

Improving the content of vario-scale maps

Mihai-Alexandru Erbaşu

Student #5130956

m.a.erbasu@student.tudelft.nl

1st supervisor: Martijn Meijers

2nd supervisor: Peter van Oosterom

June 16th, 2021

1 Introduction

Maps have had a significant historical role in humanity's journey (Figure 1), as they were utilised as tools in wars, conquering operations, new territory discovery, trade and countless other military or civilian actions which have influenced our history (Friedman, 2013). From the outset, maps have portrayed a vision of the world in accordance to the way we perceived our surroundings: from Ptolemy's world map, which depicted solely Europe and a crudely drawn version of North-Africa and parts of Asia, all the way to the complex maps that we use today.

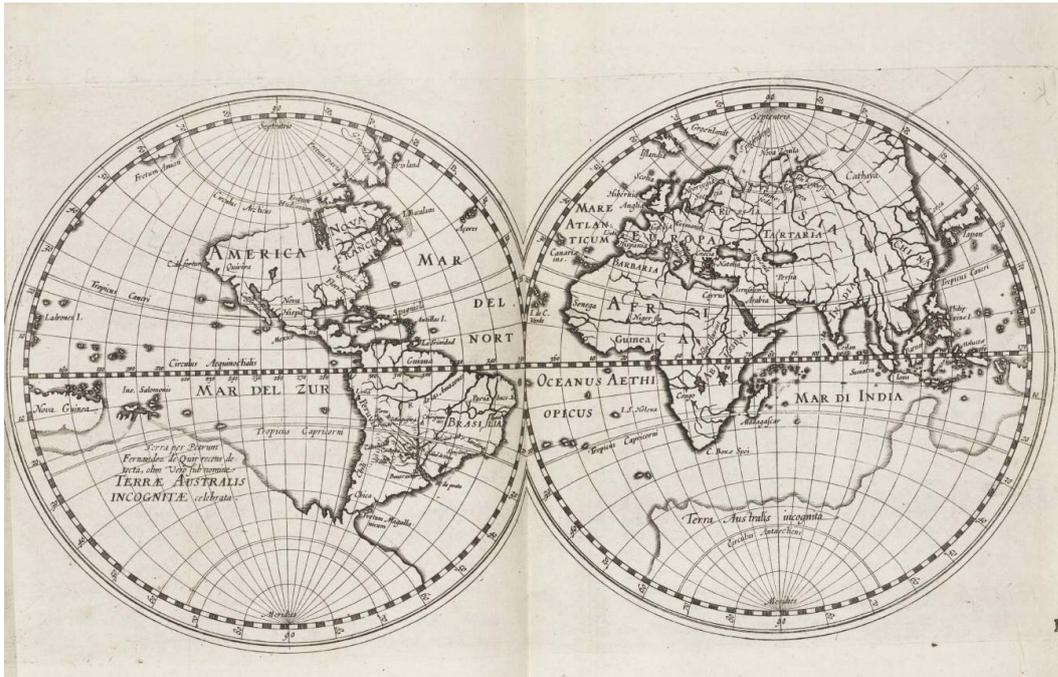


Figure 1: Example of a historical map with incomplete regions (Maps of Net, 2021)

Naturally, maps have organically evolved alongside all other societal and technological advancements (Dempsey, 2011). Nowadays, a wide variety of digital maps are available day and night at our fingertips, portraying all sorts of useful information, ranging from paramount topics such as demographics and geo-politics to displays of entertainment and leisure. Maps tell stories, display continuously changing information, help compile and find patterns in terabytes of data and perform complex analysis (Esri, 2015).

Of course, it would be nearly impossible and certainly impractical to create a map which displays a certain part of the world in its true size. This is why, since the early-developed maps, the concept of scale has been introduced, which defines the ratio between the distance that is displayed on the map and its true distance on the field (Geoscience Australia, 2021). When creating a map with a specific purpose in mind, choosing an adequate scale at which the information that needs to be disseminated can be properly visualized used to be an important requirement. However, in this day and age, as computers and smartphones became more widespread and integrated in all aspects of the society, the demands for how the scale should function have shifted as well.

Modern, web-based mapping solutions, such as Google Maps, Bing Maps or even MapBox

and OpenStreetMap utilize a concept called multi-scale mapping, where a fixed number of zoom levels is used to display the information (Open Street Map, 2019; MapBox, 2021; Antin Harasymiv, 2018). While this may seem like a smoothly-working solution when using the aforementioned platforms, that is just a "illusion", as the concept itself uses "a number of redundant and fixed map scale representations" (Chair of GIS technology, 2021), while a connection between these different scales is missing.

This is where vario-scale maps come into play: these have the property that, when a certain amount of change occurs in the map scale, this results in an equal change in the map representation (van Oosterom and Meijers, 2014). Thus, the discrete way of visualising only a limited number of scales is replaced by a continuous solution.

Vario-scale maps have, as a starting point, a vector base map (which is considered to be the largest scale available), completely filled with with polygonal objects (i.e. there are no gaps on the surface), which form a planar partition (Meijers et al., 2018). On this 2D base map, a third dimension is added, which represents the scale. Various generalization operations can be performed in parallel, such as integrating a certain area element into another based on a least-important element/most-compatible neighbour function, splitting the area across its neighbours and area to line collapse, amongst a few others (Meijers et al., 2018). All of these operations can be conceptually represented using a topological generalized area partitioning (tGAP) tree (van Oosterom, 2005) (Figure 2). Upon performing these generalizations, the content which is displayed by the map becomes simpler, as less data is shown when visualizing a larger area at a smaller scale.

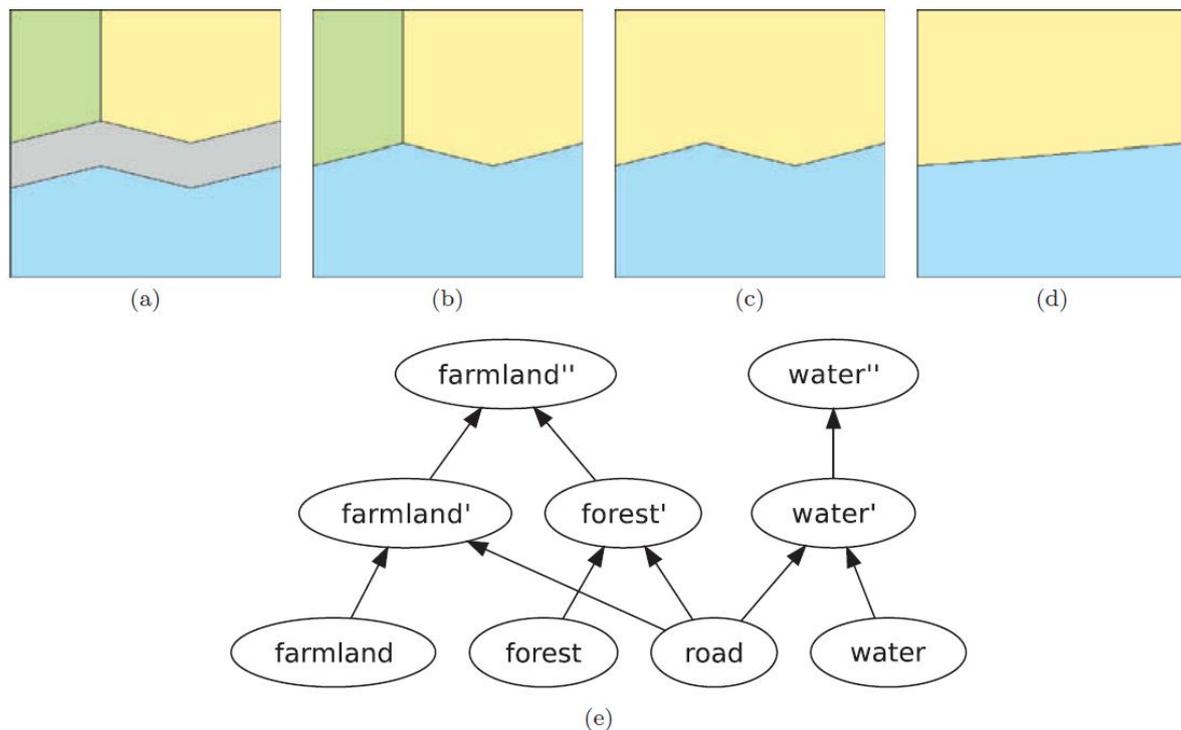


Figure 2: The four map fragments and corresponding tGAP structure: (a) original map, (b) result of collapse, (c) result of merge, (d) result of simplify, (e) corresponding tGAP structure. (van Oosterom and Meijers, 2014)

These transformations, performed continuously in accordance to the scale change, form a Space-Scale Cube (Figure 3) data structure (van Oosterom and Meijers, 2014). In the same source, it is described that selecting a visualisation is as simple as horizontally slicing the cube at the desired scale height, while also giving the possibility to create more complex slices (such as fish-eye, or diagonal slice).

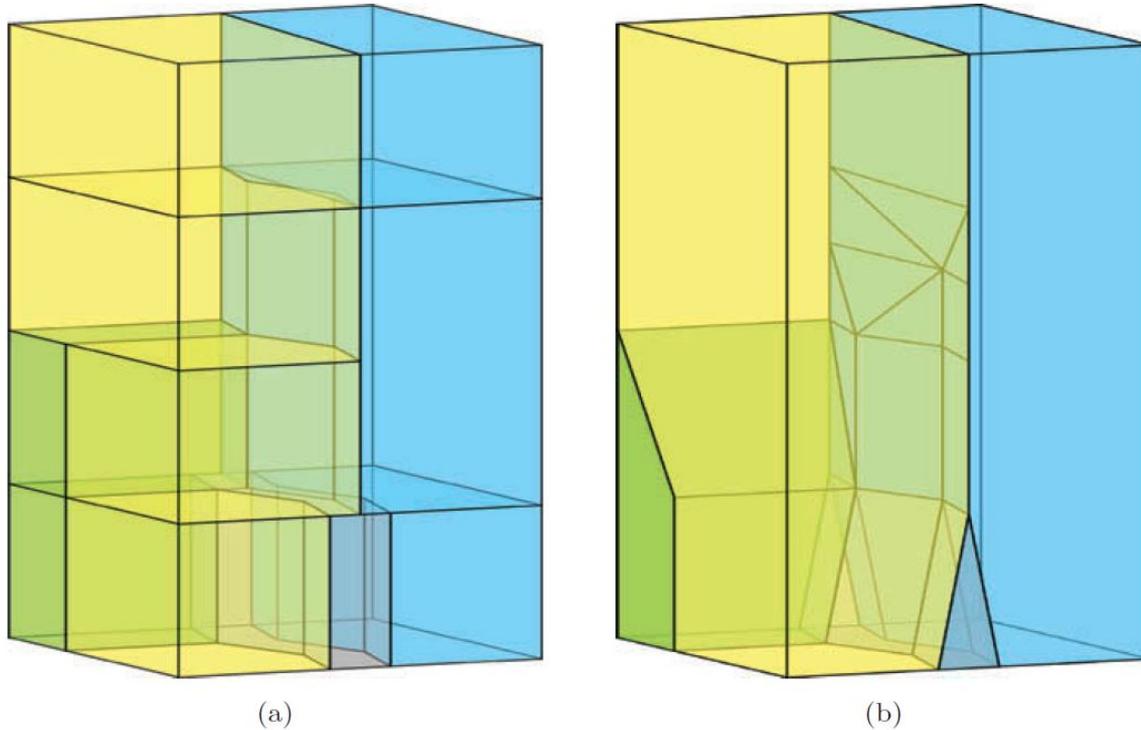


Figure 3: The space-scale cube (SSC) representation in 3D: (a) SSC for the classic tGAP structure, resulting in a collection of stacked prisms, (b) SSC for the smooth tGAP structure, resulting in a collection of arbitrarily shaped polyhedra. (van Oosterom and Meijers, 2014)

However, this topic is still a research in progress. There are certain aspects which need to be both better defined and implemented. One such example is line generalization, which is one of the possible operations that can be applied during scale change, in order to improve the overall result of the vario-scale generalization, thus creating a better solution for the users. This graduation project will research the various already-existing solutions related to line generalization, taking into account both the cartographic and the computer-graphics perspective, while trying to come up with a suitable solution for creating a good result which maintains the topological constraints (Meijers and van Oosterom, 2011).

2 Related work

2.1 Vario-scale maps

The chair of GIS Technology from the Delft University of Technology has been developing the concept of vario-scale maps for a while now, and with each iteration, the overall results gets improved. The main data structure, the tGAP, is explained in depth by van Oosterom (2005), alongside its components (the GAP-tree for the faces and the GAP-forest for the edges), as

well as an introduction to how these can be determined, constructed and implemented in a database.

Another structure, which has been developed alongside the tGAP is the Space Scale Cube (SSC) (Meijers and van Oosterom, 2011). The term, analogous with the "space-time cube", introduced by Hägerstrand (1970), conceptualizes the scale as the third dimension. Both the SSC and the tGAP are the backbone of the vario-scale concept, and represent the main way of generating such a map. A thorough analysis on both of them can be found in the paper by van Oosterom and Meijers (2014).

The functioning principle of the aforementioned data structures is that, at any arbitrary change in the zoom level of the map, one or more generalization processes are being performed. The previously mentioned paper offers an introduction into these operation. Peng et al. (2020a) explore a solution on how multiple merging generalization operations can be performed in parallel, thus creating a more seamless and smoother zoom in/out experience for the user.

On the other hand, Meijers et al. (2018) offer a framework (called the Scale Dependend Framework), which can be used as a blueprint when developing a non hard-coded solution, while at the same time offering the user the freedom to modify various importances related to the different generalization operations, i.e. to adjust which particular operations are more relevant for the overall result, tailored for varying scale intervals.

When it comes to the development side of the story, Peng et al. (2020b) offer a broad perspective on how one of the platforms which implements a vario-scale solution is developed. This will constitute a good outline for the practical part of the project. Also relevant for this part of the project is the paper by Meijers et al. (2020), which discusses how the SSC is obtained and subsequently rendered by using the GPU effectively, as well as the way it can be sent from the server to the client side through a technique which creates chunks of data.

Last, but not least, it is worth mentioning the work done by previous Geomatics students for the Synthesis project: Boersma et al. (2018), which discusses the use-case of integrating vario-scale maps in the workflow of the Kadaster in such a way that it better meets the requirements of the end-users which work with topographic data provided by the organization. Also worth noting is the article by Šuba et al. (2016), which goes in-depth into the generalization of road networks. Since streets are often represented using line segments, further analysis into the way their simplification should be performed can prove very useful for this graduation project.

2.2 Line Generalization

In the first relevant paper mentioned in the previous chapter (van Oosterom, 2005), the concept of Binary Line Generalization (BLG) is also presented. This is based on the line generalization theory introduced by Douglas and Peucker (1973), which discusses in depth one of the most important line generalization solutions, the Douglas-Peucker algorithm (Figure 4). Thus, for the graduation thesis itself, this solution will constitute a fundamental brick for the construction of a well-balanced and fully functioning line generalization mechanism.

While Douglas-Peucker is very efficient when it comes to the removal of redundant details and, in general, for data compression, it does have certain shortcomings, namely that it can result in spikes and extreme sharp angles (Esri, 2020). There are, of course, plenty of other

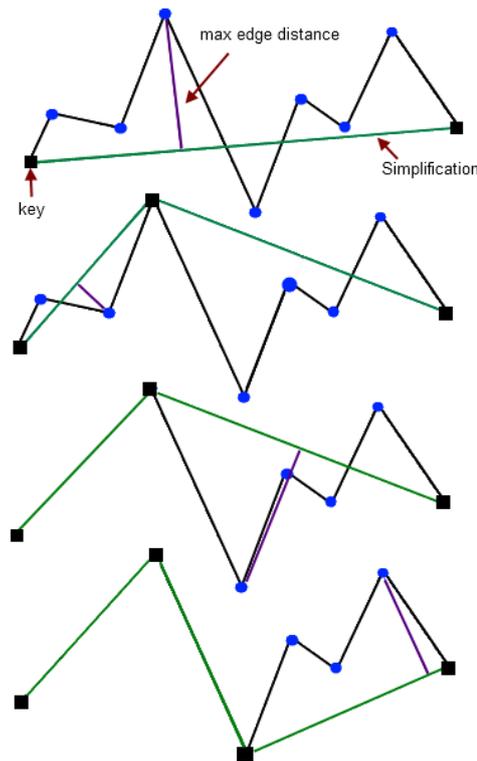


Figure 4: Functioning principle of the Douglas-Peucker algorithm (de Koning, 2011)

alternatives, such as the Wang-Müller algorithm (Wang and Müller, 1998), which comes in handy when line bends are more relevant, as the algorithm detects and removes the insignificant ones.

Other two line generalization worth mentioning are the Zhou-Jones (Zhou and Jones, 2005) and the Visvalingam-Whyatt (Visvalingam and Whyatt, 1992), both working under the principle of analysing the triangle which is formed from three adjacent points on the line. The analysis itself is, however, different.

All of these can be categorised under the simplification branch of the generalization algorithms, which work by continuously removing, each using a varying logic, points from the polyline (a visualisation of this concept was implemented by Bostock (2012)). However, another aspect of line generalization is constituted by line smoothing algorithms, which shift the position of vertices on the line in such a way that it creates a smooth border animation with the scale change (Union of Geography Professors, 2000). There are a few different algorithms which are worth mentioning, such as McMaster's Distance Weighting Algorithm (McMaster and Shea, 1992) or Chaikin's Smoothing Algorithm (Chaikin, 1974). Further insight into this topic is discussed in the paper by Mansouryar and Hedayati (2012). As it can be seen in Figure 5, applying various smoothing algorithms can result in lines with a much more aesthetic appearance, which in term is more suited for smaller scales.

An introduction into the usage of line simplification in a vario-scale context is presented by Meijers (2011). The algorithm which is introduced in this paper is applied after another simplification operation is being performed, and it works in such a way that the topology is being kept intact. The principles which are being presented represent an excellent foundation

for the dissertation project.



Figure 5: Line Smoothing example (McMaster and Shea, 1992)

3 Research Question

Vario-scale maps, in their current form, have certain shortcomings which need to be addressed in order to perfect and improve them. This graduation project will take a critical look at the line simplification and smoothing generalization operations, and will try to answer the following research question:

How can we include line-generalization algorithms in the vario-scale structure such that it preserves the topology, enables an optimal line density and allows for sufficient data transfer at any scale?

As this research question encompasses a broad range of issues, further categorised sub-questions should be asked in order to guide the scientific process towards a final goal.

Line generalization theoretical:

Which line generalization algorithms are better suited for which particular situations?
What is the most suitable way of combining said algorithms such that it upholds the technical requirements?

Implementation related issues:

What are the conditions and the implementation requirements necessary for maintaining topological correctness at any scale?
What is the optimal number of operations that need to be performed per scale change such that the line/vertices density remains constant?
How can the scale transition be performed in a smooth manner?
What is the best way, from the point of view of time and size complexity (i.e. Big O notation concepts (Kuredjian, 2017)), to perform line generalization in particular and vario-scale operations in general?

Overall result

What metrics can be constructed for evaluating the processing result?

How should the various generalization operations be tuned such that the overall result is useful (i.e. it validates the metrics)?

Does the result work for different types of datasets?

It should also be mentioned that, due to time constraints, other generalization operations will not be researched in-depth, but only taken into account as part of the entire system (Section 4.1 gives further explanations).

4 Methodology

The workflow, which also constitutes the Methodology principles which this graduation project will follow, is presented in Figure 6. It should be noted that, the foundation step (not displayed in the aforementioned figure) is composed by the analysis of the issue which needs to be solved and the creation of a proper strategy, in the form of well defined researched questions, which in term will aid to guide the entire process.

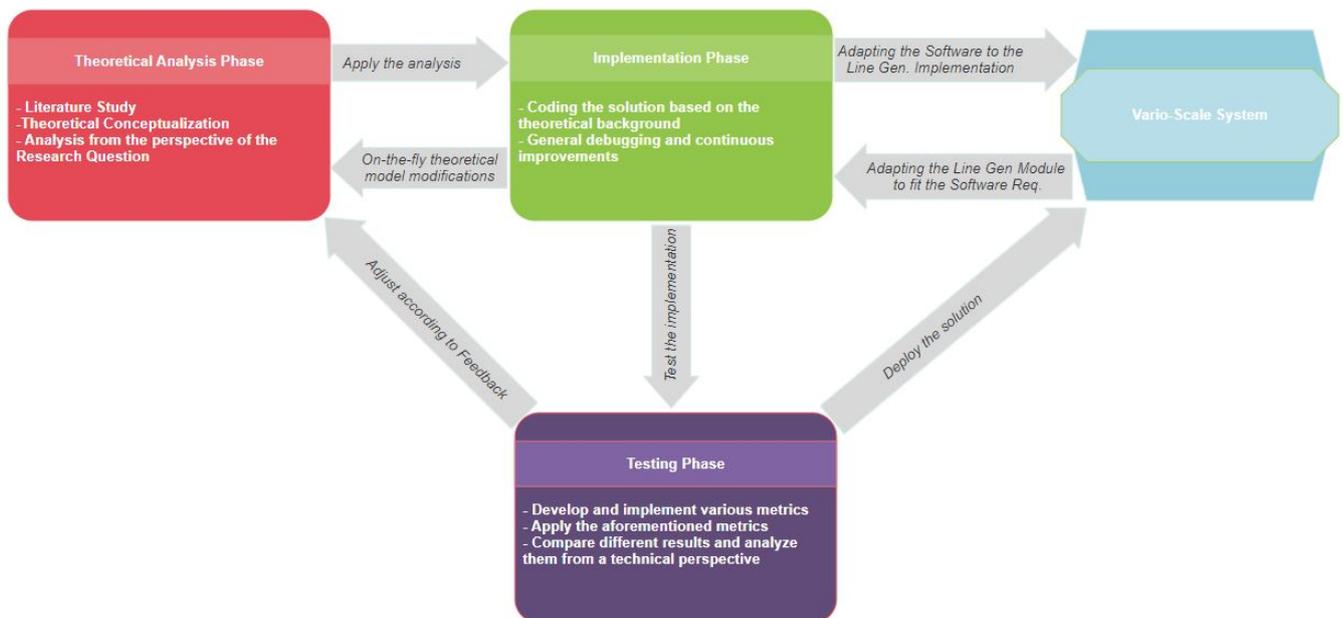


Figure 6: Project workflow

All of the phases are performed in a cyclical manner, following the Agile development principle, which is often used in software development projects due to its adaptability and flexibility to changes. In opposition to the waterfall model, in which tasks are performed successively, and where a phase is completely finished before moving on to the next one, Agile Methodology works in sprints, where all project phases are being performed more or less at the same time (Figure 7). Thus, going back and forth between different phases throughout the duration of the project will be entirely possible. Further explanation on how this will work in practice and why it is important is given in the following sub-sections.

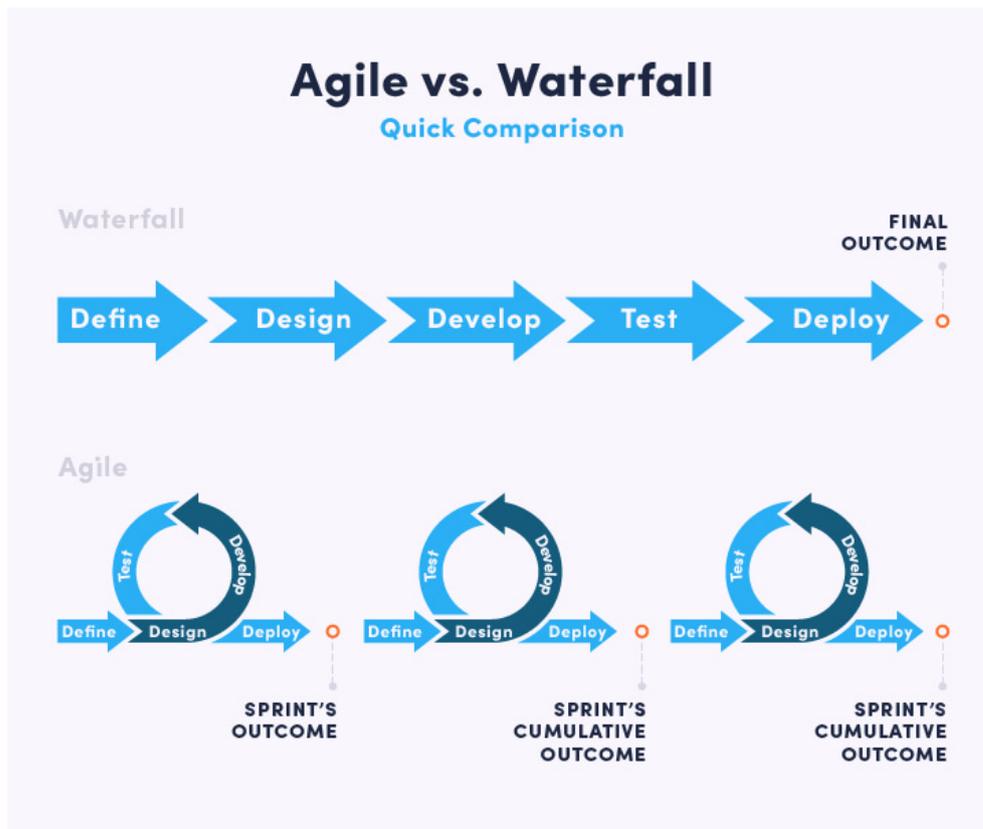


Figure 7: Comparison between the Waterfall and Agile Methodologies (Pawlicka, 2021)

4.1 Theoretical Analysis Phase

At this stage, as the name suggests, the theoretical analysis is being performed, mainly through literature study. There are plenty of resources available, such as papers and articles, some of which have already been mentioned in Section 2. Understanding what contribution each of these sources bring to the vario-scale systems, while at the same time trying to put them all together like pieces of a jigsaw puzzle will be a challenge.

In vario-scale maps, all components have an individual importance, yet they cannot function in solitary. Thus, even though this graduation thesis has as main focus the way line generalization operations function in such a system, understanding and keeping in mind the other generalization operations will be just as important.

All of this research then needs to be combined and conceptualized, with the purpose of creating a comprehensive theoretical background which is to be used when coding the solution. Of course, this model would not be set in stone, so, if needed, it would be possible to go back to this step with the experience gained from either the Implementation or the Testing phase. It could be the case that certain issues arise while working on the solution itself, so returning to this phase with the experience accumulated in the practical part of this assignment could prove very useful when developing the final solution.

Certain aspects which need to be thoroughly analyzed at this point include: which line simplification or smoothing algorithms should be applied in which situations, how can these be combined amongst each other as well as amongst the other operations which occur, at which steps should these actions be performed (such as at the tGAP formation or the SSC extraction

and visualization).

The final result of this phase would be a pseudo-code portraying all the operations which a vario-scale algorithm should perform, alongside a schema portraying the various components and their functioning principle.

4.2 Implementation Phase

This is the practical side of the assignment, where presumably most of the time and effort will be spent on. At this point, based on the theoretical analysis performed in the previous phase, a solution will be coded.

An important aspect of this phase is constituted by the relationship of my solution with the already-existing vario-scale system (illustrated in Figure 6 as the diamond-shaped element on the right hand side), which is a code base developed by the chair of GIS Technology (Meijers and Peng, 2021). This contains an already-existing solution for the creation of vario-scale maps, with the remark that the module related to line generalization is currently very basic. Developing a module which can be directly plugged into this system could be one way to go.

On the other hand, the paper by Boersma et al. (2018) does provide an alternative to the way certain elements are implemented in the legacy code-base (for example, the way the SSC is extracted). Thus, this phase will also involve some technical decisions which need to be made and a thorough analysis on which practices work best for achieving the final result.

Implementing different possible line generalization alternatives, or creating a modular and adjustable solution could lead in the end to an overall better result. And, of course, as it is the case with any software development project, debugging will most likely be a constant process throughout this phase.

4.2.1 Development Tools

Looking at the big picture, the main development endeavours will be performed using the Python language for creating the back-end, with JavaScript (JS) as the main front-end language. Alongside JS, WebGL will be utilized as the principal Application Programming Interface (API) for graphical representation, as it allows the creation of highly-interactive 2D and 3D graphics on a web client without requiring any plug-ins (MDN Web Docs, 2021).

For storing the data, as well as the generated tGAP and SSC, a Postgres database, which has been extended with PostGIS is required. For visualisation and debugging purposes, QGIS and ArcGIS can be utilized.

4.3 Testing Phase

In this phase, the result of the entire development process will be tested against a series of well crafted metrics, which should give a good picture of how well the implemented algorithms have performed, in accordance to certain parameters.

At this point, it should be noted that this project will not focus on the perspective of the user of such maps, and will only consider the technological point of view. While understanding

the end user is an interesting research perspective, it is beyond the scope of this thesis. Thus, the testing phase will be constituted solely on the technical perspective.

Whenever testing a certain piece of software, it is important to account for edge cases as well. In order to achieve this, different datasets will be utilized, so that a thorough investigation into how the software treats them can be conducted. For example, man-made structures like buildings or roads have much sharper shapes than natural elements like forests or rivers. Creating a "one-fits-all" solution can end up being a difficult undertaking, so analyzing which alternative works better for which situation is important.

At the end of the project, the testing results accumulated from the different implementation alternatives can be gathered together and compared amongst each other, to determine which was the most suited for the defined scope of the project.

4.3.1 Datasets

As mentioned previously, various datasets, containing different characteristics are required for performing a thorough analysis on the validity of the solution which has been crafted. For this purpose, the following datasets have been selected:

- **Testing datasets provided during the 2014 SIGSPATIAL GIS Competition** - simple datasets consisting out of a relatively small number of edges and vertices (which also contain a number of free-floating points - i.e. which are not part of any lines), representing various US states (such as New Hampshire (Figure 8) or Massachusetts (Figure 9)), which were utilized during the programming competition which was associated with the 22nd ACM SIGSPATIAL Conference on Advances in Geographic Information Systems (ACM Digital Library, 2021). Due to the overall simplicity of the datasets provided, as well as the range of different cases, this set should be very useful when it comes to the initial testing, as the changes can be easily observed.
- **TOP10NL - Limburg, TOP10NL - Drenthe and TOP10NL - Whole Country** - these are datasets provided by the Dutch Kadaster, consisting of object-oriented topographic files, which belong to the Basic Topographic Register (Basisregistratie Topografie) (Kadaster, 2021). It should be noted that these datasets have already been cleaned and made ready for being used in a vario-scale context, which makes them highly suitable as the main testing datasets for this project. The Limburg one is particularly useful, as it is the only set which contains building objects as well. Also worth mentioning is that the whole-country testing set might be a bit too large for any practical purposes, so working with the local ones will most likely be the best strategy.
- **Corine Land Cover 2006** - another pair of datasets, this time creating a pan-European classification of the land usage, which was created by the Copernicus Land Monitoring Service (Copernicus Programme, 2021) This has also been made ready for vario-scale usage, and will thus provide an alternative to the Dutch-only dataset in testing.

Further research into other possible datasets can be done at some point in time, but, in order to be used for the creation of a vario-scale map, the data needs to be in a certain format and to follow certain constraints. For this reason, and due to the time constraints imposed by the nature of this assignment, the aforementioned datasets will be the main ones to be used.

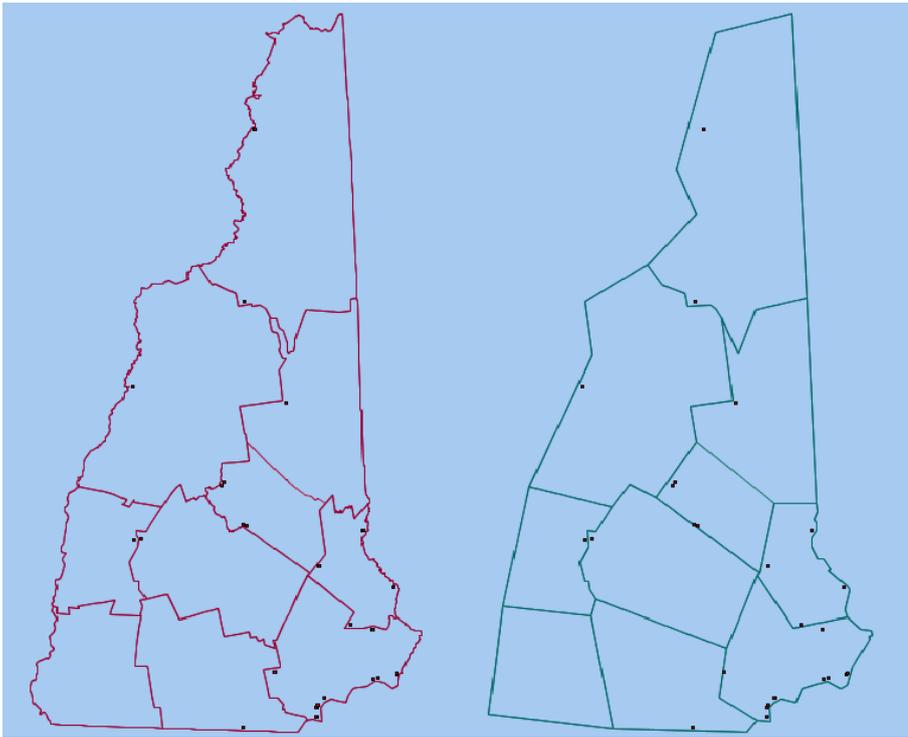


Figure 8: New Hampshire training data set - initial shape (left) and possible simplification (right) - certain separate points can also be observed. (ACM Digital Library, 2014)

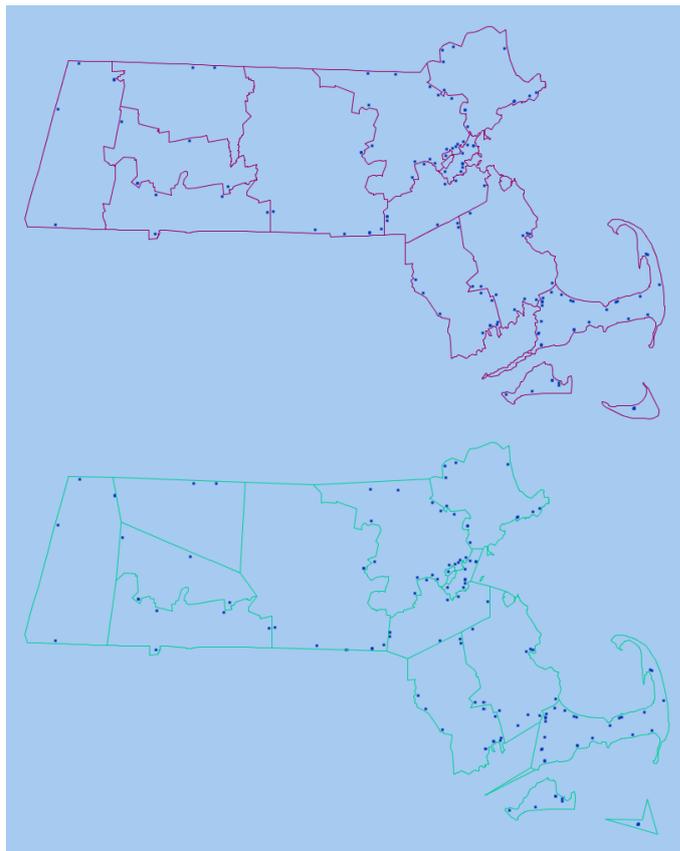


Figure 9: Massachusetts training data set - initial shape (left) and possible simplification (right) (ACM Digital Library, 2014)

5 Time planning

Figure 10 offers a general time perspective on how the graduation project will be carried out. The starting date was February 2021, and the expected completion date is mid-January 2022. There are a few things worth noting:

- The initial research part of the assignment has taken a bit longer than expected, and thus, there was a less than desired analysis of the practical aspects of the project. This, however, should not pose an issue in the big-picture, due to the fact that the Summer break months (second half of June, July and August) will also be used for working on the thesis, which overall represents around two and a half months worth of extra working time.
- The dates for P3, P4 and P5 respectively are tentative, as the precise dates have not yet been set.

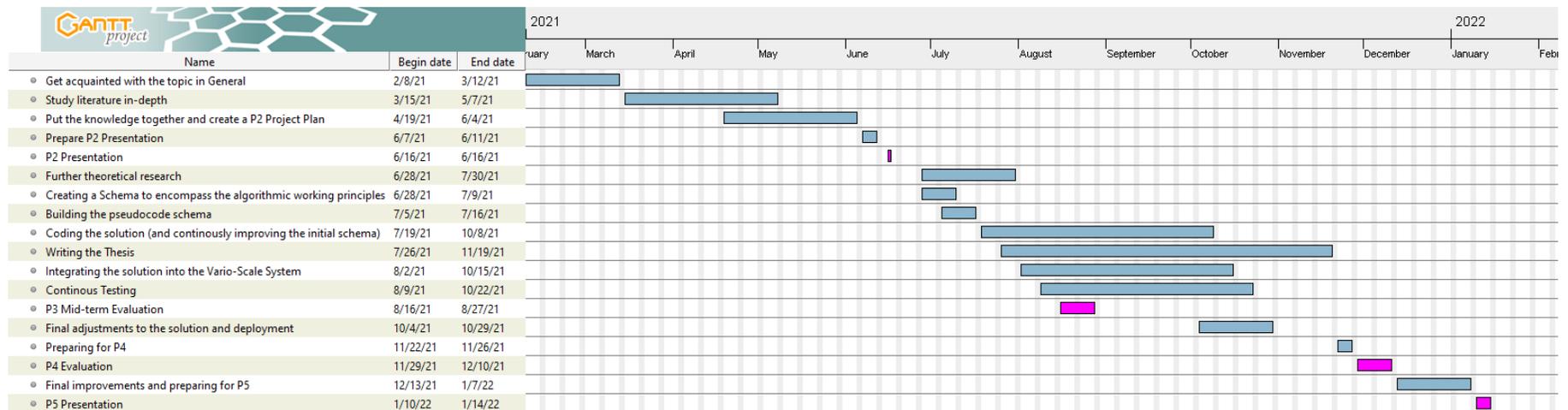


Figure 10: The Gantt Chart planification of the project

Meetings with the mentor team, which is formed by Martijn Meijers and Peter van Oosterom, will be organized every other week. These sessions will be used both for discussing the bi-weekly progress on the project and as a way to examine and debate the development direction which could be followed. In order to track the development, a progress monitor form, consisting out of a number of questions related to the way the project endeavor went, will be presented before every meeting. During this time, Meeting Minutes are being recorded.

References

- ACM Digital Library. Introduction of GIS CUP 2014, 2014. URL <http://mypages.iit.edu/~xzhang22/GISCUP2014/index.php>.
- ACM Digital Library. ACM SIGSPATIAL international conferences on advances in geographic information systems, 2021. URL <https://dl.acm.org/conference/gis>.
- Antin Harasymiv. Prototyping a smoother map - a glimpse into how google maps works, 2018. URL [https://medium.com/google-design/google-maps-cb0326d165f5#:~:text=Zoom%20Levels&text=At%20every%20incremental%20zoom%20level,map%20\(only%20more%20detailed\)](https://medium.com/google-design/google-maps-cb0326d165f5#:~:text=Zoom%20Levels&text=At%20every%20incremental%20zoom%20level,map%20(only%20more%20detailed)).
- F. Boersma, S. Du, P. Klaassen, and P. Ruben. GEO1101 project - improving vario-scale implementation based on needs of Kadaster topographic data users, 2018. URL <http://resolver.tudelft.nl/uuid:f2407da7-e85e-4f0e-92dc-990571a8a243>.
- M. Bostock. Line simplification, 2012. URL <https://bost.ocks.org/mike/simplify/>.
- G. M. Chaikin. An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3(4):346–349, 1974. ISSN 0146-664X. doi: 10.1016/0146-664X(74)90028-8.
- Chair of GIS technology. Vario-scale geo-information, 2021. URL <https://varioscale.bk.tudelft.nl/>.
- Copernicus Programme. Corine land cover, 2021. URL <https://land.copernicus.eu/pan-european/corine-land-cover>.
- E. de Koning. Douglas-Peucker, 2011. URL <http://psimpl.sourceforge.net/douglas-peucker.html>.
- C. Dempsey. Mapping through the ages: The history of cartography, 2011. URL <https://www.gislounge.com/mapping-through-the-ages/>.
- D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10:112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727.
- Esri. Six things that modern maps do, 2015. URL <https://www.esri.com/about/newsroom/insider/six-things-that-modern-maps-do/>.
- Esri. How simplify line and simplify polygon work, 2020. URL <https://pro.arcgis.com/en/pro-app/latest/tool-reference/cartography/how-simplify-line-works.htm>.
- U. Friedman. 12 maps that changed the world, 2013. URL theatlantic.com/international/archive/2013/12/12-maps-that-changed-the-world/282666/.
- Geoscience Australia. What is map scale?, 2021. URL <https://www.ga.gov.au/scientific-topics/national-location-information/topographic-maps-data/basics/what-is-map-scale#heading-1>.
- T. Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24:6–21, 1970. doi: 10.1007/BF01936872.
- Kadaster. Topnl, 2021. URL <https://www.kadaster.nl/zakelijk/producten/geo-informatie/topnl>.

- S. Kuredjian. Algorithm time complexity and big O notation, 2017. URL <https://medium.com/@StueyGK/algorithm-time-complexity-and-big-o-notation-51502e612b4d>.
- M. Mansouryar and A. Hedayati. Smoothing via iterative averaging (SIA) A basic technique for line smoothing. *International Journal of Computer and Electrical Engineering*, 4(3):307–311, 2012. ISSN 1793-8163. doi: 10.7763/IJCEE.2012.V4.501.
- MapBox. Zoom levels, 2021. URL <https://docs.mapbox.com/help/glossary/zoom-level/>.
- Maps of Net. World historical map, 2021. URL <https://www.mapsof.net/the-world/world-historical-map-1612>.
- R. McMaster and S. Shea. *Generalization in Digital Cartography Resource Publications in Geography*. Association of American Geographers, 1992. ISBN 8-89291-209-X.
- MDN Web Docs. WebGL: 2D and 3D graphics for the web, 2021. URL https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API.
- M. Meijers. *Advancing geoinformation science for a changing world*, chapter Simultaneous topologically-safe line simplification for a variable-scale planar partition, pages 337–358. *Lecture Notes in Geoinformation and Cartography*. Springer-Verlag Berlin Heidelberg, 2011. ISBN 978-3-642-26756-7.
- M. Meijers and D. Peng. sscview-js github page, 2021. URL <https://github.com/bmmeijers/sscview-js/tree/develop>.
- M. Meijers and P. van Oosterom. The space-scale cube: An integrated model for 2D polygonal areas and scale. *ISPRS - International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, XXXVIII-4/C21:95–102, Aug. 2011. doi: 10.5194/isprsarchives-XXXVIII-4-C21-95-2011.
- M. Meijers, P. van Oosterom, R. Šuba, and D. Peng. Towards a scale dependent framework for creating vario-scale maps. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4:425–432, Oct. 2018. doi: 10.5194/isprs-archives-XLII-4-425-2018.
- M. Meijers, P. van Oosterom, M. Driel, and R. Šuba. Web-based dissemination of continuously generalized Space-Scale Cube data for smooth user interaction. *International Journal of Cartography*, 6(1):152–176, 2020. doi: 10.1080/23729333.2019.1705144.
- Open Street Map. Zoom levels, 2019. URL https://wiki.openstreetmap.org/wiki/Zoom_levels.
- A. Pawlicka. Agile software development process – everything you need to know, 2021. URL <https://selleo.com/blog/agile-software-development-process-everything-you-need-to-know>.
- D. Peng, M. Meijers, and P. van Oosterom. Paralleling generalization operations to support smooth zooming: case study of merging area objects. 23rd ICA Workshop on Map Generalisation and Multiple Representation, pages 1–8, Nov. 2020a.
- D. Peng, M. Meijers, and P. van Oosterom. Multi-layer vario-scale web map comparer with dynamic transitions and visual analytical tool. Nov. 2020b.
- Union of Geography Professors. Line smoothing algorithms, 2000. URL <https://geogra.uah.es/patxi/gisweb/LGmodule/LGSmoothing.htm#:~:text=There%20are%20several%20line%20smoothing,as%20McMaster's%20Slide%20Averaging%20Algorithm>.

- P. van Oosterom. Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. *Cartography and Geographic Information Science*, 32(4):331–346, 2005. doi: 10.1559/152304005775194782.
- P. van Oosterom and M. Meijers. Vario-scale data structures supporting smooth zoom and progressive transfer of 2D and 3D data. *International Journal of Geographical Information Science*, 28(3):455–478, Mar. 2014. doi: 10.1080/13658816.2013.809724.
- M. Visvalingam and J. D. Whyatt. Line generalisation by repeated elimination of the smallest area. *Cartographic Information Systems Research Group (CISRG) Discussion Paper 10*, 1992.
- Z. Wang and J.-C. Müller. Line generalization based on analysis of shape characteristics. *Cartography and Geographic Information Systems*, 25(1):3–15, 1998. doi: 10.1559/152304098782441750.
- S. Zhou and C. Jones. Shape-aware line generalisation with weighted effective area. In *Developments in Spatial Data Handling*, pages 369–380. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26772-0. doi: 10.1007/3-540-26772-7_28.
- R. Šuba, M. Meijers, and P. van Oosterom. Continuous road network generalization throughout all scales. *International Journal of Geo-Information*, 5(8), Aug. 2016. ISSN 2220-9964. doi: 10.3390/ijgi5080145.