

```
CREATE OR REPLACE VIEW landcover_z5 AS
(
-- etldoc: ne_10m_glaciated_areas_gen_z5 ->  landcover_z5
SELECT
    geometry,
    subclass
FROM ne_10m_glaciated_areas_gen_z5
UNION ALL
-- etldoc: ne_10m_antarctic_ice_shelves_polys_gen_z5 ->  landcover_z5
SELECT
    geometry,
    subclass
FROM ne_10m_antarctic_ice_shelves_polys_gen_z5
    );
```

Figure 58: Creation of a zoom level in the land cover layer by an SQL-View in the landcover.sql.

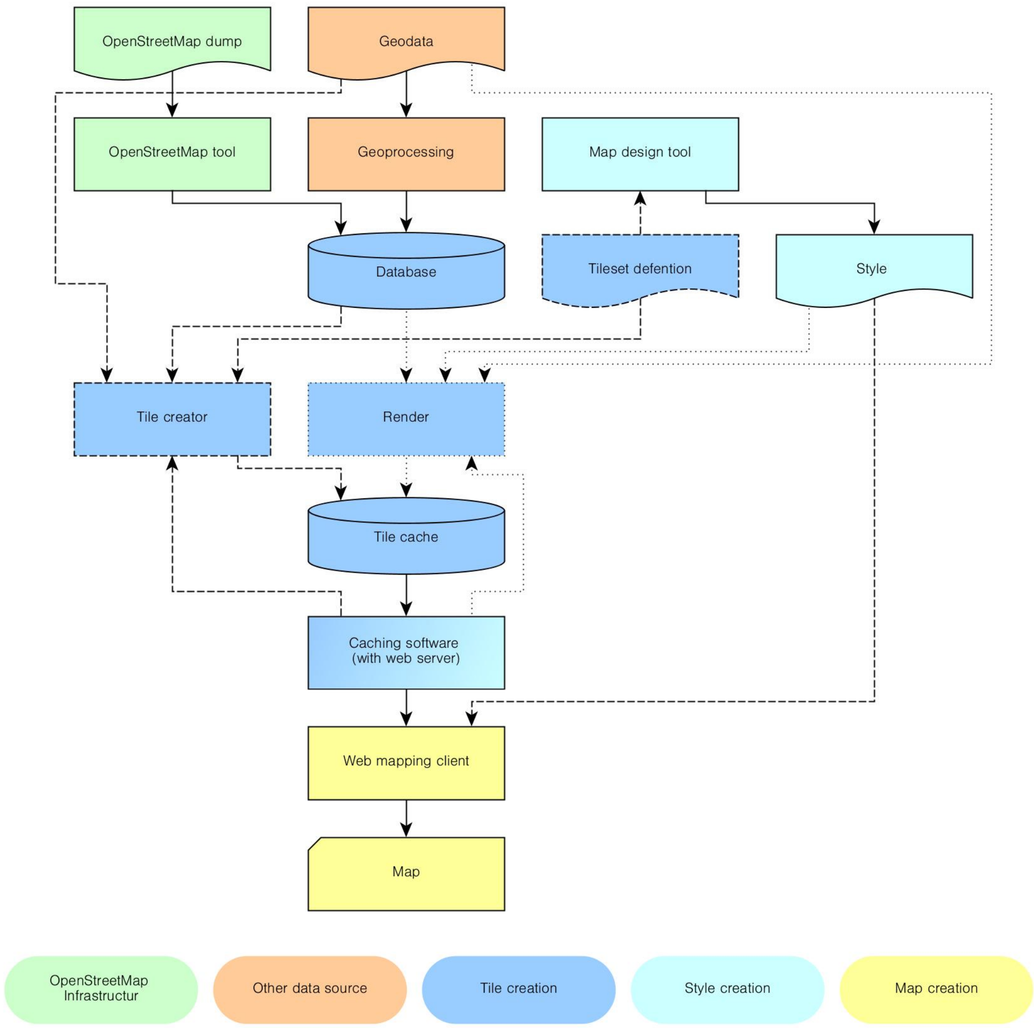


Figure 59: A general workflow for creating multiscale maps. Dashed lines and boxes are needed to produce raster tiles.

5.2 Methods of multiscale map production

This section is an abstraction of the OpenStreetMap Carto and OpenMapTiles workflow to give a general overview of creating multiscale maps based on OpenStreetMap. Figure 44 displays the workflow tools and their functions for map creation. This should be seen as a sample construction kit for a multiscale map production stack, aiming for no completeness. Figure 59 presents a general overview of the connection between the tool categories and the steps to create raster or vector tiles.

5.2.1 OpenStreetMap tools

For any worldwide map, OpenStreetMap is a possible data source because there is no other comparable data source available. OpenStreetMap uses its own data concept and is not compatible with standard GIS. There are two sophisticated tools to convert OpenStreetMap data into the usual GIS data formats: IMPOSM and osm2pgsql, which can perform a custom model transformation from OpenStreetMap data into a relational database. Nevertheless, other tools offer limited import options, such as GDAL and osm2pgrouting. Working with the diff files and the tile expiration is related to map production and is too complex to be replaced by tools other than IMPOSM and osm2pgsql. Another category of software is small helper programs, such as osmconvert or osmium, which allow pre-processing steps, such as creating regional extracts or filters for specific tags.

5.2.2 Geoprocessing

Converting geodata into another format is a less specific task and can be done with a variety of tools. Importing data into the database is a task in the map production workflow. A more complex task is processing data with algorithms to generalize. The workflow processing often happens while creating the tiles inside the database, as shown in Figure 44 and Figure 51. Usually, PostgreSQL, and its spatial extension PostGIS, is the tool of choice due to their performance and extensive collection of algorithms. Generally, this step can be summarized as geoprocessing using command line tools and a geodatabase. A desktop GIS for some pre-processing could be considered, but it would become a bottleneck for more extensive datasets.

5.2.3 Database

In the context of map production, a database is the central data storage. The ability to work with a large amount of data from relational databases by design makes it the tool of choice. PostgreSQL is the only practical option here, along with its spatial extension PostGIS. It is freely available, and there are additional tools that build on it and its functions. PostgreSQL allows read and write access for multiple users simultaneously, and performs even better than the classic Shapefile (Růžička 2016). A small-scale SQLite extended to SpatiaLite could also be an option, offering similar functions but only allowing multiple users write access. It is more common to use this as a container for tiles created in the MBTiles format, which solves disk space limitations and allows a simple exchange of tiles.

Table 6: Example of open source software for creating multiscale maps. Whether or not the tool is suitable for a toolchain “stack” of raster or vector tiles is noted.

	Stack		Possible tasks in the stack						
Tool name	Raster tiles	Vektor tiles	OSM tool	Geoprocessing	Database	Render	Cache	Map design	Web mapping client
Carto	x							x	
Fresco	x	x		x		x			
GDAL	x	x		x		x			
GeoServer	x	x				x			
GeoWebCache	x	x					x		
IMPOSM		x	x						
Kosmtik	x							x	
Leaflet	x	x							x
magnacarto	x							x	
Mapbox GL JS	x	x							x
Maperitive	x	x				x		x	
MapLibre	x	x							x
Mapnik	x	x				x			
MapProxy	x	x					x		
MapServer	x	x				x			
Maputnik	x	x						x	
Mapzen Tanagram	x	x							x
mod_tile	x						x		
OpenLayers	x	x							x
OpenMapSurfer	x					x		x	
osm2pgsql	x	x	x						
osmium	x	x	x						
Osmosis	x	x	x						
pg_tileserv		x				x			
Planetiler		x		x		x			
PostgreSQL/PostGIS	x	x		x	x	x			
QGIS	x	x		x		x		x	x
SpatiaLite		x			x				
Tegola		x		x		x			
Tilemaker		x		x		x			
TileMill	x					x		x	
Tileserver gl	x	x				x	x		
Tilestache	x	x					x		
tippecanoe		x				x			

5.2.4 Creating tiles

To create map tiles, the chosen technology is essential. In the case of raster tiles, the map style is applied to the data and the result is saved in an image format cut to a specific extent. mapnik is the tool of choice for worldwide maps because of its performance, but MapServer, QGIS, and GeoServer also offer similar functions. The type of tool used depends on the scope of the project and the developer's skills.

Vector tiles are somewhat different. For the map rendering, the necessary data and attributes are stored in a particular format—usually MVT—and saved in the cache for later serving. A tool for creating vector tiles is less a render than a file format converter, configured only to include the necessary information and the geometric accuracy to reduce the file size. Possible tools are t-rex, Tegola, and pg_tileserv.

Usually, all of these tools use the database for various geoprocessing functions, to implement generalization, and to generate the tile, including the vector tiles themselves. It is also possible to work without a database, as described in the "Stackless approach for tile creation" section.

5.2.5 Caching

It is essential to buffer the data for the massive use of the multiscale map by many users. User studies indicate that users expect a result from an action, e.g., zooming or panning a map, within 0.1 to 1 second (Nah 2004). Combined with data updates, caching is a delicate task requiring particular software. The cache must contain frequently requested map tiles showing the most recent data in the database. The most elaborate function offers the Apache extension mod_tile, which can control the updating process of cached tiles. MapProxy offers a similar function for many standards and caches. If only caching is needed, there are more tools available, like the GeoWebCache or Tilestache. There is also a wide range of solutions for saving the tiles on a disk. MapProxy's documentation[45] offers a good overview of the different approaches.

5.2.6 Styling tiles

In raster-based map production, styling is not only a description of how the data is visualized; it is also a generalization by selecting features and rendering them. In the case of vector tiles, the tileset configuration is independent of the later applied map style but limits the visualization possibilities of the generated vector tiles.

Various software solutions are available for styling. For raster tiles, TileMill, Kosmetik, and magnacarto are options. It is possible to adjust queries for the data processing and the styling in the same software. In the case of vector tiles, this is split into two parts. Tile creation is a separate subject, and editors such as Maputnik only allow activating the style or not based on the included tiles and their attributes.

---

[45] https://mapproxy.org/docs/latest/caches.html

5.2.7 Viewing tiles

Finally, a viewer is needed to puzzle the tiles together to create a seamless map. In general, all viewers work well with raster tiles—there is no need for further specifications, and the raster image tiles are loaded according to the visible extent and scale.

For vector tiles, loading the tiles is a simple task; rendering and applying the map style is complicated and needs many resources. Therefore, only a few tools are well suited: Mapbox GL JS and MapLibe, its free fork. Vector tiles need hardware acceleration to work fast enough in a browser. Leaflet and OpenLayer do not have this acceleration and suffer from various weaknesses, which limits practical use.

5.2.8 The stackless approach to tile creation

In addition to the complex workflow involving multiple tools to create tiles, a so-called "stackless approach" exists. In this case, the tiles are created directly with one program from the data source file without using a sophisticated schema transformation, database storage, or an advanced caching mechanism that allows updating single tiles. Only complete data sets are converted into tiles. The tiles are created to be directly served or stored for later distribution, according to the tool's configuration. The approach is simple because only one configuration is needed, which allows more parallelization and leads to improved performance. Possible tools for directly creating tiles with the stackless approach are t-rex, Planetiler, tilemaker, and tippecanoe, which fit in the toolchain in Figure 60 from Table 6.

Stack

A (software) stack is a set of software subsystems or components needed to create a complete solution to provide a service, platform, or product.

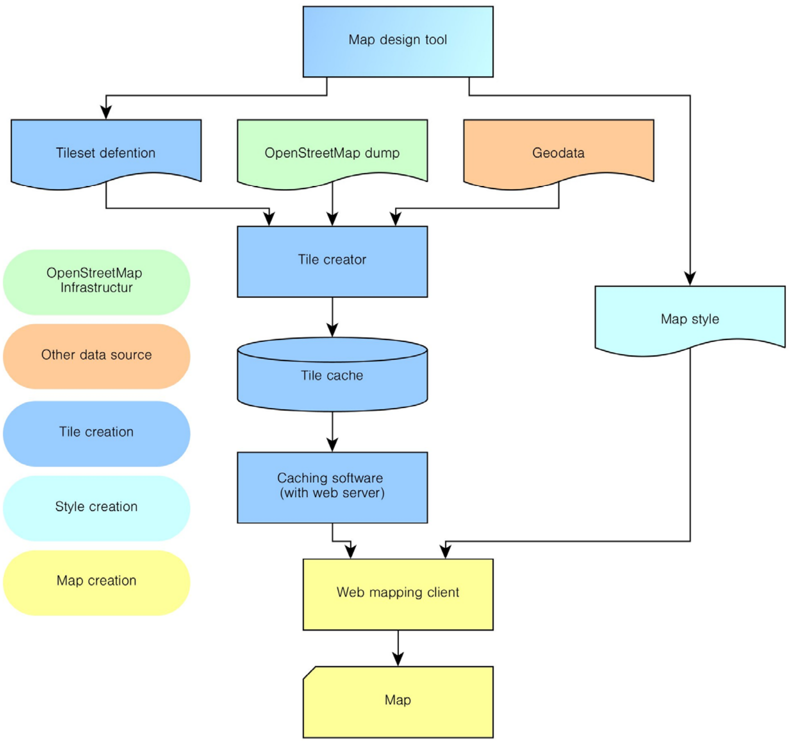


Figure 60: General workflow for the stackless creation of map tiles.

In the Planetiler and tilemaker workflow, no tool for bringing the OpenStreetMap data into a relational data model is required. Input is only raw data OpenStreetMap and from NaturalEarth for lower zoom levels. The output of these tools is vector tiles. Planetiler and tilemaker produce tiles according to the OpenMapTiles schema by default, which allows using the same map styles but is still limited due to technical reasons from zoom levels 0 to 14. The result is a vector tile set that acts as a general multiscale cartographic model created by different tools. When examined in detail, there are differences in the data due to the implementation but not the schema. The result is a dataset that can be used with every style designed for the OpenMapTiles data mode.

It would be, for example, possible to create raster and vector tiles with QGIS, which can directly read OpenStreetMap dumps. However, this would only work in a small area due to the lack of performance.

5.3 Example workflows for creating multiscale maps

This section will implement the proposed generic workflow to produce multiscale maps by applying the two newly developed approaches of micro diagrams and discrete isolation. The discussion will focus on two aspects: vector tiles as a non-standardized solution and raster tiles using the OGC service as the technology foundation for implementation.

5.3.1 Raster tiles: OGC services and micro diagrams

Implementing the micro diagram as a multiscale map in QGIS is the first choice—it can build diagram visualization, create a WMS or WMTS from a project, and implement OGC-certified standards (QGIS 2022). Creating diagrams in web mapping clients is more complicated, so raster tiles are the first choice there. In order to have a complete example of a map production workflow, caching software will be part of the stack for more caching options and better performance.

Figure 61 shows the micro diagram implementation on multiple scales using free software and the previously created terminologies. It starts with importing the captured tweets from a CSV file and some layers to PostgreSQL with the PostGIS extension, using the GDAL library's ogr2ogr tool. The NaturalEarth data is used to create a customized reference map. SQL queries transform the data to create the micro diagram and the required raster layers ready for rendering with QGIS.

Example online

The result of the described example is available online under the following URL, also encoded in the QR code: https://mathias-groebe.de/demo/raster-demo.html

MapProxy

MapProxy is an open source software for mixing, caching, and tiling geoweb services. As a proxy, it can accelerate existing geospatial services by pre-generating and caching data. Another use case is converting other service types (for example, WMS to TMS) or combing services.

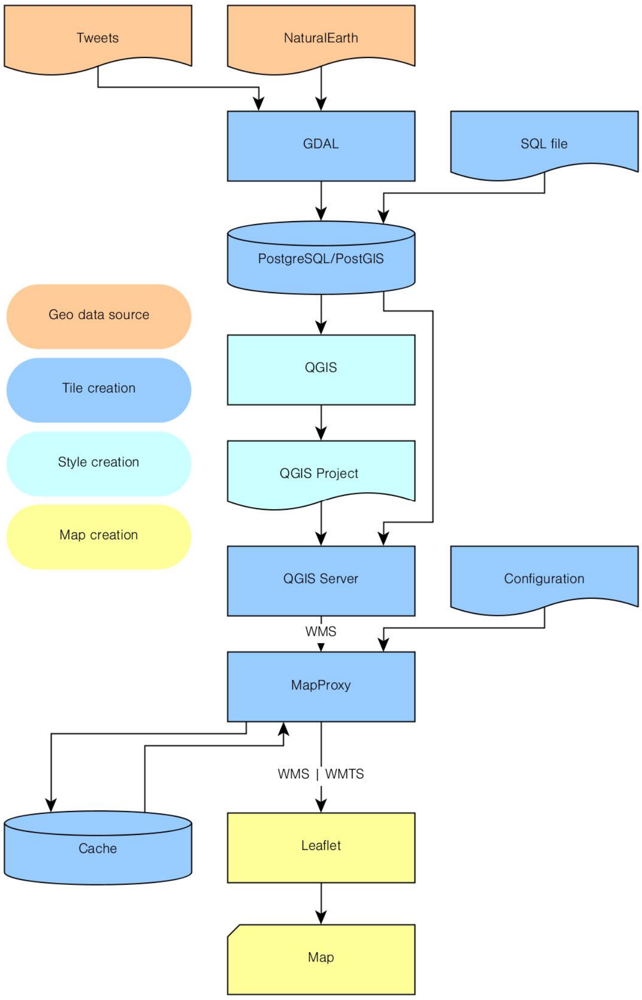
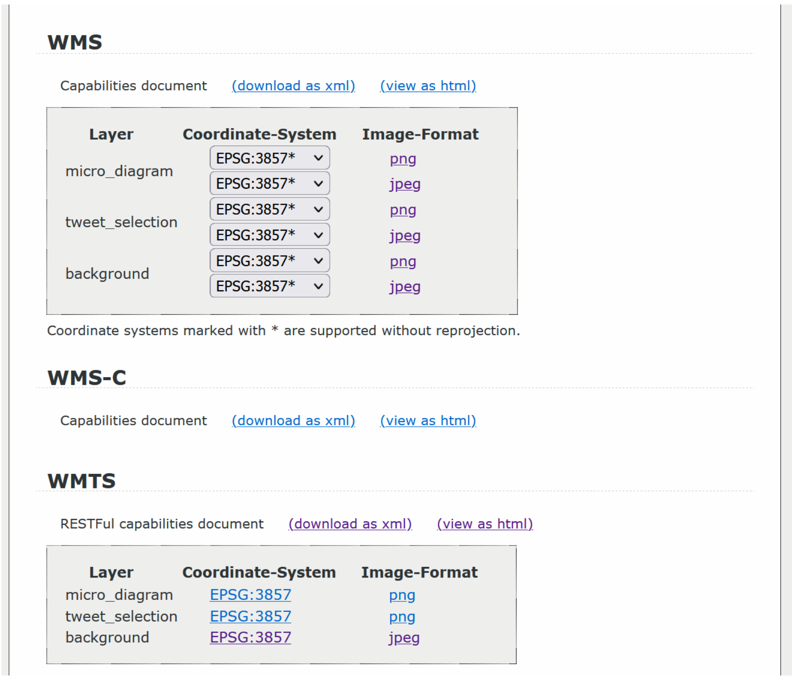


Figure 61: Implementing micro diagrams as a visualization service on multiple zoom levels with caching.

QGIS visualizes the geodata from the database and offers tools for creating styles and layers, as in Figure A84. A diagram visualization makes the micro diagrams, and the point map is made by adjusting the point symbols. A QGIS project file is made available as a WMS by fulfilling some basic rules, such as naming the layers and transferring them to a QGIS server installation. All layers will be made available as WMS layers in pre-defined coordinate systems like WMS and WMTS. Figure 63 shows the service endpoints of the MapProxy software, and Figure A85 and Figure A87 show their configuration. The MapProxy software is combined with the QGIS server by caching the layer for better performance. Due to the cache, not every map request leads to a new data rendering; only in the event of missing information will MapProxy request a rendering of the extent by the QGIS server. On the front end, Leaflet offers a library. It is possible to use the interface on every browser with no additional software.



WMS

Capabilities document (download as xml) (view as html)

Layer	Coordinate-System	Image-Format
micro_diagram	EPSG:3857*	png
	EPSG:3857*	jpeg
tweet_selection	EPSG:3857*	png
	EPSG:3857*	jpeg
background	EPSG:3857*	png
	EPSG:3857*	jpeg

Coordinate systems marked with * are supported without reprojection.

WMS-C

Capabilities document (download as xml) (view as html)

WMTS

RESTFul capabilities document (download as xml) (view as html)

Layer	Coordinate-System	Image-Format
micro_diagram	EPSG:3857	png
tweet_selection	EPSG:3857	png
background	EPSG:3857	jpeg

Figure 63: MapProxy site listing available services and an HTML site preview.

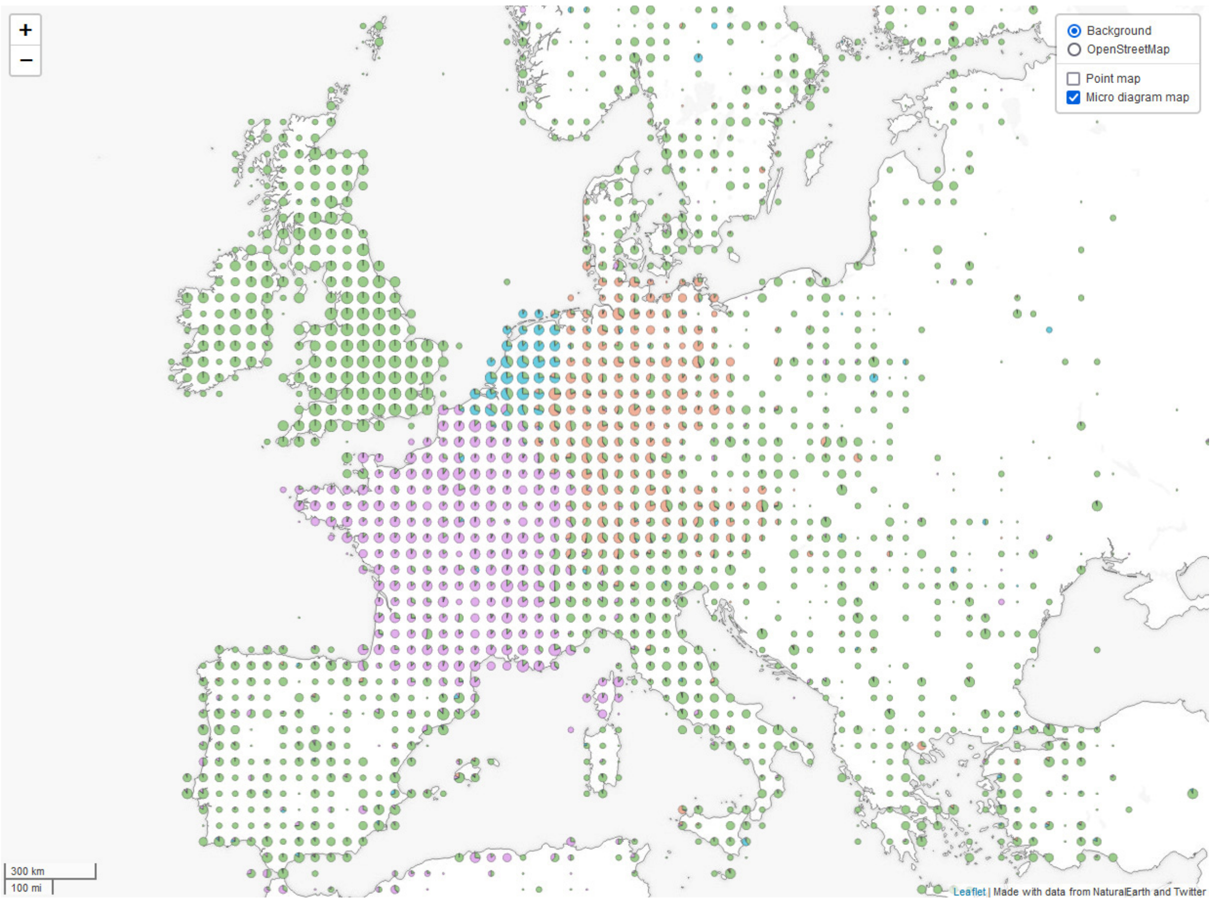


Figure 62: Example micro diagram map using a custom reference map.

The demonstration works similarly in other examples on three scales for Western Europe for English, German, French, and Dutch in Figure 12 and Figure 15. The user can choose different visualization—the point map, the micro diagram map, or the OpenStreetMap Carto style—as reference information, as Figure 62 shows. The extent is variable and allows the user to focus on different regions that can be explored. Figure A86 shows the source code of the viewer referencing the MapProxy services.

Example online

The result of the described example is available online under the following URL, also encoded in the QR code: https://mathias-groebe.de/demo/raster-demo.html

5.3.2 Vector tiles: Slippy map and vector tiles

From the flexibility of the vector tiles, it is possible to demonstrate discrete isolation very well. Storing more points in the tiles and adjusting the density of points in the StyleJSON by filtering after minimum discrete isolation offers excellent options for a designer. The possibility of interactive adjustment makes vector tiles more suitable for demonstration. The workflow for the demonstration uses non-standardized, but open, APIs. Thus, it is a complementation of the no-interactive example.

Figure 64: Workflow for creating vector tiles demonstrating the discrete isolation approach.

The workflow starts with a dump from the OpenStreetMap database, filtered for the place tags to reduce the amount of data for further processing steps. Then, the workflow follows the import into the PostgreSQL database with PostGIS extension, using the osm2pgsql tool according to pre-defined mapping, and storing the key-value-based OpenStreetMap data in a relational database, according to the configuration shown in Figure A78. Next, data processing calculates discrete isolation as a query. In addition, Natural Earth layers are imported by the ogr2ogr tool from the GDAL library into the database, as shown in Figure A77.

For the creation of the tiles, a t-rex software configuration is needed. Using t-rex makes it possible to test the configuration, which consists of basic definitions of the tileset and the SQL queries to create the zoom level content for the different layers inside the tiles. Figure A79 and Figure A80 show some examples from the configuration file. In this step, the Maputnik editor can be utilized to create tile styling. t-rex generates the finished configuration to create the tiles and store them in a standard folder hierarchy on a disk—afterward, the mb-utli imports the tiles into an SQLite database, forming an MBTiles file.

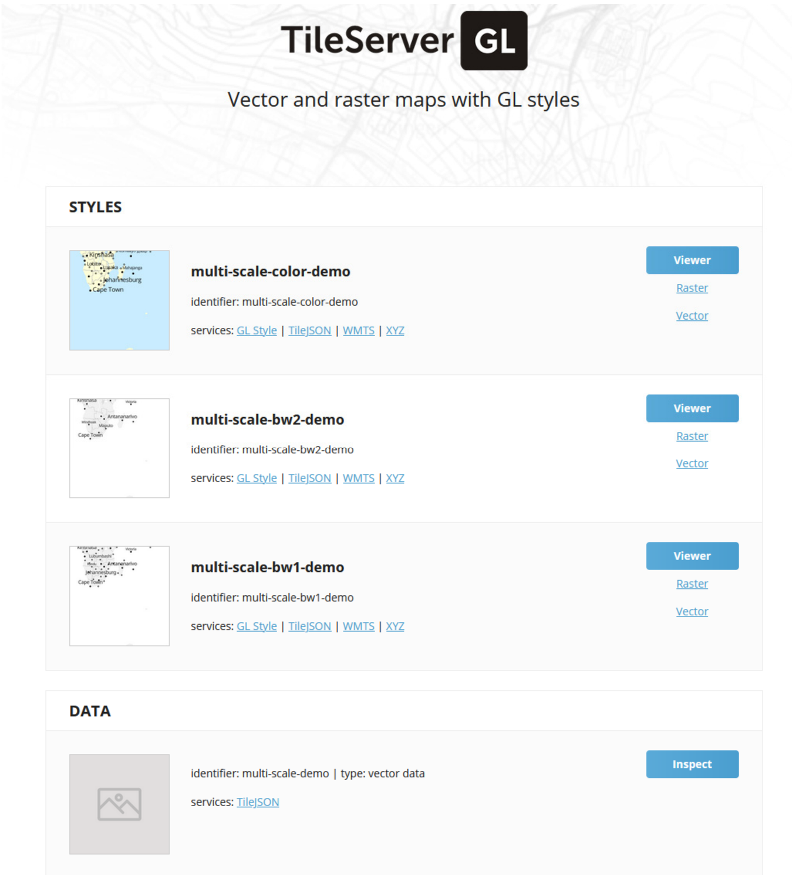


Figure 65: The landing page of the TileServerGL offers an overview of served styles and vector tiles.

Finally, the MBTiles, the glyphs serving as fonts, and the styles are distributed by the TileServer GL. A landing page previews the different styles and data, as in Figure 65. It is possible to view the styles as vector and raster tiles. Figure 66 shows the rendering, which happens when the "Viewer" button is clicked. Using the vector or raster tiles by the defined WMTS or XYZ API is also possible in other applications. The rendering Map Libre is in use, applying the style to the vector tiles or requesting the raster tiles to create the map view in a browser.

Figure 66: Example of a vector tile service using discrete isolation.

The three styles in Figure 65 use the same data by adjusting the options for rendering. It would be possible to create other styles by editing the styleJSON with a text editor. Nevertheless, the hosting is only needed to preview and serve the vector tiles. Only a web browser is necessary for viewing by the associated rendering library. The code behind the visualizations is shown in Figure A81, Figure A82 and Figure A83.

5.4 Discussion of approaches and workflows

After describing, abstracting, and applying the workflows to multiscale map production, they will be placed into the context of map production and compared with each other. The intention is not to compare the advantages and disadvantages of raster and vector tiles as such. Instead, it is examined in which situation the application of raster or vector tiles technology makes sense. Furthermore, the role of VGI and the open source map production approach will be reviewed.

5.4.1 Map production as a rendering pipeline

Generally, the creation of a map is the application of the visualization pipeline (Haber and McNabb 1990; Schumann and Müller 2000; Hauser and Schumann 2009) from the field of computer graphics. Starting with data and, in the end, delivering a map as an image is always the path of digital map production. According to Figure 67, the pipeline facilitates filtering, mapping, and rendering to convert data into the final image.

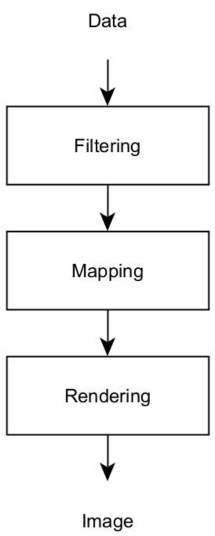


Figure 67: Visualization pipeline (Schumann und Müller 2000).

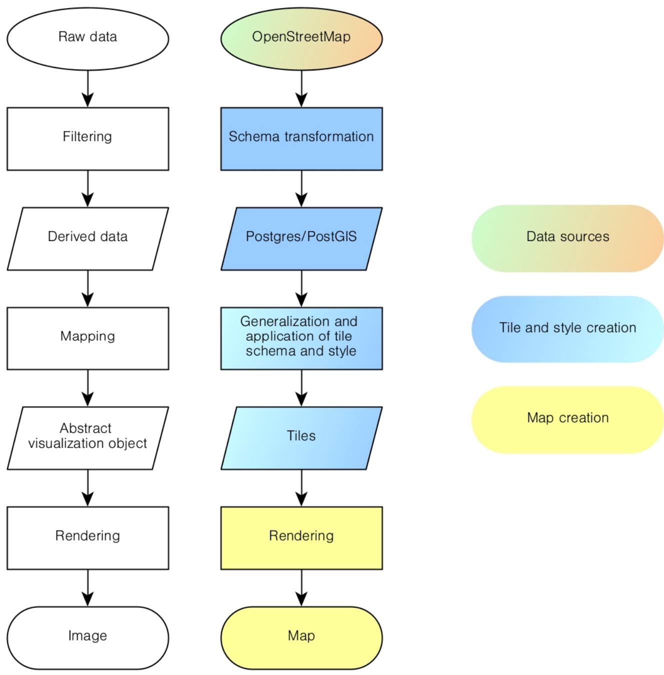


Figure 68: Visualization pipeline with intermediate data, applied in the context of the previously created multiscale map production workflow. Also, compare the coloring with Figure 41, Figure 51, and Figure 60.

The map production process as a specific implementation starts with input data, e.g., from OpenStreetMap, followed by filtering. In the example, it is more a schema transformation from the key-value model than filtering.

The data in between general procedures filtering, mapping and rendering are called intermediate or derived data (Haber and McNabb 1990; Hauser and Schumann 2009), as found in Figure 68.

Nevertheless, the data is filtered, removing unnecessary attributes and geometries during this processing stage. In the PostgreSQL database, the data exits as a derived data model adjusted to the rendering purposes of the data source. In general, mapping a tile creation schema for vector tiles or the styling information for raster tiles are models for a data schema transformation. In any case, tiles result from the mapping and can be seen as abstract objects. For vector tiles, this is much more abstract than for raster tiles because it is already a rendered image. However, a tile by itself has no image or map, making it more of an intermediate product than a ready-to-use map. For the vector tiles, the mapping is also repeated by applying the style file, which leads to the rendering of the map.

It is difficult to distinguish between different tasks in the map production workflows, according to the visualization pipeline and tools are shown in Table 6, because many programs have bundled functions in addition to specific tasks. For example, QGIS can perform all the tasks inside the visualization pipeline in the tile production workflow. Nevertheless, while it is convenient to work with QGIS, as it does not need specialized skills like mapnik or osm2pgsql, it is, however, not so efficient in performance and flexibility when performing those specific tasks. The mapnik renderer has a better rendering performance than QGIS, and for the schema transformation form, the OSM to relation database offers osm2pgsql more options. It makes sense to opt for more specialized tools for such a performance-sensitive workflow as global multiscale map production than just focusing on one software.

In general, it is possible to match the categories OSM tool, geoprocessing, database, renderer, cache, map design, and client from Table 6 and its application in Figure 44, Figure 51, Figure 59, and Figure 60 with the visualization pipeline steps. The OSM tools, geoprocessing, and database is responsible for filtering and storing the data for visualization creation. Last, combined with the map design, the renderer implements the mapping by generating abstract visualization through tile creation. In contrast, the web mapping client performs the rendering from the visualization pipeline. The cache in the workflow acts as a puffer for the visualization to speed up the process, similar to the derived database.

5.4.2 Comparison of OpenStreetMap Carto and OpenMapTiles

The multiscale maps based on the OpenStreetMap Carto style and the OpenMapTiles are likely to be the most used open source map styles based on OpenStreetMap data. Overall, the result is similar for the user: a detailed world map. However, there are more differences than just the technological approach between raster and vector tiling. The following discussion examines their similarities and differences and the reasons behind them.

Both map styles are open workflows, making it possible to fork and extend the projects to meet new needs. They also offer insights into map-making technologies that are otherwise not described. The details are hardly understandable for people who are unfamiliar with the tools and technologies. However, it is accessible as a package for unfamiliar map production users because all the main steps are automated. People familiar with the current technologies can clone the Git repositories and have a running tile service for a small demo area in minutes.

The different purposes of the projects lead to different implementations. A good example is how both services implement updates with different frequencies. OpenStreetMap Carto allows updates minutely, while OpenMapTiles aims for daily updates. It is possible to adjust this behavior, but there is a reason for the longer interval; OpenMapTiles is considered a map service for end users, while OpenStreetMap Carto is a community map that displays as many details as possible, as accurately as possible. It is more motivating for the contributors to OpenStreetMap to see their changes and the results of their work within minutes rather than days. Additionally, the frequent updates allow the rendering of visual checks of the mapping. Nevertheless, the names themselves communicate this behavior—OpenMapTiles references the "open map" from OpenStreetMap.

Accordingly, OpenStreetMap Carto is a community map, whereas OpenMapTiles aims to be a Google Maps pendant, showing only essential data, allowing adjustments to the naming, and enabling rotating labeling. Similarly, development is more community-driven for OpenStreetMap than for the MapTiler company[46] or the OpenMapTiles community. The project developed further based on a bachelor's thesis (Martinelli and Roth, 2016), creating tiles compatible with a Mapbox tiling schema. Starting as "osm2vectoriles," it was later adjusted to have its own data schema after copyright claims from Mapbox ("Move to OSM2VT-Developed Cartography · Issue #387 · Osm2vectortiles/Osm2vectortiles" 2016). Finally, the OpenMapTiles schema is also used by the free and open source software Planetiler and tilemaker, using the stackless approach for a limited zoom range. It will be interesting to see how future map production will develop—straightforward as the stackless approach or more complex as the proposed general workflow for better results.

[46] https://www.maptiler.com

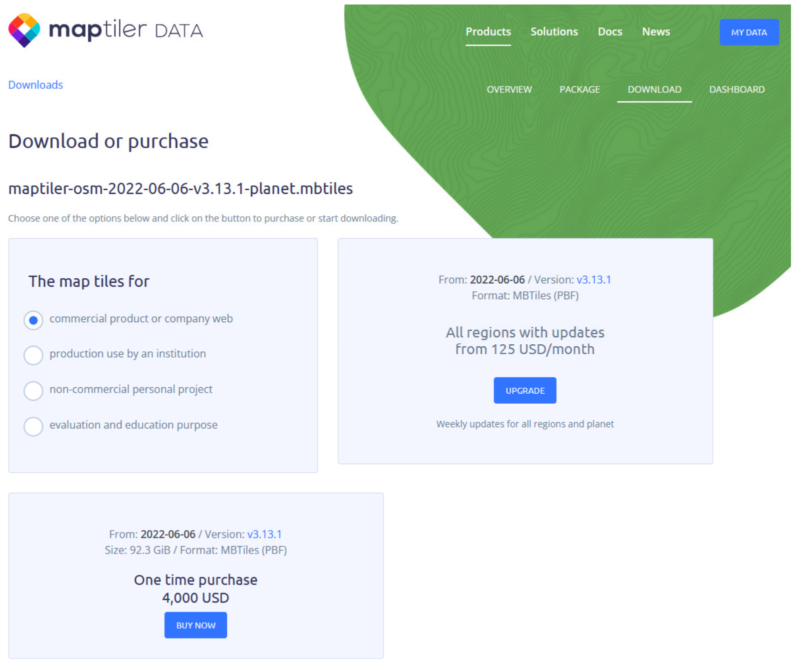


Figure 69: Bundled vector tiles offered from MapTiler (taken on 2022-06-12).

MapTiler offers paid, ready-to-use data downloads, as in Figure 69, and hosting similar to Mapbox and other companies. It is an option for the user who does not have the skills to reproduce the result using the open workflows. Nevertheless, customers can still adjust the map style to their needs, thanks to the flexibility of the vector tiles. Exploring the workflows is complex, less documented, and abstracted from the code itself. For now, map production on a worldwide scale and on multiple scales is still a sophisticated task. It requires an expert in GIS, cartography, and informatics to deal with all the particular approaches: parallelization, special software, expensive hardware, and knowledge about how to stack everything together. As a result, fewer people can make maps for the world, but everyone can contribute by improving the VGI project OpenStreetMap.

5.4.3 Discussion of the implementations

In order to apply the derived workflows from OpenStreetMap Carto and OpenMapTiles, the micro diagrams and discrete isolation were implemented with the tools from Table 6. The implementation details and decisions are bound to the technology choices and standards. The discussion combines this work with more general facts to work out the technical approaches of vector and raster tiles.

5.4.3.1 Raster approach

The main reason for implementing the micro diagrams with raster technology is that there is no vector tile renderer for diagrams. The possibilities for rendering diagrams on maps depend on the software used, which makes rendering with software and sharing only the rendered data a promising approach, in combination with raster tiling technology.

The result is a rendered dataset with one style that allows no adjustments but creates a straightforward workflow from Figure 60: QGIS, with its GUI, which allows the configuration of the style and the service. The data stored in a database is not necessary but enables excellent performance and prevents problems with multiple users accessing the data source simultaneously. The QGIS server makes the rendered data accessible by known and widely used standards, WMS and WMTS. Besides setting up the software, there is no need for more profound knowledge. Caching with MapProxy improves performance and allows the usage of other APIs, such as TMS and XYZ. Nevertheless, a simple webspace could also host if standards are not relevant and tile creation is done with QGIS tools.

5.4.3.2 Vector approach

Demonstrating the flexibility of discrete isolation, the pre-calculated selection measures of vector tiles are more suitable than raster approaches. With the possibility of creating multiple styles from one dataset, the discrete isolation value can be used to create an adjusted information density, as the different styles associated with one vector tile dataset in Figure 65 shows. It allows the fine-tuning of information density by providing parameters to the map designer.

Overall, the complex workflow from Figure 51 requires experience to set up and work with it. The configuration is mainly done with text files. The Maputnik editor has a graphic interface only for the styling part. In addition, there are more configuration options and more things to host, such as the fonts and the mbtiles. Creating and rendering the vector takes time and requires sophisticated graphic acceleration software. Currently, desktop GIS, e.g., QGIS, can display vector tiles directly but need to convert the style for their rendering engine, resulting in a slightly different visualization.[47]

Render library standards restrict the styling possibilities, so implementing the micro diagrams is not possible, which is not very flexible. Otherwise, demonstrating discrete isolation with raster technologies would require several layers and rendering. It is still a de facto standard until the OGC[48] finishes work on their specifications.

5.4.3.3 Juxtaposing

The contradiction between the established but not yet standardized vector tiles will be examined in more detail by comparing them with the raster tiles. Table 7 provides a comprehensive list of raster and vector tile features. As stated in the previous paragraphs, raster tiles only allow one style per tile set, while many styles are possible with vector tiles. In consequence, a raster approach requires more disk storage to display different styles compared to a vector approach. By comparison, raster tile creation is more

---

[47] https://docs.qgis.org/3.22/en/docs/user_manual/working_with_vector_tiles/vector_tiles_properties.html (accessed 2022-11-06)
[48] https://github.com/opengeospatial/ogcapi-tiles

straightforward than vector tile creation. Many GIS can style vector data and export it as raster images, which can be cut into tiles.

For vector tiles, the first step is the creation of the tiles. Afterward, a suitable style needs to be designed with more complex tools, as the demonstrated implementation shows. Also, the visualization possibilities are limited due to the available functions of the renderers and style features from the MapBox GL style specifications. There is a need for open standardization improvements that are not driven by the needs and work of big companies as they are currently.

Table 7: Juxtaposing raster and vector tiles.

Raster tiles	Vector tiles
One tile set, one style	One tile set, many styles
Simple creation	More complex creation
Rendering with GIS software	Rendering with special software
Very flexible styling	Limited styling
Multiple open standards	Standard created by one company
High information density per tile possible	Information density limited by visualization and rendering resources
Need lots of storage capacity	Need less storage capacity
Hide original data	Allow access to geometry and attributes

Figure 70: Examples of different information densities for OpenMapTiles Bright style (upper, vector tiles), OpenStreetMap Carto style (middle, raster tiles), and OpenTopoMap style (lower, raster tiles). (Visualization OpenStreetMap Contributors CC BY-SA 2.0 2022)

The often-mentioned statement that vector tiles save a great deal of storage space compared to raster tiles does not consider information density. Figure 70 demonstrates the different behaviors. The vector tile map provides fewer details and information than the raster tiles. In this example, the reduction of storage also means the loss of information. A related point in this juxtaposing; raster tiles allow no access to the data behind the attributes and geometries. Vector tiles contain inaccurate geometric information, but extracting and re-using the provided data is much easier than digitalization from raster data.

5.4.4 Generalization in map production workflows

Different examples of generalization in multiscale map production were part of this project's exploration. The discussion aims to place those approaches into the framework of the literature for generalization, explaining which generalization operators are in use and the different stages of generalization in the workflows.

Klammer and Burghardt conducted a similar case study (Burghardt, Duchêne, and Mackaness 2014, 139) focused on comparing OpenStreetMap's different generalization models to the workflow of national mapping agencies and how their models are defined. The following are the applied techniques and algorithms of the exploration and discussion.

5.4.4.1 Stages of generalization multiscale production workflows

The generalization process has different stages, which can be described as object generalization, model generalization, and cartographic generalization (Grünreich 1992; Burghardt, Duchêne, and Mackaness 2014, 140). Figure 71 matches the generalization types to the described workflow for OpenStreetMap. Object generalization happens when. An object is digitalized according to a data model. In the case of OpenStreetMap, the contributor captures real-world features with the help of an OpenStreetMap editor and stores them in the database.

Map production needs an adjusted, GIS-based data model created inside a rendering database using specialized schema transformation tools. This is where model generalization happens; depending on the configuration of the OSM tools, the output can differ. The rendering database content is between the initial data model for data capture and the cartographic model. It is optimized for data processing, storage, and fast access but not optimized for the final rendering. There is always an additional transformation before the rendering happens.

In the case of raster tiles, what follows is a combination of model and cartographic generalization. Spatial SQL queries and matching symbols to the data from the renderer configuration further adjust the data model. This is a precise combination of model and cartographic generalization. In this case, no cartographic model exists; it is immediately rendered into an image and can be kept separate because the style is separated from the queries. The queries primarily concern model generalization, while the styling, in some ways, is aimed at cartographic generalization by visually ordering and merging the objects. The labeling is much clearer, a part of the cartographic generalization, and the mapnik render allows different label positions to be tried out in order to choose the best one.

In contrast, creating vector tiles is more like model generalization because it is not bound to the final map style. Nevertheless, some implications make the process a somewhat cartographic generalization. Only a reduced set of attributes are available, and displacement is only possible during tile creation when features for the zoom levels are selected. The vector tile set is a dataset made only for rendering, challenging for other usage. According to this limitation, it can be called a cartographic model. The map style is finally applied to the client, leaving space for further adjustments, e.g., which features are shown and drawn in what order also falls under generalization a bit but more like rendering, applying the map style to the data. Here as well, the labeling is a good example. In vector tiles, labels can only be placed or not placed according to prioritization values—there is no possibility of trying different positions. By saving a separate data model, there is a clear border between data transformation and styling.

Both cases have a fluid boundary between cartographic and model generalizations. However, a cartographic model exists for vector tiles, made for rendering independent of the map style. The vector tiles also have some implications for the map style, reducing the attributes and preparing things for styling. The typical cartographic generalization operation does not happen in these workflows, as Figure 59 shows, not like in the three-stage Grünreich model, which does not fit well here (Grünreich 1992).

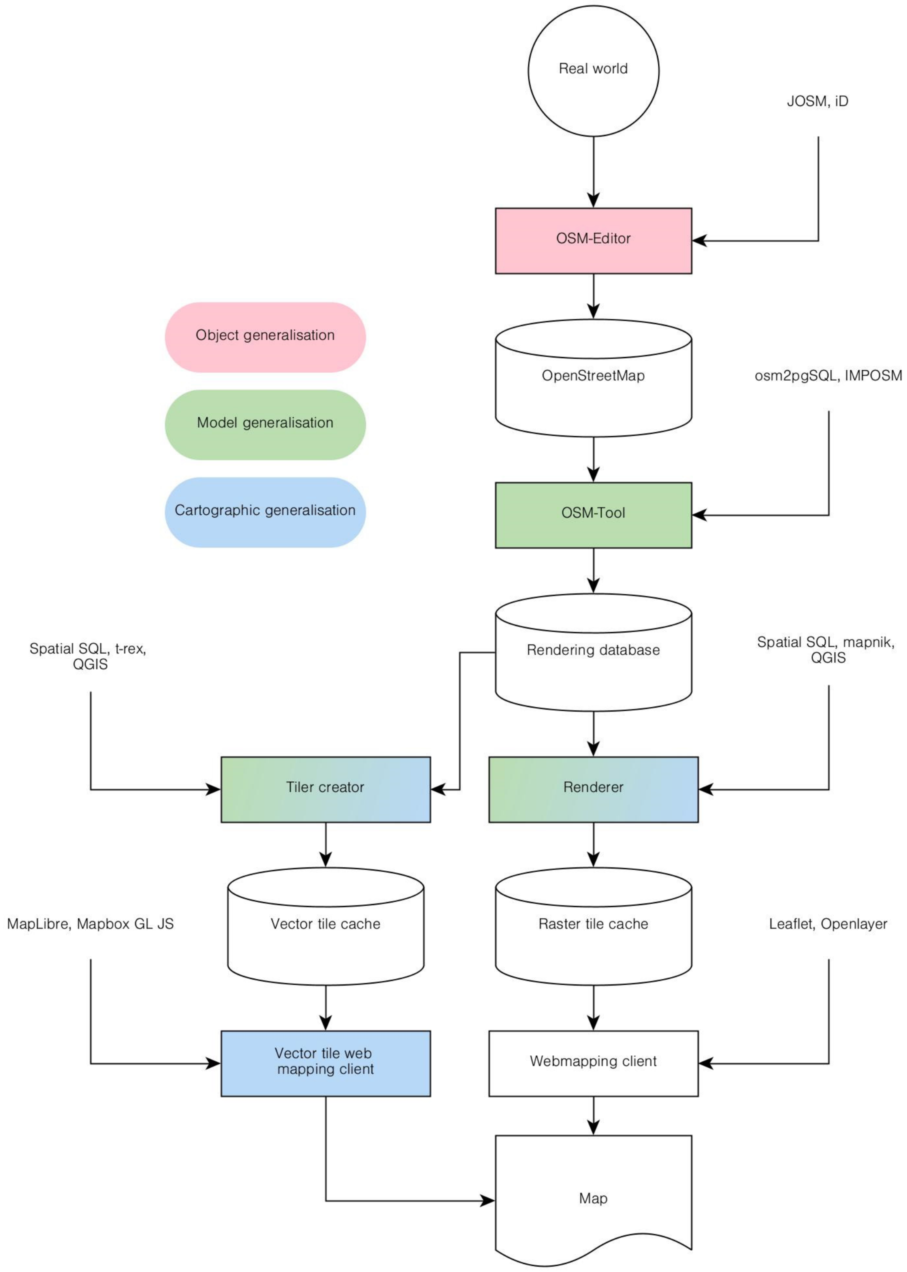


Figure 71: Showing the different stages of generalization in multiscale map production using OpenStreetMap for raster and vector tiles with exemplary software.

Finally, only the map is viewable for raster and vector tiles at the client level. At this point, the tiles are merged into a viewable map. The tiles in the caches are in an intermediate technical format, which has no implications for generalization. They are utilized as a partitioning unit for processing the data, allowing parallelization when the tiles are generated. It is a kind of quadtree, which is also discussed in the literature (Touya et al. 2017).

5.4.4.2 Generalization operators in use

A set of generalization operators in the literature allows the derivation of smaller-scale map data from larger-scale map data on a theoretical basis. Table 8 originally compares commercial software and the different implementation algorithms challenges (Burghardt, Duchêne, and Mackaness 2014, 164). Here the table is a bit simplified according to the needs of the workflows. The table summarizes explorations made in the OpenStreetMap Carto and OpenMapTiles projects on a higher level of abstraction and shows which generalization operators are missing. Processing means that that action occurs during the model generalization while rendering means that the operation happens during the data rendering and may only exist visually for the map viewer.

Line and area simplification happen in both projects. Here, the Douglas-Peucker (Douglas and Peucker 1973) and the Visvalingam–Whyatt (Visvalingam and Whyatt 1992) algorithms are used with PostGIS. The ST_SnapToGrid[49] function reduces coordinate precision for the vector tiles. Line smoothing is only done by mapnik render and used by OpenStreetMap Carto. OpenMapTiles does not use this type of feature because it would increase tile size.

Line merging is used in both projects during processing and rendering. Figure 45 shows an example of the visual merging of lines, which is simple, effective, and saves processing time. For labeling reasons, lines with the same names are often stitched together to avoid unnecessarily repeated labels. Another use case is the generalization of lower zoom levels of lines when they consist of many short pieces. The lines are first merged and, afterward, simplified to avoid the creation of gaps.

Aggregation is also part of both projects and is implemented in the model generalization. This action makes sense for merging object classes before styling, reducing the number of map style rules. The features are collected using the WHERE and CASE functions of PostgreSQL. Geometrically, it would be possible to do a ST_Union[50] with PostGIS, but only OpenMapTiles uses it in combination with ST_ClusterDBSCAN[51]. Often, this would take much more time due to the need to resolve boundaries. It finally happens visually by applying the same style to the features.

---

[49] https://postgis.net/docs/ST_SnapToGrid.html
[50] https://postgis.net/docs/ST_Union.html
[51] https://postgis.net/docs/ST_ClusterDBSCAN.html

Collapse and typification are connected due to similar effects if one object is replaced by an abstract symbolization. There is no typification where one object stands for multiple items. Polygons are often collapsed to points for labeling using the ST_PointOnSurface[52] function to ensure that the point is inside the polygon. POIs are often typified as points because they can be mapped as points, lines, or areas in OpenStreetMap. Figure 48 is an example of this behavior—the color typifies the POI into categories, and the geometries are collapsed into points.

Finally, there are a number of operators missing in the described workflows and other map production software. Line exaggeration seems unnecessary in the workflow but would be possible with a simple buffer and similar area enhancement. Network simplification, neighborhood detection, and displacement are not in use because they are complex computations, which would not fit into the quick update policy of the projects and not be implemented in the used tools. The fact that those operations do not happen in multiscale production is a technical limitation.

Table 8: Overview of applied generalization operators within OpenStreetMap Carto and OpenMapTiles

Generalization operators or algorithms	OpenStreetMap Carto		OpenMapTiles	
	Processing	Rendering	Processing	Rendering
Line simplification			x	
Line smoothing		x		
Line exaggeration				
Line merging	x	x	x	x
Area simplification			x	
Area enhancement				
Aggregation	x		x	
Collapse	x		x	
Typification	x		x	
Network simplification				
Neighborhood detection				
Displacement				

Nevertheless, one principle from the literature is missing in Table 8: selection or refinement. (Hake, Grünreich, and Meng 2002; Slocum et al. 2009, 102; Kraak and Ormeling 2010, 101). It might also be the most used one: Selecting features by their attribute is possible in every GIS, is easy to implement, and is an effective solution for reducing the number of features. It is the essential generalization operator in the reviewed multiscale map production workflows. The model transformation based on those operations and rendering by rules always utilizes this principle, selecting appropriate data for the zoom level according to attributes. It is often mixed with symbolization, as in Figure 49 and Figure 47, especially for raster tiles. Figure 58 and Figure 56 show a clean model transformation for vector tiles. Both tile production workflows use a simple principle: painter's algorithm

[52] https://postgis.net/docs/ST_PointOnSurface.html

(Hughes 2014; Wikipedia 2022), which highlights that drawing order matters. Object rendering happens in a specific order, placing layers and their objects over each other. It is implemented in rendering software such as mapnik, MapLibre GL JS, and others ("MapDesign · Mapnik/Mapnik Wiki" n.d.). The effects are shown in Figure 72 by drawing the layers in an appropriate order so the rivers and lakes do not need special preprocessing. Using this approach saves processing time and simplifies data handling because these lines do not always exist, as the water area in the lower right corner shows. Overall it is a variation of the already explored concept of drawing essential things first. In this example, the more important objects overlay the less important ones.

5.4.4.3 Summary of generalization in the workflows

After closely reviewing the workflow, it can be stated that the theory of cartographic generalization is applicable. The ladder and star solution for data derivation is presented. The object, model, and cartographic generalization steps are present but not as straightforward as in the national mapping agency workflows. There is always a logical separation between style and data transformation, but it can be a part of the style file for raster tiles. Nevertheless, it is not identical to cartographic and model generalization.

OpenMapTiles uses more generalization approaches. It could be due to the need to reduce vector tile size and the fact that the render does not have as many options for rendering as mapnik. Each project uses helper functions, which are adjusted to the Mercator projection's properties, referencing the pixel size, not the natural size, of features. The displacement operator is missing in both workflows; all other operators are present in some way. The reason for this behavior is the complex displacement process, which needs much more computation power and is not present in the used tools.

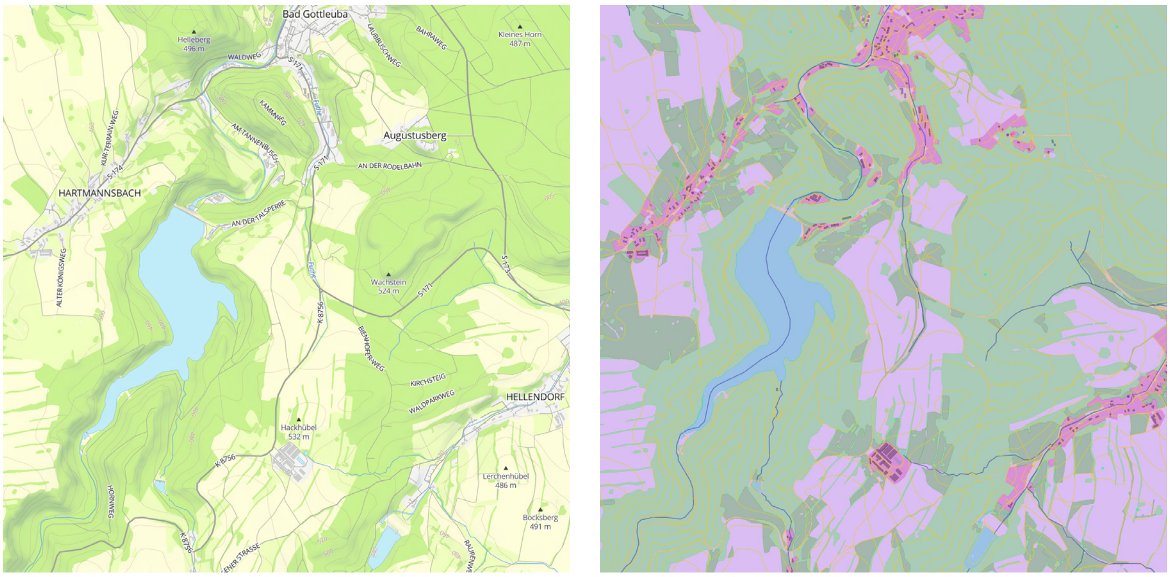


Figure 72: Demonstration of painter's algorithms in map production. The drawing order hides some features, such as the line geometry under the water area. Similarly, the labeling covers all other map features. Both images show the same area with different rendering. Example is from the meinGrün style using the data rendering of the tileserver gl. Data OpenStreetMap contributors 2021.

5.4.5 Conclusions

In 2017, there was less VGI influence on map production (Olteanu-Raimond et al. 2017; Rönneberg, Laakso, and Sarjakoski 2019); OpenStreetMap was only considered a source for the identification, interpretation, and analysis of its own data or missing national mapping agency data. The reasons for this seem to be mainly legal. Things change over time, and users asked German authorities asked for a multiscale map comparable to Google Maps and OpenStreetMap, developed from official data in Germany (Kunz 2018b). The result was the TopPlus Open Project (Kunz and Bobrich 2019), mixing official open data and OpenStreetMap data to close gaps or, ironically, to replace official data, which is not allowed to be used due to legal issues (Kunz 2018b). The work continues with the SmartMapping project and the use of vector tiles (Seifert 2021). So far, developing multiscale maps is now an essential topic for national mapping agencies.

Workflow for worldwide map production

Currently, most worldwide web maps made with OpenStreetMap use the following workflow. The source data is imported into a relational database, and the data model is adjusted and optimized for rendering. Afterward, the tiles are derived by applying the map style for raster tiles or a vector tile model. Generalization happens on the fly, while tiles are created using SQL queries or by selecting datasets appropriate for the target scale. The tiles are pre-processed for fast delivery and sent on demand to the user.

The example of OpenStreetMap and its different multiscale maps shows what is possible by rethinking current map production. It triggers the need for further developments in map production and generalization in the national mapping agencies in combination with the work of big commercial map providers. The developed workflows for raster- and vector-based mapping should simplify the transfer of gained knowledge.

There is always a set of steps for creating a worldwide multiscale map. First, the source data is imported into a relational database, and the data model is adjusted and optimized for rendering. Afterward, the tiles are derived by applying the map style for raster tiles. An intermediate data cartographic landscape model exists for vector tiles before applying the style. The generalization happens on the fly, while tiles are created using SQL queries or by selecting datasets appropriate for the aimed scale. The tiles are pre-processed, saved in caches for fast delivery, and sent on demand to the client. A software library merges the tiles and shows the map to the user. There is also an uprising stackless approach, which directly creates the tiles from the source data and avoids the intermediate rendering database. This is speedy and easy to use but limits the use of tools.

Stackless approach

In the future, the stackless approach for multiscale map creation is the most promising, due to its simplicity and excellent performance.

Currently, most OpenStreetMap plans are made using the more comprehensive traditional workflow, which is easier to adapt. For example, in Germany, the workflows for national mapping agencies would look quite similar. Instead of OSM tools, such as IMPOSM or osm2pgsql, a tool for converting the particular NAS format (Wikipedia 2020) is needed to make the official geodata ready for GIS-based workflows. Similarly, there is a need for other European countries, such as the Netherlands, Norway, and the Czech Republic, to go the same way with their data formats, which are implemented in the GDAL library (GDAL 2022b).

OpenStreetMap and its ecosystem are also highly interesting for researchers who use the data but are also exploring the effects of the contributor and developer community. Some topics include humanitarian mapping for disaster events (Auer et al. 2018), analyzing the mapping and changes (Raifer, Martin et al. 2021), or supporting disabled persons (Mobasheri et al.

2017). However, to date, there is no visible scientific output exploring cartographic development created by the project. There is a link missing between cartographers, cartographic literature, and people who do map production. The manuscript aims to close this gap by explaining the workflows and describing the generalization approaches.

There are new perspectives, such as vector tiles as digital multiscale cartographic models with open schema implementation using different software. The resulting possibilities should be explored, and an exchange between the cartographic community, the developers of multiscale maps, and VGI contributors should be established to further improve map quality. This new technology's cartographic implication has already been explored and discussed in a master's thesis (Beukelaar 2018) and draws a similar conclusion. The topic is also already in discussion in the German, Dutch, and Swiss national mapping agencies (Beukelaar 2018; Seifert 2021; Pippig et al. 2022). The technology's development was driven by companies like Google and Mapbox, starting with raster tiles and moving to vector tiles. However, without VGI, these companies would not be able to earn money and also provide services and software for free in exchange. Whether there is a balance between contributors and companies is another discussion.

Raster tile technology allows map creation and viewing with simple tools combined with high information density. There is no limitation to the visualization possibilities other than the chosen renderer. In contrast, vector tiles need a particular renderer with limited features. Nevertheless, vector tiles are handy for mobile devices, and it is worth investing more resources in their creation and rendering compared to raster tiles. Currently, both technologies have advantages and are widely used. Further vector tile development will allow more visualization flexibility and easier tile creation.

6 Discussion

In addition to discussing the research questions in the related chapters, this chapter focuses on interconnecting topics that merit further consideration. It is a step towards making the thesis findings more general and building a framework for the approaches to multiscale visualization for VGI and LBSM, in addition to the conducted research.

6.1 Development for web mapping

The section "Technologies for serving multiscale maps" introduces a timeline describing the technological development, focusing on the foundation of multiscale map production. The following paragraphs try to sort through the collected information in the web mapping eras according to the literature and the use of official standards.

6.1.1 The role of standards in map production

The analysis of the technological development has shown that maps for a broad audience do not use official OGC standards. The question is, why? The established standards, such as WMS and WMTS, are not used in platforms for broad audiences. The standards are primarily implemented in GIS, and the complimentary software is made for experts, not the mass market. The standards offer needed options, such as different map projections, metadata, protection from unauthorized access, and customization. Experts deal with all of that information and need it for their work.

Different worldwide maps, such as OpenStreetMap, Google Maps, and Bing Maps have their own standards, which are a subset of the official standards. This is probably because the mass market needs good performance, simple maps, and not every option, as there would not be a significant benefit for most users. Furthermore, fewer functions also mean cheaper and more robust software—this is a result of current map technology development driven by big companies. The companies only think about their needs, and the standards maintained by the OGC are created afterward in consultation with the expert community.

6.1.2 Technological development

Publications are trying to summarize web mapping developments by defining different eras. Another perspective sees three time periods (Tsou 2009), starting with people gaining "GIS awareness" by understanding what is possible with GIS technology around 2000. After 2005, with the start of Google Maps, "new computing technologies for the internet GIS" led to the age of "Internet GIS services users and providers," along with its services and APIs for maps and geodata.

There is also a complex view of the different eras (Veenendaal, Brovelli, and Li 2017), mainly driven by the technologies. A static web mapping era, with maps as images, leads to the dynamic web mapping era with adjustable views, followed by the services web mapping era and its APIs. This view also incorporates interactive web mapping and a collaborative web mapping era that allows interaction and uses VGI for map production. The digital globe web mapping era means using globes more for viewing geodata. The mobile and location-based web mapping and cloud web mapping era are also connected by the adjustments that were made for maps on mobile devices. In contrast, the intelligent web mapping era tries to provide context-adjusted complexity. Overall, this approach is an attempt to reflect all technical developments.

Tsou's approach is much more general, matching the fundamental upheavals from developments and providing a better overview. For multiscale map production, it makes sense to define three stages by considering the change of production and styling possibilities.

> **Eras of multiscale map production**
>
> In the pre-tiling era, there were limited options for changing the extent and scale. This was followed by raster tiling, which allowed worldwide scale and extent changes with fixed scales called zoom levels. The latest development is vector tiles, with their adjustable style and flexible zoom levels.

In the pre-tiling era, maps were provided as raster images with less possibility for changing the extent and scale. The significant change was raster tiling, with Google Maps leading the way. Changing the scale and extent on worldwide maps had become the norm, but the scale was bound to a set of scales called zoom levels. The change was the introduction of vector tiles with open specifications by Mapbox, making the style zoomable, moveable, and somewhat independent from the data. In addition, vector tiles allow smooth map zooming; a zoom level is now a float number and not an integer jumping from scale to scale. The vector tile and raster tile era reflect the technological changes that Tsou identified and are now offered as a service. GIS awareness has enabled these developments through compelling examples such as Hurricane Katrina or the Iraq War, portrayed with maps and geodata (Kilday 2018; Tsou et al. 2013).

6.2 New data, new mapping techniques?

As stated in the introduction and the literature (Tsou 2015), new or adjusted methods are needed to deal with the new user-generated geodata. LBSM and VGI have particular properties; their heterogeneous data quality may be redundant, unevenly distributed over time, and come with a complex semantic captured for no fixed scale as usual geodata (Burghardt, Duchêne, and Mackaness 2014, 120).

There are new possibilities connected with the new data sources; global data is easily accessed by API, mostly without fees and good documentation on how to use it. The APIs for Flickr and Twitter are well documented, as OpenStreetMap has done with its Wiki. Compared to cartographic data models from national mapping agencies, the English documentation for those APIs makes them easy to use, and the data model is not so complex.

Assuming that the new data requires the development of new visualization methods (Tsou 2015), relatively little of this is visible in the literature. Otherwise, it would have been uncovered in the "Visualization approaches for crowdsourced geodata" section. There are newer methods like the space-time cube or the micro diagrams, but they have only ins used in a few publications. The identified most often used methods, such as point map, heat map, choropleth map, and nominal line symbol, are not freshly developed for the new data source VGI or LBSM. They are already known and applied to the data. Still, no search was conducted for new visualization methods made for the new data. However, it makes sense that the rising or frequently used methods had been noticeable in the review of publications related to the topic. Of course, there are new visualization methods, such as the approaches of discrete isolation and micro diagrams developed in this thesis. Of course, their use cannot be evaluated yet. Other examples would be the tag maps (Dunkel 2015) or BinSq (Chua and Vande Moere 2016).

Furthermore, new possibilities of known techniques are being used in ways to manage the data; for example, massive point plotting is now possible, and transparency is used to estimate a number. Simple solutions seem to be widely used, as shown by the evaluation of OpenStreetMap map production generalization.

The proposed generalization operators for the new data (Burghardt, Duchêne, and Mackaness 2014, 130) suggest more operators achieve new visualizations; dimension change and displacement are complex techniques with a complex implementation that is hard to manage with standard GIS tools. The simpler ones, such as selection, typification, and aggregation, are implemented with the new micro diagrams. The enhancement of small objects is also complex, but discrete isolation may help filter them out if they are essential according to a numerical measure.

In summary, the development of new visualization methods is driven less by data than by technical capabilities. Complex generalization operators or visualization techniques are used less due to low penetration and complex implementation. The same also applies to the new methods developed in this thesis. It exists for the selection problem of the POIs, the solutions of the label grid, and for the visualization of point collection, the point map visualization, already known and in some points with better properties such as performance.

New data—new visualization methods?

New methods for VGI and LBSM have been developed, but they are rarely used. Instead, existing technical possibilities for data visualization are used.

7 Conclusion

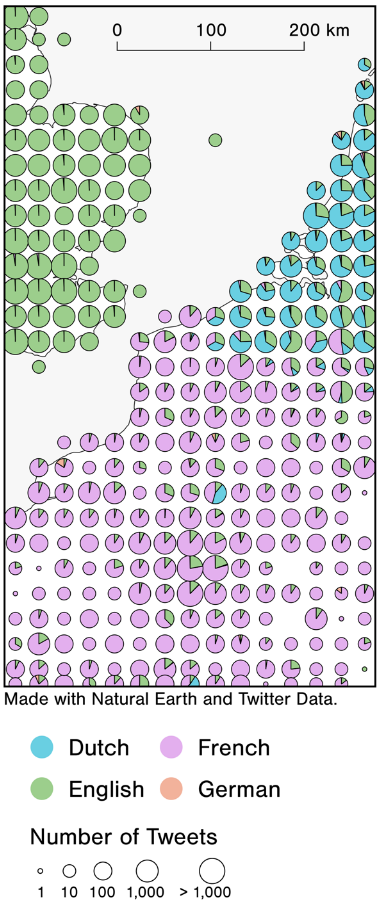


Figure 73: Quantitative visualization for Dutch, French, English, and German Tweets using micro diagrams.

After discussing aspects of the research about new visualization methods for user-generated geodata and models for map production on multiple scales in the previous chapters, the derived findings will be summarized here. In order to achieve this goal, a synthesis of the three research questions will be provided. This will allow the gained specific knowledge to be applied to the broader field of cartography rather than just visualizing geo-related, user-generated content.

7.1 Visualization of point collections

Point collections are generic homogeneous points with a relative position—meaning the contrast of a point of interest with a unique identity and an absolute position—and are very heterogenous (Bereuter 2015). The classification of points in these two groups is flexible and depends on the specific use case and the scale. The new micro diagram visualization method, as in Figure 73, works well for visualizing such point collections, which are typical for user-generated content, with the ability to visualize quantitative nominal scaled attributes. It can deal with an extensive point data set and visualize it without cluttering. The micro diagram is part of the toolset to visualize mapping for location-based social media and big data (Burghardt, Duchêne, and Mackaness 2014, 138; Tsou 2015), offering a new visualization approach suitable for visual analytics and useable as an interactive web map at multiple zoom levels and adjusted aggregation levels.

The micro diagram creation process offers variables to adjust the resulting visualization to meet specific needs. Aggregation can be realized in real-time by geospatial data structures or by classic grids. The diagram type can be chosen according to the data and the number of categories. Diagrams can be scaled by classification or an adjustment function in order to visualize the aggregated data. All of these variables influence the appearance of the final micro diagram map.

The case study and the user test show that micro diagrams are a visualization method that supports the quantitative estimation of overlapping points, especially for different categories. It is feasible to read the visualized category percentages and draw conclusions about the number of visualized values, as demonstrated in the user study. However, it is hard to obtain values for specific categories. This gap can be filled using interactive techniques, such as displaying the numerical proportion and absolute number by hovering a pointer over the diagram or grid cell.

The micro diagram method is a visualization approach that allows experienced map readers, such as data scientists, analysts, and decision-makers, to view and visually analyze extensive point data collections. It offers insights by providing quantitative information, such as numerical proportions and absolute numbers. Its main application is social media visualization. In addition, it can be applied to statistical data, such as a census, and protects privacy by showing only aggregated values for sensitive data.

First research question

How to visualize point collection data with quantitative categorized attributes in as much detail as possible?

Brief answer

Micro diagrams can visualize point collections with nominal scaled attributes without cluttering effects, allowing numerical proportions to be read and the overall

7.2 Visualization of points of interest

Points of interest have unique identities and absolute positions and are heterogeneous information (Bereuter 2015). For visualization, it is essential to show as many of those unique points as the area on the chosen scale allows. A numerical attribution should help rank the priority of the points in addition to spatial distribution.

The new measure of discrete isolation selects points by considering spatial distribution and a numerical attribute. Figure 75 shows an example of the usage comparable to the described problem from the introduction. It is a flexible selection measure derived from the principle of topographic isolation. There are also two other methods: the label grid approach, often used for web map production, and a measure called functional importance. Implementing the three approaches as QGIS plugin point selection algorithms[53] makes them a feasible way to compare different use cases and evaluate performance. The results show that discrete isolation is a versatile selection measure, able to select points on a broad range of scales or zoom levels. Three different use cases evaluated potential applications for discrete isolation. A cartographic use case for selecting populated places was the first example, followed by an analysis of social media data. These use cases demonstrate that the method can help analyze point data sets and identify locally essential points. Finally, an example using a set of scales shows how the selection measure ensures consistent selection, with essential features remaining while others are removed.

Discrete isolation is useable for selecting populated places, peaks, viewpoints, and elevation points for visualization or analysis purposes, as shown in the Twitter data example. A comparison of algorithmic performances has shown that discrete isolation is not as efficient as the label grid. However, it only needs to be computed once for all scales or zoom levels and is combinable with other attributes. The ability to flexibly combine different attributes for selection that only need one computation for different scales of zoom levels makes discrete isolation convenient.

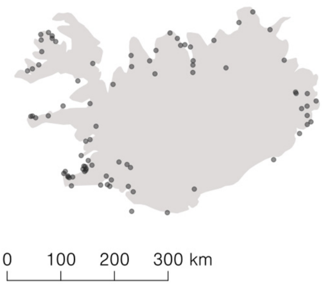


Figure 74: All populated places in Iceland from OpenStreetMap. To obtain an overview, a selection is necessary. (Data OpenStreetMap Contributors 2021 OdbL 1.0)

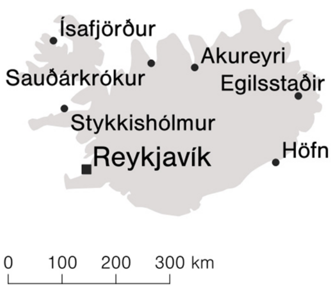


Figure 75: Improved selection of places for Iceland using discrete isolation. (Data OpenStreetMap Contributors 2021 OdbL 1.0)

7.3 Production of multiscale maps

As stated in the introductory chapter, users today expect digital maps with multiple scales due to their experience using maps from commercial players. A review of open source projects explores three different OpenStreetMap projects to answer how worldwide, interactive multiscale maps are made by describing them. The result was a general workflow that can be used to set up new multiscale map services.

OpenStreetMap Carto implements a classic raster tile workflow, while OpenMapTiles focuses on vector tile service. Although both workflows use a similar source, the output and technological approach diverge due to different goals. Another common feature of the two projects is that they are not based on current state-of-the-art web mapping standards.

Second research question

How is a scale-dependent selection of POIs possible for visualization that considers spatial distribution and numerical attributes?

Brief answer

Discrete isolation allows the selection of points of interest by considering spatial distribution and numerical attribute.

[53] https://plugins.qgis.org/plugins/point_selection/

A stackless approach example was also given as an additional, upcoming new technique. A graphic workflow was derived for each project. The three examples created a general workflow based on user-generated geodata from OpenStreetMap as the primary data source.

Third research question

What are the possible workflows to produce a multiscale map, which is currently the most frequently used map?

Brief answer

Identified workflow stages are preprocessing the OpenStreetMap data, storing the data in a relational database, creating the tiles and their caching styling, and finally, the map viewer. The new stackless approach does the preprocessing and tiling in one stage without using intermediate data storage.

Identified workflow stages are the application of OpenStreetMap tools to pre-processing, storing data in a relational database, creating the tiles and their caching, styling the tiles, and finally, the map viewer. In the case of raster tiles, the tiles are styled before caching. The stackless approach performs the pre-processing and tiling in one stage without using intermediate data storage. A selection of suitable software allows the general character of the workflow and its stages to be demonstrated. Vector tiles offer the advantage of adjustable visualization, while raster tiles allow more complex visualization effects.

Two examples show an implementation of the micro diagram and discrete isolation selection approaches using official and informal web standards with open source software. The workflow for multiscale map production is included in this example. Finally, a juxtaposition of raster and vector tile technology completed the examples.

For the micro diagram, raster tiles are the preference because there is no way to create pie charts using WebGL-based rendering libraries, such as MapBox GL and its fork MapLibre. It is less complicated to render raster tiles with a specific GIS, resulting in raster images that every device can display. However, there is no adjustment ability in the visualization, which is an advantage of vector tiles. Discrete isolation values can be stored in the tiles, and the vector tile style can be used to adjust the selection at the final device, creating different styles from one vector tile set. Therefore, it depends on the needed visualization if raster or vector tile technology is the choice, besides various other constraints such as storage capacity, standards, and workflow complexity.

7.4 Synthesis of the research questions

Creating visualization and producing maps on multiple scales is, and probably will remain, a task for experts due to the complexity of the workflows. Map production on a worldwide scale and on multiple scales needs particular approaches: parallelization, special software, expensive hardware, and knowledge about how to handle it. As a result, fewer people can make maps for the whole world, but everyone can contribute to those maps thanks to VGI projects like OpenStreetMap. In addition, open source software provides insights for every enthusiast for free, but participation will still be limited due to the required skill set.

There seems to be a trend toward the stackless approach, which does not need intermediate storage and schema transformation, simplifying multiscale map production. The stackless approach is less complicated due to its shorter toolchain and fewer tools, making debugging and configuration more accessible and reducing computation time (Astrakhan 2022). The free cartographic model of the OpenMapTiles schema[54] makes map style independent from the tile creation process (Pohanka 2022). It is possible to create the tiles with different software or tool chains and apply the same final style, leaving room for improvements in the geometric generalization if the data model is compatible.

Currently, vector tile technology limits the visualization methods. It is foreseeable that this limitation will be less severe in the future, with more kinds of visualizations becoming possible. Furthermore, methods like micro diagrams will be created with vector tiles. Current trends for visualizing the massive amount of new user-generated data will lead to a new tool set.

Those new possibilities will allow for the more efficient creation of multiscale visualizations of user-generated geodata and other geodata. If the foundation, workflows, and tools are recognized, reproducing and adjusting visualizations to specific data outputs will be straightforward. Analyzing vast global datasets is challenging, but it may show hidden patterns in VGI and LBSM. The new micro diagram visualization can explore categories and large amounts of data on these types of multiscale maps. At the same time, discrete isolation can select the most critical points in a data set for a scale.

7.5 Contributions

Contributions from this research are made to two fields: providing methods for visualizing geo-related, user-generated content and a framework for producing multiscale maps. In addition, the technological upheavals from modern web map development were discussed, and the generalization approaches used in the OpenStreetMap Carto and OpenMapTiles projects for map production were explored.

Micro diagrams allow the quantitative visualization of point collations on multiple scales, verified by a user study. Discrete isolation offers a selection of points, which helps identify locations that are relatively more critical to their surrounding points according to a numerical attribute. In addition, an overview of other comparable selection methods and of QGIS plugin implementation were provided.

The second visualization method contribution is a framework and overview of tools for multiscale map production based on the frequently used OpenStreetMap data, showing that multiscale maps are produced alongside standards worldwide. It starts with exploring the technological development of multiscale maps, beginning with raster tiles, and explains the change to vector tile technology. Using the suggested framework, it is possible to build a workflow to produce multiscale maps from OpenStreetMap or another data source. It includes an overview of necessary software categories and possible tools.

---

[54] https://openmaptiles.org/schema/

Figure 76: Example for a different map and its building generalization. OpenStreetMap Carto style (no generalization), openmap.lt (building generalization), and OpenMapTiles Street style (no generalization) at zoom level 15 in Kaunaus.

As a supplement to the map production workflows, OpenMapTiles and OpenStreetMap Carto project generalization were explained through examples and detailed views from existing publications (Burghardt, Duchêne, and Mackaness 2014; Touya and Reimer 2015). There is no sophisticated generalization implementation for a map with many different scales. Due to the need for good performance, generalization is kept simple, and complex cartographic generalizations, such as displacement and network simplification, are not used. Therefore, selection by attribute or geometric measures is used to reduce the number of features. Another reason to keep things less sophisticated is that the projects are developed by non-expert volunteers, who cannot invest more time (Burghardt, Duchêne, and Mackaness 2014, 400).

The result acknowledges the need for future improvements to the cartographic quality of these types of maps by developing adjusted generalization methods and mapping tools that fit into the workflow needs. In addition, traditional approaches, such as radical law, could be adapted to zoomable maps to create a better theoretical foundation for multiscale map production beyond the usual manufacture of topographic maps.

7.6 Limitations

This research on multiscale VGI and LBSM visualization does have limitations. From a scientific point of view, the work is sometimes very technical. Understanding some parts of this thesis requires a technical background, making the research hard to explain when attempting to discuss the outcomes.

The micro diagram map approach offers actual advantages over the point map, but it is still limited. Size and color variations are limited, and they should be clearly distinguishable and readable. This results in fewer quantitative visualizations in cases of extensive data ranges and many categories.

It would have been interesting to implement the identified older point selection approaches and compare them with recent approaches and discrete isolation. The result would have been more approaches to compare and choose from for the QGIS plugin, which might lead to other conclusions.

The multiscale production workflow analysis stems from the author's own years of exploration based on conference presentations, talks, and code repositories. It is hard to cite such big workflows and explain all of the details correctly; a general idea of how map production is implemented should be conveyed. The same applies to the explanation about generalization, which does not aim for completeness, only insights.

There are other open map projects that have, in some parts, more sophisticated generalizations. OpenTopoMap[55] offers an excellent approach to lake labeling, using topographic isolation to select peaks. Freemap.sk[56] creates a

[55] https://opentopomap.org
[56] https://www.freemap.sk

good relief depiction with unique hill shading. openmap.lt[57] uses complex generalization procedures to merge parallel roads, simplify buildings, and label lakes. Figure A77 offers an example of different renderings for the same data and the generalization performed in openmap.lt.

In general, this thesis focused on free and open source software for two reasons: open workflows need open sources, and free knowledge such as VGI is comparable to open software. In addition, no commercial software is comparable to free tools like mapnik and PostgreSQL/PostGIS to create worldwide, multiscale maps.

7.7 Outlook

Exploring the production of today's worldwide, multiscale maps has shown that VGI, incredibly OpenStreetMap, is the data source for maps. The contributions are not stopped by borders; the data is accessible for free and constantly updated by many volunteers. Therefore, every cartographer should know how to improve, use, and handle this valuable data source.

The explored workflow for map production builds on the knowledge of OpenStreetMap and enables the production of multiscale maps. It should be included as a part of primary cartographic education, as print map production works similarly to multiscale map production. A map built on this general workflow using OpenStreetMap can be shifted around and will work everywhere, assuming the topic and scale stay the same and the map projection is adjusted. Web Mercator projection should not be used for printed maps. Nonetheless, a map will always benefit from specific adjustments and manual improvements for better quality.

In order to improve the quality of multiscale maps, cartographic research should accept the challenge and develop techniques that fit into the multiscale map production workflow. The new methods should be able to work on the fly and provide good results within a short processing time. The aim should be to improve map quality by integrating cartographic approaches into the workflows. To achieve this integration, knowledge exchange between cartographers and open source programmers is necessary (Burghardt, Duchêne, and Mackaness 2014, 400), starting with reviewing current possibilities and combining the skills of both fields, as interviews with experts have already shown (Beukelaar 2018). New methods and algorithms should be implemented as open source. This ensures that everyone can test and integrate the new knowledge into an adapted workflow. A good example is the polylabel[58] approach for finding the optimal placement for a point's text label on a polygon. It was inspired by a publication identifying the pole of inaccessibility (Garcia-Castellanos and Lombardo 2007) and implemented by an employee of Mapbox. Afterward, the approach was incorporated into other software due to its excellent performance.

[57] https://openmap.lt
[58] https://github.com/mapbox/polylabel

In addition, existing algorithms should be reviewed and, if suitable, integrated into recently acquired tools. Currently, many functions are lacking that would be helpful for map production. For example, an adjusted line simplification function for buildings in PostGIS would improve OpenStreetMap-based maps like OpenStreetMap Carto or OpenMapTiles. Predictably, the tools will change, but more integration into other software will follow once a good solution is known.

Finally, creating an overview of current visualization methods for VGI and LBSM and their applications would be a solid contribution to the cartography discussion. There is still a knowledge gap for how to visualize classic geodata and manage the complex and extensive data sets of the new user-generated data. Having classifications for the new data and visualization approaches could lead to better visualizations that are easier to create and implement using GIS software.

8 References

Allan, Andy. (2012) 2019. "A General-Purpose OpenStreetMap Mapnik Style, in CartoCSS: Gravitystorm/Openstreetmap-Carto." April 1, 2019. https://github.com/gravitystorm/openstreetmap-carto.

Andrae, Christine, Christian Graul, Martin Over, and Alexander Zipf. 2011. *Web Portrayal Services: OpenGIS Web Map Service, Styled Layer Descriptor, Symbology Encoding Und ISO 19117 Portrayal Vorgestellt Und Erläutert*. Berlin, Offenbach: Wichmann Verlag. http://slubdd.de/katalog?TN_libero_mab216312504.

Antoniou, Vyron, Jeremy Morley, and Mordechai (Muki) Haklay. 2009. "Tiled Vectors: A Method for Vector Transmission over the Web." In *Web and Wireless Geographical Information Systems*, 56–71. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10601-9_5.

Arnberger, Erik. 1993. *Thematische Kartographie: Mit Einer Kurzeinführung Über EDV-Unterstützte Kartographie Mit Quellen Der Fernerkundung*. 3rd ed. Das Geographische Seminar. Braunschweig: Westermann.

Astrakhan, Yuri. 2022. "OSM Planet Data to Vector Tiles in a Few Hours: OpenMapTiles & Planetiler." Presented at the FOSS4G 2022, Firenze, August 26. https://talks.osgeo.org/foss4g-2022/talk/KJAEP8/.

Auer, Michael, Melanie Eckle, Sascha Fendrich, Luisa Griesbaum, Fabian Kowatsch, Sabrina Marx, Martin Raifer, Moritz Schott, Rafael Troilo, and Alexander Zipf. 2018. "Towards Using the Potential of OpenStreetMap History for Disaster Activation Monitoring." In *ISCRAM*.

Battersby, Sarah E., Michael P. Finn, E. Lynn Usery, and Kristina H. Yamamoto. 2014. "Implications of Web Mercator and Its Use in Online Mapping." *Cartographica: The International Journal for Geographic Information and Geovisualization* 49 (2): 85–101. https://doi.org/10.3138/carto.49.2.2313.

Bauer, Christian Alexander. 2010. "User Generated Content – Urheberrechtliche Zulässigkeit Nutzergenerierter Medieninhalte." In *Nutzergenerierte Inhalte Als Gegenstand Des Privatrechts*, edited by Henning Große Ruse-Khan, Nadine Klass, and Silke von Lewinski, 15:1–42. MPI Studies on Intellectual Property, Competition and Tax Law. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-12411-2_1.

Ben, J., X. Tong, and R. Chen. 2010. "A Spatial Indexing Method for the Hexagon Discrete Global Grid System." In *2010 18th International Conference on Geoinformatics*, 1–5. https://doi.org/10.1109/GEOINFORMATICS.2010.5567972.

Bereuter, Pia Sandra. 2015. "Quadtree-Based Real-Time Point Generalisation for Web and Mobile Mapping." Zürich: Universität Zürich.

Bertin, Jacques. 1974. *Graphische Semiologie*. Berlin, New York: DE GRUYTER. https://doi.org/10.1515/9783110834901.

Beukelaar, Ingmar de. 2018. "Cartographic Implications of Vector Tile Technology." https://studenttheses.uu.nl/handle/20.500.12932/30601.

Bleisch, Susanne, and Daria Hollenstein. 2017. "Exploring Multivariate Representations of Indices along Linear Geographic Features." In . Washington.

Bollmann, Jürgen, ed. 2001. *C*. Spektrum Akad. Verl. http://swbplus.bsz-bw.de/bsz089023811rez.htm.

Bostock, Mike and Calvin Metcalf. (2012) 2013. "TopoJSON." 2013. https://github.com/topojson/topojson.

Brewer, Cynthia A. 2005. *Designing Better Maps: A Guide for GIS Users*. 1st ed. Redlands, Calif: ESRI Press.

Brewer, Cynthia A., and Barbara P. Buttenfield. 2007. "Framing Guidelines for Multi-Scale Map Design Using Databases at Multiple Resolutions." *Cartography and Geographic Information Science* 34 (1): 3–15. https://doi.org/10.1559/152304007780279078.

———. 2010. "Mastering Map Scale: Balancing Workloads Using Display and Geometry Change in Multi-Scale Mapping." *GeoInformatica* 14 (2): 221–39. https://doi.org/10.1007/s10707-009-0083-6.

Burghardt, Dirk, Cécile Duchêne, and William Mackaness, eds. 2014. *Abstracting Geographic Information in a Data Rich World: Methodologies and Applications of Map Generalisation*. Lecture Notes in Geoinformation and Cartography. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-00203-3.

Butler, H., M. Daly, A. Doyle, Sean Gillies, T. Schaub, and Stefan Hagen. 2016. "The GeoJSON Format." Request for Comments RFC 7946. Internet Engineering Task Force. https://doi.org/10.17487/RFC7946.

Cao, Guofeng, Shaowen Wang, Myunghwa Hwang, Anand Padmanabhan, Zhenhua Zhang, and Kiumars Soltani. 2015. "A Scalable Framework for Spatiotemporal Analysis of Location-Based Social Media Data." *Computers, Environment and Urban Systems* 51 (May): 70–82. https://doi.org/10.1016/j.compenvurbsys.2015.01.002.

Cao, Nan, Yu-Ru Lin, Xiaohua Sun, David Lazer, Shixia Liu, and Huamin Qu. 2012. "Whisper: Tracing the Spatiotemporal Process of Information Diffusion in Real Time." *IEEE Transactions on Visualization and Computer Graphics* 18 (12): 2649–58.

Cavazzi, Stefano. 2018. "OGC Testbed-13: Vector Tiles Engineering Report." OGC. http://www.opengis.net/doc/PER/t13-DS001.

cfis. 2006. "Google Maps Deconstructed | CFIS." May 3, 2006. https://cfis.savagexi.com/2006/05/03/google-maps-deconstructed/.

Chen, Siming, Lijing Lin, and Xiaoru Yuan. 2017. "Social Media Visual Analytics." *Computer Graphics Forum* 36 (3): 563–87. https://doi.org/10.1111/cgf.13211.

Cheng, Tao, Garavig Tanaksaranond, Andy Emmonds, and Damilola Sonoiki. 2010. "Multi-Scale Visualisation of Inbound and Outbound Traffic Delays in London." *The Cartographic Journal* 47 (4): 323–29. https://doi.org/10.1179/000870410X12911311788152.
Chua, Alvin, and Andrew Vande Moere. 2016. "BinSq: Visualizing Geographic Dot Density Patterns with Gridded Maps." *Cartography and Geographic Information Science*, April, 1–20. https://doi.org/10.1080/15230406.2016.1174623.
Church, Richard L. 2002. "Geographical Information Systems and Location Science." *Computers & Operations Research* 29 (6): 541–62. https://doi.org/10.1016/S0305-0548(99)00104-5.
"Coastline – OpenStreetMap Wiki." 2022. February 11, 2022. https://wiki.openstreetmap.org/w/index.php?title=Coastline&oldid=2263474.
"Coastline Processing." 2022. 2022. https://osmdata.openstreetmap.de/processing/coastline.html.
Cohen, Peter. 2007. "Macworld Expo Keynote Live Update: Introducing the IPhone." *Macworld* (blog). January 8, 2007. https://www.macworld.com/article/183052/liveupdate-15.html.
Crampton, Jeremy W., Mark Graham, Ate Poorthuis, Taylor Shelton, Monica Stephens, Matthew W. Wilson, and Matthew Zook. 2013. "Beyond the Geotag: Situating 'Big Data' and Leveraging the Potential of the Geoweb." *Cartography and Geographic Information Science* 40 (2): 130–39. https://doi.org/10.1080/15230406.2013.777137.
Cresswell, T. 2009. "Place." In *International Encyclopedia of Human Geography*, 169–77. Elsevier. https://doi.org/10.1016/B978-008044910-4.00310-2.
Cynthia Brewer. 2017. "ColorBrewer: Color Advice for Maps." 2017. http://colorbrewer2.org/#.
Douglas, David H, and Thomas K Peucker. 1973. "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature." *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (2): 112–22.
Dransch, Doris, Mike Sips, and Andrea Unger. 2019. "GeoVisual Analytics." In *Geoinformatik*, edited by Monika Sester, 21–44. Springer Reference Naturwissenschaften. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-47096-1_60.
Dumont, Marion, Guillaume Touya, and Cécile Duchêne. 2015. "Automated Generalisation of Intermediate Levels in a Multi-Scale Pyramid." In . Rio, Brazil.
Dumont, Marion, Guillaume Touya, and Cecile Duchene. 2016. "A Comparative Study of Existing Multi-Scale Maps: What Content at Which Scale?" *International Conference on GIScience Short Paper Proceedings* 1. https://doi.org/10.21433/B3112993D46.

Dumont, Marion, Guillaume Touya, and Cécile Duchêne. 2016. "Assessing the Variation of Visual Complexity in Multi-Scale Maps with Clutter Measures." In *19th ICA Workshop on Generalisation and Multiple Representation*. Helsinki, Finland. https://doi.org/10.13140/RG.2.1.4775.3849.

Dunkel, Alexander. 2015. "Visualizing the Perceived Environment Using Crowdsourced Photo Geodata." *Landscape and Urban Planning*, Special Issue: Critical Approaches to Landscape Visualization, 142 (October): 173–86. https://doi.org/10.1016/j.landurbplan.2015.02.022.

Dunkel, Alexander, Gennady Andrienko, Natalia Andrienko, Dirk Burghardt, Eva Hauthal, and Ross Purves. 2019. "A Conceptual Framework for Studying Collective Reactions to Events in Location-Based Social Media." *International Journal of Geographical Information Science* 33 (4): 780–804. https://doi.org/10.1080/13658816.2018.1546390.

Elden, S. 2009. "Space I." In *International Encyclopedia of Human Geography*, 262–67. Elsevier. https://doi.org/10.1016/B978-008044910-4.00320-5.

Ellis, G., and A. Dix. 2007. "A Taxonomy of Clutter Reduction for Information Visualisation." *IEEE Transactions on Visualization and Computer Graphics* 13 (6): 1216–23. https://doi.org/10.1109/TVCG.2007.70535.

Elwood, Sarah, Michael F. Goodchild, and Daniel Sui. 2013. "Prospects for VGI Research and the Emerging Fourth Paradigm." In *Crowdsourcing Geographic Knowledge*, edited by Daniel Sui, Sarah Elwood, and Michael Goodchild, 361–75. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-4587-2_20.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," kdd, 96: 226–31.

Farkas, Gábor. 2020. "Creating the Foundations of a Universal Client-Side Web GIS System." Pécs: University of Pécs.

Ferreira, Nivan, Jorge Poco, Huy T. Vo, Juliana Freire, and Cláudio T. Silva. 2013. "Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips." *IEEE Transactions on Visualization and Computer Graphics* 19 (12): 2149–58.

Fischer, Eric. 2010. "The Geotaggers' World Atlas #5: Berlin." May 19, 2010. https://www.flickr.com/photos/walkingsf/4622375112/in/photostream/.

———. 2011. "Language Communities of Twitter." October 24, 2011. https://www.flickr.com/photos/walkingsf/6277163176/.

Flanagin, Andrew J., and Miriam J. Metzger. 2008. "The Credibility of Volunteered Geographic Information." *GeoJournal* 72 (3–4): 137–48. https://doi.org/10.1007/s10708-008-9188-y.

Flewelling, Douglas M., and Max J. Egenhofer. 1993. "Formalizing Importance: Parameters for Settlement Selection in a Geographic

Database." In , 167–75. https://cartogis.org/docs/proceedings/archive/auto-carto-11/pdf/formalizing-importance-parametersfor-settlement-selection-from-a-geographic-database.pdf.
Fujita, Hideyuki. 2013. "Geo-Tagged Twitter Collection and Visualization System." *Cartography and Geographic Information Science* 40 (3): 183–91. https://doi.org/10.1080/15230406.2013.800272.
Gaffuri, Julien. 2012. "Toward Web Mapping with Vector Data." In *Geographic Information Science*, 87–101. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33024-7_7.
Garcia-Castellanos, Daniel, and Umberto Lombardo. 2007. "Poles of Inaccessibility: A Calculation Algorithm for the Remotest Places on Earth." *Scottish Geographical Journal* 123 (3): 227–33. https://doi.org/10.1080/14702540801897809.
GDAL. 2022a. "SVG - Scalable Vector Graphics — GDAL Documentation." 2022. https://gdal.org/drivers/vector/svg.html.
———. 2022b. "Vector Drivers." 2022. https://gdal.org/drivers/vector/index.html.
"Geohash." 2017. Wikipedia. January 10, 2017. https://en.wikipedia.org/w/index.php?title=Geohash&oldid=759257309.
Ghosh, Debarchana (Debs), and Rajarshi Guha. 2013. "What Are We 'Tweeting' about Obesity? Mapping Tweets with Topic Modeling and Geographic Information System." *Cartography and Geographic Information Science* 40 (2): 90–102. https://doi.org/10.1080/15230406.2013.776210.
"Glyphs | Style Specification | Mapbox GL JS." n.d. Mapbox. Accessed January 18, 2022. https://docs.mapbox.com/mapbox-gl-js/style-spec/glyphs/.
Gonçalves, Tiago, Ana Paula Afonso, and Bruno Martins. 2015. "Cartographic Visualization of Human Trajectory Data: Overview and Analysis." *Journal of Location Based Services* 9 (2): 138–66. https://doi.org/10.1080/17489725.2015.1074736.
Goodchild, Michael F. 1992. "Geographical Information Science." *International Journal of Geographical Information Systems* 6 (1): 31–45. https://doi.org/10.1080/02693799208901893.
———. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69 (4): 211–21. https://doi.org/10.1007/s10708-007-9111-y.
———. 2011. "Challenges in Geographical Information Science." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467 (2133): 2431–43. https://doi.org/10.1098/rspa.2011.0114.
Goodchild, Michael, and Linna Li. 2016. "Formalizing Space and Place." In . Paris. https://hal.archives-ouvertes.fr/hal-01353206/document.
Graham, Mark, Scott A. Hale, and Devin Gaffney. 2014. "Where in the World Are You? Geolocation and Language Identification in Twitter." *The Professional Geographer* 66 (4): 568–78.

Gröbe, Mathias. 2019. "Verification of Multi-Scale Map Design." Presented at the ICC2019 Tokyo, Workshop Abstraction, Scale and Perception, Tokyo, July 15. https://generalisation.icaci.org/downloads/abs2019/Abs2019_paper_7.pdf.

Gröbe, Mathias, and Dirk Burghardt. 2020. "Micro Diagrams: Visualization of Categorical Point Data from Location-Based Social Media." *Cartography and Geographic Information Science* 47 (4): 305–20. https://doi.org/10.1080/15230406.2020.1733438.

Grünreich, Dietmar. 1992. "ATKIS-A Topographic Information System as a Basis for a GIS and Digital Cartography in West Germany." *Geologisches Jahrbuch. Reihe A, Allgemeine Und Regionale Geologie BR Deutschland Und Nachbargebiete, Tektonik, Stratigraphie, Paläontologie*, no. 122: 207–15.

Gugulica, Madalina, Eva Hauthal, and Dirk Burghardt. 2020. "Affective Route Planning Based on Information Extracted from Location-Based Social Media," January. https://doi.org/10.5281/ZENODO.3628877.

Günzel, Stephan, and Franziska Kümmerling, eds. 2010. *Raum: Ein Interdisziplinäres Handbuch*. Stuttgart: Metzler.

Haber, Robert B, and David A McNabb. 1990. "Visualization Idioms: A Conceptual Model for Scientific Visualization Systems." *Visualization in Scientific Computing* 74: 93.

Hahmann, Stefan, and Dirk Burghardt. 2013. "How Much Information Is Geospatially Referenced? Networks and Cognition." *International Journal of Geographical Information Science* 27 (6): 1171–89. https://doi.org/10.1080/13658816.2012.743664.

Hake, Günter, Dietmar Grünreich, and Liqiu Meng. 2002. *Kartographie: Visualisierung Raum-Zeitlicher Informationen*. 8., Vollst. neu bearb. und erw. Aufl. Berlin: De Gruyter. https://www.degruyter.com/viewbooktoc/product/12592?rskey=ecb6xP&result=14.

Hauser, Helwig, and Heidrun Schumann. 2009. "Visualization Pipeline." In *Encyclopedia of Database Systems*, edited by Ling Liu and M. Tamer Özsu, 3414–16. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_1133.

Hauthal, Eva, Dirk Burghardt, and Alexander Dunkel. 2019. "Analyzing and Visualizing Emotional Reactions Expressed by Emojis in Location-Based Social Media." *ISPRS International Journal of Geo-Information* 8 (3): 113. https://doi.org/10.3390/ijgi8030113.

Hauthal, Eva, Alexander Dunkel, and Dirk Burghardt. 2021. "Emojis as Contextual Indicants in Location-Based Social Media Posts." *ISPRS International Journal of Geo-Information* 10 (6): 407. https://doi.org/10.3390/ijgi10060407.

"HCL Wizard - Somewhere over the Rainbow." n.d. HCL Wizard - Somewhere over the Rainbow. Accessed May 23, 2019. http://hclwizard.org/index.html.

Heimerl, Florian, Chih-Ching Chang, Alper Sarikaya, and Michael Gleicher. 2018. "Visual Designs for Binned Aggregation of Multi-

Class Scatterplots." *ArXiv:1810.02445 [Cs]*, October. http://arxiv.org/abs/1810.02445.
heise online. 2018. "Keine flache Erde mehr: Google Maps wechselt zu Globus-Ansicht." heise online. August 4, 2018. https://www.heise.de/newsticker/meldung/Keine-flache-Erde-mehr-Google-Maps-wechselt-zu-Globus-Ansicht-4129252.html.
"History of OpenStreetMap – OpenStreetMap Wiki." n.d. Accessed March 30, 2021. https://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap.
Hollenstein, Livia, and Ross Purves. 2010. "Exploring Place through User-Generated Content: Using Flickr to Describe City Cores." *Journal of Spatial Information Science*, no. 1 (July). https://doi.org/10.5311/JOSIS.2010.1.3.
Hormann, Christoph. 2015a. "Imagico.de - Populated Places in OpenStreetMap." Populated Places in OpenStreetMap. May 2015. http://imagico.de/map/osm_populated_en.php.
———. 2015b. "Science and OpenStreetMap and Why Not Call It 'VGI' | Imagico.De." August 29, 2015. http://blog.imagico.de/science-and-openstreetmap-and-why-not-call-it-vgi/.
———. 2018. "OpenStreetMap-Kartographie - Aus Der Praxis Der Kartengestaltung Mit Offenen Daten." DGfK Vortrag, Dresden, October 30. http://www.imagico.de/files/osmcarto_dresden_2018_small.pdf.
———. 2020. "Weniger Ist Mehr - Zur Auswahl Darzustellender Elemente in Der Digitalen Kartographie." Freiburg im Breisgau, November 3.
———. 2021. "Auswahl Für Karten: Funktionelle Bedeutung," January 12, 2021.
Huck, Jonny, Duncan Whyatt, and Paul Coulton. 2015. "Visualizing Patterns in Spatially Ambiguous Point Data." *Journal of Spatial Information Science*, no. 10 (June). https://doi.org/10.5311/JOSIS.2015.10.211.
Hughes, John F. 2014. *Computer Graphics: Principles and Practice*. 3rd ed. Upper Saddle River, N.J.: Addison-Wesley.
"Humanitarian OSM Team – OpenStreetMap Wiki." n.d. Accessed December 9, 2021. https://wiki.openstreetmap.org/wiki/Humanitarian_OSM_Team.
Imhof, Eduard. 1972. *Thematische Kartographie*. Berlin; New York: De Gruyter. http://public.eblib.com/choice/publicfullrecord.aspx?p=938530.
ircama. 2018. "Description of the OSM Rendering Process." OpenStreetMap Carto Tutorials. May 10, 2018. https://ircama.github.io/osm-carto-tutorials/osm-rendering-process/.
Jaakkola, Olli. 1998. "Multi-Scale Categorical Data Bases with Automatic Generalization Transformations Based on Map Algebra." *Cartography and Geographic Information Systems* 25 (4): 195–207. https://doi.org/10.1559/152304098782383016.

John, Steffen, Stefan Hahmann, Adam Rousell, Marc-O. Löwner, and Alexander Zipf. 2017. "Deriving Incline Values for Street Networks from Voluntarily Collected GPS Traces." *Cartography and Geographic Information Science* 44 (2): 152–69. https://doi.org/10.1080/15230406.2016.1190300.

Jones, Christopher B., and J. Mark Ware. 2005. "Map Generalization in the Web Age." *International Journal of Geographical Information Science* 19 (8–9): 859–70. https://doi.org/10.1080/13658810500161104.

Karsznia, Izabela, and Karolina Sielicka. 2020. "When Traditional Selection Fails: How to Improve Settlement Selection for Small-Scale Maps Using Machine Learning." *ISPRS International Journal of Geo-Information* 9 (4): 230. https://doi.org/10.3390/ijgi9040230.

Karsznia, Izabela, and Robert Weibel. 2018. "Improving Settlement Selection for Small-Scale Maps Using Data Enrichment and Machine Learning." *Cartography and Geographic Information Science* 45 (2): 111–27. https://doi.org/10.1080/15230406.2016.1274237.

Kemp, Karen K., ed. 2008. *Encyclopedia of Geographic Information Science*. Los Angeles: SAGE Publications.

Kessler, Fritz. 2011. "Volunteered Geographic Information: A Bicycling Enthusiast Perspective." *Cartography and Geographic Information Science* 38 (3): 258–68. https://doi.org/10.1559/15230406382258.

Kilday, Bill. 2018. *Never Lost Again: The Google Mapping Revolution That Sparked New Industries & Augmented Our Reality*. First edition. New York, NY: HarperBusiness.

Kounadi, Ourania, Alina Ristea, Michael Leitner, and Chad Langford. 2018. "Population at Risk: Using Areal Interpolation and Twitter Messages to Create Population Models for Burglaries and Robberies." *Cartography and Geographic Information Science* 45 (3): 205–20. https://doi.org/10.1080/15230406.2017.1304243.

Kraak, M. J., and Ferjan Ormeling. 2010. *Cartography: Visualization of Geospatial Data*. 3rd ed. Harlow; New York: Prentice Hall.

Kreveld, Marc Van, Rene Van Oostrum, and Jack Snoeyink. 1997. "Efficient Settlement Selection for Interactive Display." In *In Proc. Auto-Carto 13: ACSM/ASPRS Annual Convention Technical Papers*, 287–96.

Krumm, John, Robert Gruen, and Daniel Delling. 2013. "From Destination Prediction to Route Prediction." *Journal of Location Based Services* 7 (2): 98–120. https://doi.org/10.1080/17489725.2013.788228.

Kunz, Peter. 2018a. "Kartographische Herausforderungen Bei Der Herstellung Der TopPlus-Web-Open." In *38. Wissenschaftlich-Technische Jahrestagung Der DGPF Und PFGK18 Tagung in München*. München.

———. 2018b. "TopPlusOpen." Dresden, November 13.

Kunz, Peter, and Joachim Bobrich. 2019. “Multiscale Cartographic Visualization of Harmonized Datasets.” *International Journal of Cartography* 5 (2–3): 178–94. https://doi.org/10.1080/23729333.2019.1610931.

Langhans, Kerstin, and Stefan Wiemann. 2021. “Amtliche Kartographie in Sachsen – Offen, Digital Und Blattschnittfrei.” Dresden, May 11. https://dresden.dgfk.net/Veranstaltung/amtliche-kartographie-in-sachsen-offen-digital-und-blattschnittfrei/.

Langran, MS. 1986. “Integration of Name Selection and Name Placement.” In *Proceedings, Second International Symposium on Spatial Data Handling, 1986*.

Lardinois, Frederic. 2012. “Google Maps API Gets Massive Price Cut In The Wake Of Developer Defections.” TechCrunch. *Google Maps API Gets Massive Price Cut In The Wake Of Developer Defections* (blog). June 22, 2012. https://techcrunch.com/2012/06/22/google-maps-api-gets-massive-price-cut-in-the-wake-of-developer-defections/?guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAADSpFEHn65Zg2dAQev-KMvFLyITA1nrsgjQ6qEJdycwJSnPHgSptvplxO2Ie-KiHwSdl0eoNR6lMMjreJBk-nep-nbUitP_6qq3m0eW_7m0Sdy4SgB7DiGbc10fXAYAsw5iGoIWD10lsn0k2IVXK1cwK2oU7BfS7LCKzqNBY3jdkTyc&guc-counter=2.

Laubert, Hans, Erwin Woska, and Rudolf Habel. 1988. *Lehrbuch Für Kartographiefacharbeiter: 2 Kartengestaltung*. 2. Aufl. Gotha: Haack, Geographisch-Kartographische Anst. http://slubdd.de/katalog?TN_libero_mab21321219.

Lee, Wang-Chien, and Mao Ye. 2017. “Location-Based Social Networks.” In *Encyclopedia of Social Network Analysis and Mining*, edited by Reda Alhajj and Jon Rokne, 1–14. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-7163-9_319-1.

Leetaru, Kalev, Shaowen Wang, Guofeng Cao, Anand Padmanabhan, and Eric Shook. 2013. “Mapping the Global Twitter Heartbeat: The Geography of Twitter” 18 (5). http://journals.uic.edu/ojs/index.php/fm/article/view/4366.

Li, Lin, Wei Hu, Haihong Zhu, You Li, and Hang Zhang. 2017. “Tiled Vector Data Model for the Geographical Features of Symbolized Maps.” Edited by George-John Nychas. *PLOS ONE* 12 (5): e0176387. https://doi.org/10.1371/journal.pone.0176387.

Li, Linna, Michael F. Goodchild, and Bo Xu. 2013. “Spatial, Temporal, and Socioeconomic Patterns in the Use of Twitter and Flickr.” *Cartography and Geographic Information Science* 40 (2): 61–77. https://doi.org/10.1080/15230406.2013.777139.

Li, Zhilin. 2007. *Algorithmic Foundation of Multi-Scale Spatial Representation*. CRC.

Lienert, Christophe, Bernhard Jenny, Olaf Schnabel, and Lorenz Hurni. 2012. “Current Trends in Vector-Based Internet Mapping: A

Technical Review." In *Online Maps with APIs and WebServices*, 23–36. Lecture Notes in Geoinformation and Cartography. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27485-5_3.

Lin, Bingxian, Liangchen Zhou, Depeng Xu, A.-Xing Zhu, and Guonian Lu. 2018. "A Discrete Global Grid System for Earth System Modeling." *International Journal of Geographical Information Science* 32 (4): 711–37. https://doi.org/10.1080/13658816.2017.1391389.

MacEachren, Alan M., Anuj Jaiswal, Anthony C. Robinson, Scott Pezanowski, Alexander Savelyev, Prasenjit Mitra, Xiao Zhang, and Justine Blanford. 2011. "Senseplace2: Geotwitter Analytics Support for Situational Awareness." In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference On*, 181–90. IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6102456.

Madelin, Malika, Claude Grasland, Hélène Mathian, Léna Sanders, and Jean-Marc Vincent. 2009. "Das „MAUP ": Modifiable Areal Unit-Problem Oder Fortschritt?" *Informationen Zur Raumentwicklung* 10: 645–60.

Mamei, Marco, and Massimo Colonna. 2018. "Analysis of Tourist Classification from Cellular Network Data." *Journal of Location Based Services* 12 (1): 19–39. https://doi.org/10.1080/17489725.2018.1463466.

Mapbox. (2014) 2021. "Mapbox Vector Tile Specification." Mapbox Vector Tile Specification. December 2, 2021. https://github.com/mapbox/vector-tile-spec.

"MapDesign · Mapnik/Mapnik Wiki." n.d. GitHub. Accessed December 4, 2022. https://github.com/mapnik/mapnik.

Martinelli, Lukas, and Manuel Roth. 2016. "Updatable Vector Tiles from OpenStreetMap." Rapperswil: Hochschule für Technik Rapperswil. https://eprints.hsr.ch/536/1/thesis_updatable_vector_tiles_from_openstreetmap.pdf.

Mau, Ronald. 2018. "Digitales Landschaftsmodell 1:250 000 Digitales Landschaftsmodell 1:1 000 000 Grundlagen, Erstellung, Aktualisierung Und Zukünftige Entwicklung." Dresden, September 1.

Mazúr, and Urbánek. 1983. "Space in Geography." *GeoJournal* 7 (2). https://doi.org/10.1007/BF00185159.

McKenzie, Grant, and Benjamin Adams. 2017. "Juxtaposing Thematic Regions Derived from Spatial and Platial User-Generated Content." Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany. https://doi.org/10.4230/lipics.cosit.2017.20.

McNeill, Graham, and Scott A. Hale. 2017. "Generating Tile Maps." *Computer Graphics Forum* 36 (3): 435–45. https://doi.org/10.1111/cgf.13200.

"Meta Tiles – OpenStreetMap Wiki." 2015. June 4, 2015. https://wiki.openstreetmap.org/wiki/Meta_tiles.

Mobasheri, Amin, Yeran Sun, Lukas Loos, and Ahmed Ali. 2017. "Are Crowdsourced Datasets Suitable for Specialized Routing Services? Case Study of OpenStreetMap for Routing of People with Limited Mobility." *Sustainability* 9 (6): 997. https://doi.org/10.3390/su9060997.
"Move to OSM2VT-Developed Cartography · Issue #387 · Osm2vectortiles/Osm2vectortiles." 2016. GitHub. 2016. https://github.com/osm2vectortiles/osm2vectortiles/issues/387.
Nah, Fiona Fui-Hoon. 2004. "A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait?" *Behaviour & Information Technology* 23 (3): 153–63. https://doi.org/10.1080/01449290410001669914.
Nordan, Robert Patrick Victor. 2012. "An Investigation of Potential Methods for Topology Preservation in Interactive Vector Tile Map Applications." Master's Thesis, Institutt for bygg, anlegg og transport. http://hdl.handle.net/11250/232116.
OGC. 2000. "OpenGIS® Web Map Server Interface Implementation Specification." https://portal.ogc.org/files/?artifact_id=7196.
———. 2006. "OpenGIS® Web Map Server Implementation Specification." https://portal.ogc.org/files/?artifact_id=14416.
———. 2010. "OpenGIS® Web Map Tile Service Implementation Standard." https://portal.ogc.org/files/?artifact_id=35326.
———. 2014. "OGC OWS Context Conceptual Model." https://portal.ogc.org/files/?artifact_id=55182.
———. (2019) 2021. "OGC API - Tiles." December 4, 2021. https://github.com/opengeospatial/ogcapi-tiles.
Olteanu-Raimond, Ana-Maria, Glen Hart, Giles M. Foody, Guillaume Touya, Tobias Kellenberger, and Demetris Demetriou. 2017. "The Scale of VGI in Map Production: A Perspective on European National Mapping Agencies: VGI and the National Mapping Agencies." *Transactions in GIS* 21 (1): 74–90. https://doi.org/10.1111/tgis.12189.
OpenStreetMap Wiki. 2019. "TMS – OpenStreetMap Wiki." TMS. 2019. https://wiki.openstreetmap.org/w/index.php?title=TMS&oldid=1894410.
———. 2021a. "Slippy Map Tilenames – OpenStreetMap Wiki." Slippy Map Tilenames. 2021. https://wiki.openstreetmap.org/w/index.php?title=Slippy_map_tilenames&oldid=2189832.
———. 2021b. "Tiles and Tiling/Setting up WMS or TMS – OpenStreetMap Wiki." Tiles and Tiling/Setting up WMS or TMS. 2021. https://wiki.openstreetmap.org/w/index.php?title=Tiles_and_tiling/Setting_up_WMS_or_TMS&oldid=2125228.
———. 2022a. "EPSG:3857 – OpenStreetMap Wiki." EPSG:3857. January 1, 2022. https://wiki.openstreetmap.org/w/index.php?title=EPSG:3857&oldid=2239091.

———. 2022b. “OpenMapTiles – OpenStreetMap Wiki.” OpenMapTiles – OpenStreetMap Wiki. February 7, 2022. https://wiki.openstreetmap.org/w/index.php?title=OpenMapTiles&oldid=2261733.

———. 2022c. “Standard Tile Layer – OpenStreetMap Wiki.” Standard Tile Layer – OpenStreetMap Wiki. March 16, 2022. https://wiki.openstreetmap.org/w/index.php?title=Standard_tile_layer&oldid=2289339.

OSGeo. 2012a. “Tile Map Service Specification - OSGeo.” Tile Map Service Specification. 2012. https://wiki.osgeo.org/w/index.php?title=Tile_Map_Service_Specification&oldid=62686.

———. 2012b. “WMS Tile Caching - OSGeo.” WMS Tile Caching. December 24, 2012. https://wiki.osgeo.org/w/index.php?title=WMS_Tile_Caching&oldid=68003.

“Osm2pgsql Manual - Osm2pgsql.” n.d. Accessed March 31, 2022. https://osm2pgsql.org/doc/manual.html.

“Osmosis – OpenStreetMap Wiki.” 2022. January 5, 2022. https://wiki.openstreetmap.org/w/index.php?title=Osmosis&oldid=2240690.

Pahwa, Aroon, Brian A. Lenoski, and Mathieu Courtemanche. 2015. Dynamic vector map tiles. US9798926B2, filed September 2, 2014, and issued September 1, 2015. https://patents.google.com/patent/US9798926B2/en.

Pippig, Karsten, Sebastian Denier, Dominik Käuferle, and Olaf Forte. 2022. “Towards a Next Generation of Swiss Maps.” *Abstracts of the ICA* 5 (September): 1–2. https://doi.org/10.5194/ica-abs-5-143-2022.

“Planet.Osm – OpenStreetMap Wiki.” 2022. March 16, 2022. https://wiki.openstreetmap.org/w/index.php?title=Planet.osm&oldid=2289644.

“Planet.Osm/Diffs – OpenStreetMap Wiki.” 2021. March 12, 2021. https://wiki.openstreetmap.org/w/index.php?title=Planet.osm/diffs&oldid=2124991.

Pohanka, Tomáš. 2022. “OpenMapTiles 3.14 - Vector Tiles from OpenStreetMap & Natural Earth Data.” Presented at the FOSS4G 2022, Firenze, August 26. https://talks.osgeo.org/foss4g-2022/talk/RAUZWM/.

QGIS. 2022. “OGC Certification.” OGC Certification. June 7, 2022. https://qgis.org/en/site/getinvolved/daily_reports/ogc.html.

Raifer, Martin, Troilo, Rafael, Mocnik, Franz-Benjamin, and Schott, Moritz. 2021. “OSHDB - OpenStreetMap History Data Analysis.” Zenodo. https://doi.org/10.5281/ZENODO.4146990.

Ramm, Frederik, and Jochen Topf. 2011. *OpenStreetMap: using and enhancing the free map of the world*. Cambridge: UIT Cambridge.

Rasmussen, Jens Eilstrup, Lars Eilstrup Rasmussen, Bret Steven Taylor, James Christopher Norris, Stephen Ma, Andrew Robert Kirmse, Noel Phillip Gordon, and Seth Michael LaForge. 2004. Digital mapping system. US8014946B2, issued 2004.

Resch, Bernd, Florian Usländer, and Clemens Havas. 2018. "Combining Machine-Learning Topic Models and Spatiotemporal Analysis of Social Media Data for Disaster Footprint and Damage Assessment." *Cartography and Geographic Information Science* 45 (4): 362–76. https://doi.org/10.1080/15230406.2017.1356242.

Rönneberg, Mikko, Mari Laakso, and Tapani Sarjakoski. 2019. "Map Gretel: Social Map Service Supporting a National Mapping Agency in Data Collection." *Journal of Geographical Systems* 21 (1): 43–59. https://doi.org/10.1007/s10109-018-0288-z.

Roth, Robert E., Cynthia A. Brewer, and Michael S. Stryker. 2011. "A Typology of Operators for Maintaining Legible Map Designs at Multiple Scales." *Cartographic Perspectives*, no. 68 (March): 29–64. https://doi.org/10.14714/CP68.7.

Růžička, Jan. 2016. "Comparing Speed of Web Map Service with GeoServer on ESRI Shapefile and PostGIS." *Geoinformatics FCE CTU* 15 (1): 3–9. https://doi.org/10.14311/gi.15.1.1.

Rybarczyk, Greg, Syagnik Banerjee, Melissa D. Starking-Szymanski, and Richard R. Shaker. 2018. "Travel and Us: The Impact of Mode Share on Sentiment Using Geo-Social Media and GIS." *Journal of Location Based Services* 12 (1): 40–62. https://doi.org/10.1080/17489725.2018.1468039.

Saman Bemel-Benrud, Thomas Christopher MacWright, Eden Halperin, Lauren Budorick, Qian Gao, and Scott Andrews. 2017. Modifying style layer properties of a digital map, issued 2017.

Samet, Hanan. 1984. "The Quadtree and Related Hierarchical Data Structures." *ACM Comput. Surv.* 16 (2): 187–260. https://doi.org/10.1145/356924.356930.

Samsonov, Timofey. 2011. "Multiscale Hypsometric Mapping." In *Advances in Cartography and GIScience. Volume 1: Selection from ICC 2011, Paris*, edited by Anne Ruas, 497–520. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19143-5_28.

Samsonov, Timofey, Anton Podolsky, and Nina Yurova. 2013. "Multimapper – Prototype System for Designing Multi-Scale Maps." In . Dresden. https://icaci.org/files/documents/ICC_proceedings/ICC2013/_extendedAbstract/219_proceeding.pdf.

Schaick, Jeroen van. 2010. "Future Scenarios for the Relation between Advanced Tracking Research and Urban Design and Planning." *Journal of Location Based Services* 4 (2): 70–92. https://doi.org/10.1080/17489725.2010.506663.

Scheider, Simon, and Krzysztof Janowicz. 2014. "Place Reference Systems: A Constructive Activity Model of Reference to Places." *Applied Ontology* 9 (2): 97–127. https://doi.org/10.3233/AO-140134.

Schiller, Jochen, and Agnès Voisard. 2004. *Location-Based Services*. Amsterdam: Elsevier. http://slubdd.de/katalog?TN_libero_mab213704380.

Schmidt, Christopher. 2007. "Google Projection: 900913." *Technical Ramblings* (blog). August 7, 2007. https://crschmidt.net/blog/archives/243/google-projection-900913/.

Scholz, Stefan, Paul Knight, Melanie Eckle, Sabrina Marx, and Alexander Zipf. 2018. "Volunteered Geographic Information for Disaster Risk Reduction—The Missing Maps Approach and Its Potential within the Red Cross and Red Crescent Movement." *Remote Sensing* 10 (8): 1239. https://doi.org/10.3390/rs10081239.

Schroeck, Michael, Rebecca Shockley, Janet Smart, Dolores Romero-Morales, and Peter Tufano. 2012. "Analytics: The Real-World Use of Big Data." Somers: IBM Global Services.

Schumann, Heidrun, and Wolfgang Müller. 2000. *Visualisierung: Grundlagen und allgemeine Methoden*. Berlin Heidelberg: Springer.

Schwartz, Joe. 2022. "Bing Maps Tile System - Bing Maps." Bing Maps Tile System. June 8, 2022. https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system.

Seifert, Markus. 2021. "Das Smartmapping Projekt Der AdV." Dresden, March 9. https://dresden.dgfk.net/Veranstaltung/vortrag-das-smartmapping-projekt-der-adv/.

Shea, K. Stuart. 1988. "Cartographic Generalization." Technical Report NOS 127 CGS 12. NOAA Technical Report. Reston, VA: NOAA. https://www.ngs.noaa.gov/PUBS_LIB/Cartographic_Generalization_TR_NOS127_CGS12.pdf.

"Slippy Map Tilenames – OpenStreetMap Wiki." n.d. Accessed March 1, 2019. https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames.

Slocum, Terry A., Rober B. McMaster, Fritz C. Kessler, and Hugh H. Howard. 2009. *Thematic Cartography and Geovisualization*. 3rd ed. Prentice Hall Series in Geographic Information Science. Upper Saddle River, NJ: Pearson Prentice Hall.

Snyder, John P. 1987. "Map Projections: A Working Manual." Report 1395. Professional Paper. Washington, D.C. USGS Publications Warehouse. https://doi.org/10.3133/pp1395.

Spence, Ian, and Stephan Lewandowsky. 1991. "Displaying Proportions and Percentages." *Applied Cognitive Psychology* 5 (1): 61–77. https://doi.org/10.1002/acp.2350050106.

"Sprite | Style Specification | Mapbox GL JS." n.d. Mapbox. Accessed January 18, 2022. https://docs.mapbox.com/mapbox-gl-js/style-spec/sprite/.

Srnka, Erhardt. 1970. "The Analytical Solution of Regular Generalization on Cartography." In *Internationales Jahrbuch Für Kartographie*, 48–62. Internationales Jahrbuch Für Kartographie, X. Güterloh: Kartographisches Institut Bertelsmann.

Stanislawski, Lawrence V., Barbara P. Buttenfield, Pia Bereuter, Sandro Savino, and Cynthia A. Brewer. 2014. "Generalisation Operators." In *Abstracting Geographic Information in a Data Rich World*, edited by Dirk Burghardt, Cécile Duchêne, and William

Mackaness, 157–95. Lecture Notes in Geoinformation and Cartography. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-00203-3_6.

Sui, Daniel Z., Sarah Elwood, and Michael F. Goodchild, eds. 2013. *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*. Dordrecht; New York: Springer.

Thomas, James J., and Kristin A. Cook, eds. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. Los Alamitos, Calif: IEEE Computer Soc.

Thompson, Mackenzie Blake, and Ryan Joseph Clark. 2018. Efficient processing for vector tile generation. US10921136, issued 2018.

Töpfer, Friedrich. 1974. *Kartographische Generalisierung*. Gotha: Hermann Haack Geographisch-Kartographische Anstalt. http://slubdd.de/katalog?TN_libero_mab21163026.

Touya, Guillaume, Justin Berli, Imran Lokhat, and Nicolas Regnauld. 2017. "Experiments to Distribute and Parallelize Map Generalization Processes." *The Cartographic Journal* 54 (4): 322–32. https://doi.org/10.1080/00087041.2017.1413787.

Touya, Guillaume, and Andreas Reimer. 2015. "Inferring the Scale of OpenStreetMap Features." In *OpenStreetMap in GIScience*, edited by Jamal Jokar Arsanjani, Alexander Zipf, Peter Mooney, and Marco Helbich, 81–99. Lecture Notes in Geoinformation and Cartography. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-14280-7_5.

Tsorlini, Angeliki, René Sieber, Lorenz Hurni, Hubert Klauser, and Thomas Gloor. 2017. "Visualising Statistical Data on Thematic Maps with the Help of a Rule-Based Wizard." *Cartographic Perspectives* 0 (86). https://doi.org/10.14714/CP86.1392.

Tsou, Ming-Hsiang. 2009. "Recent Developments in Internet GIS." *Geospatial World* (blog). September 10, 2009. https://www.geospatialworld.net/article/recent-developments-in-internet-gis/.

———. 2015. "Research Challenges and Opportunities in Mapping Social Media and Big Data." *Cartography and Geographic Information Science* 42 (sup1): 70–74. https://doi.org/10.1080/15230406.2015.1059251.

Tsou, Ming-Hsiang, Jiue-An Yang, Daniel Lusher, Su Han, Brian Spitzberg, Jean Mark Gawron, Dipak Gupta, and Li An. 2013. "Mapping Social Activities and Concepts with Social Media (Twitter) and Web Search Engines (Yahoo and Bing): A Case Study in 2012 US Presidential Election." *Cartography and Geographic Information Science* 40 (4): 337–48. https://doi.org/10.1080/15230406.2013.799738.

Tuan, Yi-Fu. 1977. *Space and Place: The Perspective of Experience*. Minneapolis: University of Minnesota press.

Veenendaal, Bert, Maria Antonia Brovelli, and Songnian Li. 2017. "Review of Web Mapping: Eras, Trends and Directions." *ISPRS International Journal of Geo-Information* 6 (10): 317. https://doi.org/10.3390/ijgi6100317.

Visvalingam, Maheswari, and James Duncan Whyatt. 1992. *Line Generalisation by Repeated Elimination of the Smallest Area*. https://hull-repository.worktribe.com/output/459275.

W3C. 2001. "Scalable Vector Graphics (SVG) 1.0 Specification." 2001. https://www.w3.org/TR/SVG10/.

Wagner, Daniel, Alexander Zipf, and Rene Westerholt. 2020. "Place in the GIScience Community – an Indicative and Preliminary Systematic Literature Review," January. https://doi.org/10.5281/ZENODO.3628855.

Wang, Zheye, and Xinyue Ye. 2019. "Space, Time, and Situational Awareness in Natural Hazards: A Case Study of Hurricane Sandy with Social Media Data." *Cartography and Geographic Information Science* 46 (4): 334–46. https://doi.org/10.1080/15230406.2018.1483740.

Westerholt, René, Franz-Benjamin Mocnik, and Alexis Comber. 2020. "A Place for Place: Modelling and Analysing Platial Representations." *Transactions in GIS* 24 (4): 811–18. https://doi.org/10.1111/tgis.12647.

Wider, Thomas, Damien Palacio, and Ross S. Purves. 2013. "Georeferencing Images Using Tags: Application with Flickr." In *AGILE International Conference on Geographic Information Science*. http://www.agile-online.org/Conference_Paper/CDs/agile_2013/Short_Papers/SP_S2.3_Palacio.pdf.

Wikipedia. 2020. "Normbasierte Austauschschnittstelle — Wikipedia, Die Freie Enzyklopädie." 2020. https://de.wikipedia.org/w/index.php?title=Normbasierte_Austauschschnittstelle&oldid=198037518.

———. 2021a. "Tiled Web Map." Wikipedia. March 23, 2021. https://en.wikipedia.org/w/index.php?title=Tiled_web_map&oldid=1013818035.

———. 2021b. "Web Mercator Projection." Wikipedia. October 28, 2021. https://en.wikipedia.org/w/index.php?title=Web_Mercator_projection&oldid=1052364617.

———. 2021c. "Smartphone." Wikipedia. November 29, 2021. https://en.wikipedia.org/w/index.php?title=Smartphone&oldid=1057681111.

———. 2022. "Painter's Algorithm." Wikipedia. 2022. https://en.wikipedia.org/w/index.php?title=Painter%27s_algorithm&oldid=1124511104.

Winter, Stephan, and Christian Freksa. 2012. "Approaching the Notion of Place by Contrast." *Journal of Spatial Information Science*, no. 5 (December): 31–50. https://doi.org/10.5311/JOSIS.2012.5.90.

Zeileis, Achim, Jason C. Fisher, Kurt Hornik, Ross Ihaka, Claire D. McWhite, Paul Murrell, Reto Stauffer, and Claus O. Wilke. 2019.

"Colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes." *ArXiv:1903.06490 [Cs, Stat]*, March. http://arxiv.org/abs/1903.06490.

Zeileis, Achim, Kurt Hornik, and Paul Murrell. 2009. "Escaping RGBland: Selecting Colors for Statistical Graphics." *Computational Statistics & Data Analysis* 53 (9): 3259–70. https://doi.org/10.1016/j.csda.2008.11.033.

Zhao, Xiaoyun. 2015. "On Processing GPS Tracking Data of Spatio-Temporal Car Movements: A Case Study." *Journal of Location Based Services* 9 (4): 235–53. https://doi.org/10.1080/17489725.2015.1098738.

Zheng, Yu. 2011. "Location-Based Social Networks: Users." In *Computing with Spatial Trajectories*, edited by Yu Zheng and Xiaofang Zhou, 243–76. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-1629-6_8.

Zhou, Xiaolu, and Liang Zhang. 2016. "Crowdsourcing Functions of the Living City from Twitter and Foursquare Data." *Cartography and Geographic Information Science* 43 (5): 393–404. https://doi.org/10.1080/15230406.2015.1128852.

"Zoom Levels – OpenStreetMap Wiki." 2019. 2019. https://wiki.openstreetmap.org/wiki/Zoom_levels.

9 Appendix

9.1 Zoom levels and Scale

Table A9: Zoom levels for raster and vector tiles, with the resolution (meter per pixel) and rounded scale at the Equator according to OpenStreetMap Wiki.[59]

Zoom levels raster tiles [256 x 256 px]	Zoom levels vector tiles [512 x 512 px]	m / px at equator	Rounded scale at equator
0		156,412	500,000,000
1	0	78,206	250,000,000
2	1	39,103	150,000,000
3	2	19,551	70,000,000
4	3	9,776	35,000,000
5	4	4,888	15,000,000
6	5	2,444	10,000,000
7	6	1,222	4,000,000
8	7	611	2,000,000
9	8	305	1,000,000
10	9	153	500,000
11	10	76	250,000
12	11	38	150,000
13	12	19	70,000
14	13	10	35,000
15	14	5	15,000
16	15	2	8,000
17	16	1	4,000
18	17	0.6	2,000
19	18	0.3	1,000

[59] https://wiki.openstreetmap.org/w/index.php?title=Zoom_levels&oldid=1895097

9.3 Full information about selected UGC papers

Table A10: Overview of selected UGC papers with maps.

Year	Journal	Paper
2008	JLBS	Girardin et al., “Leveraging Explicitly Disclosed Location Information to Understand Tourist Dynamics.”
2010	JLBS	Andrienko et al., “A Framework for Using Self-Organising Maps to Analyse Spatio-Temporal Patterns, Exemplified by Analysis of Mobile Phone Usage.”
2010	TCJ	Cheng et al., “Multi-Scale Visualisation of Inbound and Outbound Traffic Delays in London.”
2010	JLBS	Guo, Liu, and Jin, “A Graph-Based Approach to Vehicle Trajectory Analysis.”
2010	TCJ	Haklay et al., “How Many Volunteers Does It Take to Map an Area Well?”
2010	JLBS	van Schaick, “Future Scenarios for the Relation between Advanced Tracking Research and Urban Design and Planning.”
2011	CaGIS	Kessler, “Volunteered Geographic Information.”
2013	CaGIS	Crampton et al., “Beyond the Geotag.”
2013	CaGIS	Fujita, “Geo-Tagged Twitter Collection and Visualization System.”
2013	CaGIS	Ghosh and Guha, “What Are We ‘Tweeting’ about Obesity?”
2013	CaGIS	Kent and Capello, “Spatial Patterns and Demographic Indicators of Effective Social Media Content during The Horsethief Canyon Fire of 2012.”
2013	JLBS	Krumm, Gruen, and Delling, “From Destination Prediction to Route Prediction.”
2013	CaGIS	Li, Goodchild, and Xu, “Spatial, Temporal, and Socioeconomic Patterns in the Use of Twitter and Flickr.”
2013	JLBS	Mooney, Rehrl, and Hochmair, “Action and Interaction in Volunteered Geographic Information.”
2013	JLBS	Patel, “Incorporating Duration and Region Association Information in Trajectory Classification.”
2013	JLBS	Pippig, Burghardt, and Prechtel, “Semantic Similarity Analysis of User-Generated Content for Theme-Based Route Planning.”
2013	CaGIS	Stefanidis et al., “Demarcating New Boundaries.”
2013	CaGIS	Tsou et al., “Mapping Social Activities and Concepts with Social Media (Twitter) and Web Search Engines (Yahoo and Bing).”
2013	CaGIS	Xu, Wong, and Yang, “Evaluating the ‘Geographical Awareness’ of Individuals.”
2014	CaGIS	Camponovo and Freundschuh, “Assessing Uncertainty in VGI for Emergency Response.”
2014	JLBS	Gröchenig, Brunauer, and Rehrl, “Digging into the History of VGI Data-Sets.”
2014	CaGIS	Hawelka et al., “Geo-Located Twitter as Proxy for Global Mobility Patterns.”
2014	CaGIS	Sagl, Delmelle, and Delmelle, “Mapping Collective Human Activity in an Urban Environment Based on Mobile Phone Data.”
2014	JLBS	You and Krumm, “Transit Tomography Using Probabilistic Time Geography.”
2015	JLBS	Gonçalves, Afonso, and Martins, “Cartographic Visualization of Human Trajectory Data.”
2015	CaGIS	Malleson and Andresen, “The Impact of Using Social Media Data in Crime Rate Calculations.”
2015	JLBS	Tessem et al., “Word Cloud Visualisation of Locative Information.”
2015	JLBS	Zhao, “On Processing GPS Tracking Data of Spatio-Temporal Car Movements.”

2016	CaGIS	Gong et al., "Inferring Trip Purposes and Uncovering Travel Patterns from Taxi Trajectory Data."
2016	TCJ	Hauthal and Burghardt, "Mapping Space-Related Emotions out of User-Generated Photo Metadata Considering Grammatical Issues."
2016	JLBS	Oliveira et al., "Gazetteer Enrichment for Addressing Urban Areas."
2016	JLBS	Qiu and Wang, "Inferring Road Maps from Sparsely Sampled GPS Traces."
2016	CaGIS	Robertson and Feick, "Bumps and Bruises in the Digital Skins of Cities."
2016	CaGIS	Shook and Turner, "The Socio-Environmental Data Explorer (SEDE)."
2016	CaGIS	Zhou and Zhang, "Crowdsourcing Functions of the Living City from Twitter and Foursquare Data."
2017	CaGIS	Baker et al., "Crowdsourcing a Cyclist Perspective on Suggested Recreational Paths in Real-World Networks."
2017	CaGIS	John et al., "Deriving Incline Values for Street Networks from Voluntarily Collected GPS Traces."
2018	JLBS	Almaslukh, Magdy, and Rey, "Spatio-Temporal Analysis of Meta-Data Semantics of Market Shares over Large Public Geosocial Media Data."
2018	CaGIS	Kounadi et al., "Population at Risk."
2018	JLBS	Mamei and Colonna, "Analysis of Tourist Classification from Cellular Network Data."
2018	CaGIS	Resch, Usländer, and Havas, "Combining Machine-Learning Topic Models and Spatiotemporal Analysis of Social Media Data for Disaster Footprint and Damage Assessment."
2018	JLBS	Rybarczyk et al., "Travel and Us."
2019	JLBS	Çay, Nagel, and Yantaç, "What Is Happening in the City?"
2019	CaGIS	Jiang, Li, and Ye, "Understanding Demographic and Socioeconomic Biases of Geotagged Twitter Users at the County Level."
2019	JLBS	Keler and Krisp, "Understanding the Relationship between Complicated Crossings and Frequently Visited Locations – a Case Study with Boro Taxis in Manhattan."
2019	CaGIS	Koylu et al., "CarSenToGram."
2019	JLBS	Viel et al., "An Original Approach to Positioning with Cellular Fingerprints Based on Decision Tree Ensembles."
2019	CaGIS	Wang and Ye, "Space, Time, and Situational Awareness in Natural Hazards."
2020	CaGIS	Gröbe and Burghardt, "Micro Diagrams."
2020	CaGIS	Hu, Li, and Ye, "Delineating and Modeling Activity Space Using Geotagged Social Media Data."
2020	JLBS	Kveladze and Agerholm, "GeoVisual Analytics for Understanding the Distribution of Speeding Patterns on Arterial Roads."

Journal of Location Based Services (JLBS)
The Cartographic Journal (TCJ)
Cartography and Geographic Information Science (CaGIS)

9.4 Timeline of mapping technologies

Table A11: Overview of mapping technologies and their year of introduction.

Name	Year	Source
WMS	2000	https://portal.ogc.org/files/?artifact_id=7196
Raster Tiles	2005	https://cfis.savagexi.com/2006/05/03/google-maps-deconstructed/
WMST	2010	https://www.ogc.org/standards/wmts
Vektor Tiles	2014	https://github.com/mapbox/vector-tile-spec/tree/master/1.0.0
TMS	2006	https://en.wikipedia.org/w/index.php?title=Tile_Map_Service&oldid=1026187764
OGC API Tiles	2022	https://github.com/opengeospatial/ogcapi-tiles

9.5 Timeline of map providers

Table A12: Timeline of map providers.

Company	Start	End	Source
MapQuest	1994		https://en.wikipedia.org/w/index.php?ti-tle=MapQuest&oldid=1059991486
GoogleEarth	2001		https://en.wikipedia.org/w/index.php?ti-tle=Google_Earth&oldid=1058170806
Yahoo! Maps	2002	2015	https://web.archive.org/web/20080713214826/http://yhoo.cli-ent.shareholder.com/press/timeline.cfm
OpenStreetMap	2005		https://en.wikipedia.org/w/index.php?title=Open-StreetMap&oldid=1056217878
Google Maps	2005		https://en.wikipedia.org/w/index.php?ti-tle=Google_Maps&oldid=1058425513
Bing	2005		https://en.wikipedia.org/w/index.php?ti-tle=Bing_Maps&oldid=1055988490
CloudMade	2007	2015	https://wiki.openstreetmap.org/w/index.php?title=Cloud-Made&oldid=2076236
Thunderforest	2008		https://www.thunderforest.com/about/
Mapbox	2010		https://en.wikipedia.org/w/index.php?title=Map-box&oldid=1043203114
Humanitarian OSM Team	2010		https://wiki.openstreetmap.org/w/index.php?title=Humanitar-ian_OSM_Team&oldid=2199189
Apple Maps	2012		https://en.wikipedia.org/w/index.php?title=Ap-ple_Maps&oldid=1056830468
Mapzen	2013	2018	https://en.wikipedia.org/w/index.php?ti-tle=Mapzen&oldid=1056386905
HERE We Go	2013		https://en.wikipedia.org/w/index.php?ti-tle=Here_WeGo&oldid=1056698851
OpenMapTiles	2016		Martinelli and Roth, "Updatable Vector Tiles from OpenStreetMap." 2016

9.6 Code snippets from own map production workflows

9.6.1 Vector tiles workflow

```
local tables = {}
local mySchema = 'import'

tables.places = osm2pgsql.define_node_table('places', {
    { column = 'name',     type = 'text' },
    { column = 'type',     type = 'text' },
    { column = 'population', type = 'text' },
    { column = 'capital', type = 'text' },
    { column = 'geom',     type = 'point' },
}
, {schema = mySchema}
)
tables.peaks = osm2pgsql.define_node_table('peaks', {
    { column = 'name',     type = 'text' },
    { column = 'name_en',     type = 'text' },
    { column = 'type',     type = 'text' },
    { column = 'ele', type = 'text' },
    { column = 'geom',     type = 'point', projection = 4326 },
}, {schema = mySchema})

function osm2pgsql.process_node(object)

    if object.tags.place then
        tables.places:add_row({
            name = object.tags.name,
            type = object.tags.place,
            population = object.tags.population,
            capital = object.tags.capital,
        })
    end

    if object.tags.natural == 'peak' or object.tags.natural == 'vulcano' then
        tables.peaks:add_row({
            name = object.tags.name,
            name_en = object.tags['name:en'],
            type = object.tags.natural,
            ele = object.tags.ele,
        })
    end
```

Figure A78: Content of the configuration file for osm2psql that imports places and peaks from OpenStreetMap tags to defined database tables.

```
ogr2ogr PG:"dbname=multi-scale-demo user=postgres password=Dell2300" -clipsrc -180 -85 180 85
-t_srs EPSG:3857  natural_earth_vector.sqlite ne_110m_lakes -nln ne_110m_lakes -nlt
PROMOTE_TO_MULTI -lco SCHEMA=naturalearth
ogr2ogr PG:"dbname=multi-scale-demo user=postgres password=Dell2300" -clipsrc -180 -85 180 85
-t_srs EPSG:3857  natural_earth_vector.sqlite ne_110m_land -nln ne_110m_land -nlt
PROMOTE_TO_MULTI -lco SCHEMA=naturalearth
ogr2ogr PG:"dbname=multi-scale-demo user=postgres password=Dell2300" -clipsrc -180 -85 180 85
-t_srs EPSG:3857  natural_earth_vector.sqlite ne_110m_rivers_lake_centerlines -nln
ne_110m_rivers_lake_centerlines -nlt PROMOTE_TO_MULTI -lco SCHEMA=naturalearth
ogr2ogr PG:"dbname=multi-scale-demo user=postgres password=Dell2300" -clipsrc -180 -85 180 85
-t_srs EPSG:3857  natural_earth_vector.sqlite ne_110m_admin_0_boundary_lines_land -nln
ne_110m_admin_0_boundary_lines_land -nlt PROMOTE_TO_MULTI -lco SCHEMA=naturalearth
ogr2ogr PG:"dbname=multi-scale-demo user=postgres password=Dell2300" -clipsrc -180 -85 180 85
-t_srs EPSG:3857  natural_earth_vector.sqlite  ne_110m_populated_places  -nln
ne_110m_populated_places -lco SCHEMA=naturalearth
```

Figure A77: Import of the NaturalEarth data into the rendering database using org2ogr from GDAL.

```
[[tileset.layer]]
name = "lakes"
geometry_field = "geometry"
geometry_type = "POLYGON"
srid = 3857
buffer_size = 10
simplify = true
  [[tileset.layer.query]]
  minzoom = 0
  sql = """SELECT geometry FROM naturalearth.ne_110m_lakes WHERE geometry && !bbox! AND
  scalerank = 0"""
  [[tileset.layer.query]]
  minzoom = 3
  sql = """SELECT geometry FROM naturalearth.ne_50m_lakes WHERE geometry && !bbox! AND
  scalerank <= 3"""
  [[tileset.layer.query]]
  minzoom = 5
  sql = """SELECT geometry FROM naturalearth.ne_10m_lakes WHERE geometry && !bbox! AND
  scalerank <= 5"""
```

Figure A79: Definition of the lakes layer in the vector tiles tile set for t-rex. Several queries define the content of the different zoom levels according to attributes. The “!bbox!” variable is replaced during tile generation with the extent of a tile, while “&&” compares it with the geometry of a column, limiting the results.

```
[[tileset.layer.query]]
minzoom = 8
sql = """SELECT name, type, round(population)::float as population, geom AS geometry,
round(population_isolation)::float AS isolation, CASE WHEN capital = 'yes' THEN 'yes' ELSE
'no' END AS capital
FROM vector.places
WHERE (population_isolation > 30000 AND type = 'city' AND geom && !bbox!)
OR (population_isolation > 40000 AND type = 'town' AND geom && !bbox!)
ORDER BY population DESC """
```

Figure A80: The more complex query for a specific zoom level for t-rex. The selected columns in the query are available as attributes in the vector tiles.

```
version:                      8
name:                         "multi-scale-bw1-demo"
▼ metadata:
    mapbox:autocomposite:     false
    maputnik:renderer:        "mbgljs"
▼ sources:
  ▼ multi-scale-demo:
      type:                   "vector"
      url:                    "http://127.0.0.1/data/multi-scale-demo.json"
glyphs:                       "http://127.0.0.1/fonts/{fontstack}/{range}.pbf"
▼ layers:
  ▼ 0:
      id:                     "background_"
      type:                   "background"
    ▼ paint:
        background-color:     "rgba(255, 255, 255, 1)"
  ▼ 1:
      id:                     "land"
      type:                   "fill"
      source:                 "multi-scale-demo"
      source-layer:           "land"
    ▼ paint:
        fill-color:           "rgba(242, 242, 242, 1)"
        fill-antialias:       false
  ▼ 2:
      id:                     "rivers"
      type:                   "line"
      source:                 "multi-scale-demo"
      source-layer:           "rivers"
    ▼ paint:
        line-color:           "rgba(255, 255, 255, 1)"
```

Figure A81: Excerpt of the style file (StyleJSON) for rendering vector tiles, referencing tiles and glyphs for the example.

```
tiles:
  0:              "http://localhost/data/multi-scale-demo/{z}/{x}/{y}.pbf"
  1:              "http://127.0.0.1/data/multi-scale-demo/{z}/{x}/{y}.pbf"
name:             "multi-scale-demo"
format:           "pbf"
basename:         "multi-scale-demo"
id:               "multi-scale-demo"
attribution:      "(c) OpenStreetMap contributors | NaturalEarth"
bounds:
  0:              -180
  1:              -90
  2:              180
  3:              90
center:
  0:              0
  1:              0
  2:              2
description:      "multi-scale-demo"
Layer:
  0:
    description:  ""
    fields:
      capital:    ""
      isolation:  ""
      name:       ""
      population: ""
      type:       ""
    id:           "cities"
    name:         "cities"
    properties:
      buffer-size: 0
      maxzoom:    8
      minzoom:    0
  1:
    description:  ""
    fields:       {}
    id:           "rivers"
    name:         "rivers"
```


Figure A82: Excerpt of the tile set description (TileJSON) for the example workflow describing the tile set extent and layer with their attribute fields.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>multi-scale-bw1-demo - TileServer GL</title>
  <link rel="stylesheet" type="text/css" href="/mapbox-gl.css" />
  <script src="/mapbox-gl.js"></script>
  <link rel="stylesheet" type="text/css" href="/mapbox.css" />
  <script src="/mapbox.js"></script>
  <script src="/leaflet-hash.js"></script>
  <style>
    body { margin:0; padding:0; }
    #map { position:absolute; top:0; bottom:0; width:100%; }
  </style>
</head>
<body>
  <h1 style="display:none;">multi-scale-bw1-demo</h1>
  <div id='map'></div>
  <script>
    var q = (location.search || '').substr(1).split('&');
    var preference =
      q.indexOf('vector') >= 0 ? 'vector' :
        (q.indexOf('raster') >= 0 ? 'raster' :
          (mapboxgl.supported() ? 'vector' : 'raster'));
    if (preference == 'vector') {
      mapboxgl.setRTLTextPlugin('/mapbox-gl-rtl-text.js');
      var map = new mapboxgl.Map({
        container: 'map',
        style: '/styles/multi-scale-bw1-demo/style.json',
        hash: true
      });
      map.addControl(new mapboxgl.NavigationControl());
    } else {
      var map = L.mapbox.map('map', '/styles/multi-scale-bw1-demo.json', { zoomControl: false });
      new L.Control.Zoom({ position: 'topright' }).addTo(map);
      setTimeout(function() {
        new L.Hash(map);
      }, 0);
    }
  </script>
</body>
</html>
```

Figure A83: Source code of the viewer for the vector tiles example, referencing the style and setting up the viewer.

9.6.2 Raster tiles workflow

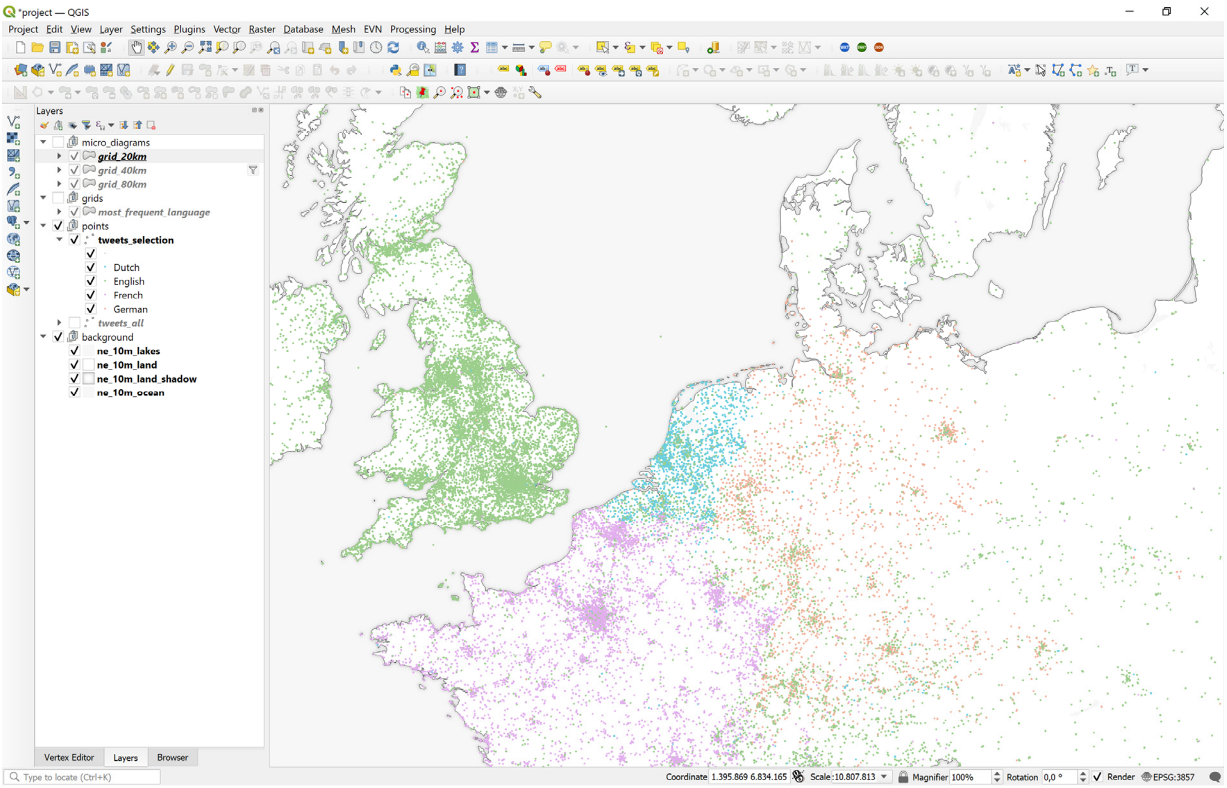


Figure A84: QGIS project for tweet visualization. All project layers are available as layers in the QGIS Server WMS later on and can be referenced.

```
sources:
  micro_diagrams:
    type: wms
    wms_opts:
      version: 1.3.0
    req:
      url: http://172.28.1.2/service?
      layers: micro_diagrams
      transparent: true
    coverage:
      bbox: [-27.34,26.4,50.44,67.23]
      srs: 'EPSG:4326'

  tweets_selection:
    type: wms
    wms_opts:
      version: 1.3.0
    req:
      url: http://172.28.1.2/service?
      layers: tweets_selection
      transparent: true
    coverage:
      bbox: [-27.34,26.4,50.44,67.23]
      srs: 'EPSG:4326'
```

Figure A85: Excerpt of the MapProxy configuration referencing layers from the QGIS project as a WMS from the QGIS server.

```yaml
services:
  demo:
  # XYZ/TMS needs a cache!
  tms:
    use_grid_names: false
    # origin for /tiles service, no access via /TMS!
    origin: 'nw'
  wms:
    md:
      title: Micro diagram multi scale demo
      abstract: Map Preview
  wmts:
    kvp: false
    restful: true

layers:
  - name: micro_diagram
    title: Micro diagram map
    sources: [micro_diagrams_cache]
  - name: tweet_selection
    title: Point map
    sources: [tweets_selection_cache]
  - name: background
    title: Background map
    sources: [background_cache]

caches:
  micro_diagrams_cache:
    grids: ['GLOBAL_WEBMERCATOR']
    sources: [micro_diagrams]
    format: my_format
    cache:
      type: sqlite

  tweets_selection_cache:
    grids: ['GLOBAL_WEBMERCATOR']
    sources: [tweets_selection]
    format: my_format
    cache:
      type: sqlite
```

Figure A87: Configuration of the caches and services of MapProxy for the different layers.

```html
<body>
<div id="map"></div>
<script>

    var background = L.tileLayer('http://localhost/wmts/background/GLOBAL_WEBMERCATOR/{z}/{x}/{y}.jpeg', {attribution:
    "Made with data from NaturalEarth and Twitter"});
    var osm = L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {attribution: "&copy; <a
    href='https://www.openstreetmap.org/copyright'>OpenStreetMap</a> contributors and Twitter"});

    var diagrams = L.tileLayer.wms('http://localhost/service?', {layers: 'micro_diagram', format: "image/png",
    transparent: true})
    //var diagrams = L.tileLayer('http://localhost/tiles/micro_diagram_EPSG3857/{z}/{x}/{y}.png');
    var points = L.tileLayer.wms('http://localhost/service?', {layers: 'tweet_selection', format: "image/png",
    transparent: true})
    //var points = L.tileLayer('http://localhost/tiles/tweet_selection_EPSG3857/{z}/{x}/{y}.png');

    var map = L.map('map',{ center:[49,2], zoom:5,  layers: [osm]});
    L.control.scale({position: 'bottomleft'}).addTo(map);

    var baseMaps = {
            "Background": background,
            "OpenStreetMap": osm};
    var overlayMaps = {
            "Point map": points,
            "Micro diagram map": diagrams};
    var layerControl = L.control.layers(baseMaps,overlayMaps).addTo(map);

</script>
</body>
</html>
```

Figure A86: Viewer for the raster tiles referencing services from MapProxy. It is possible to use XYZ and WMS to access the layers with the same result and comparable performance.