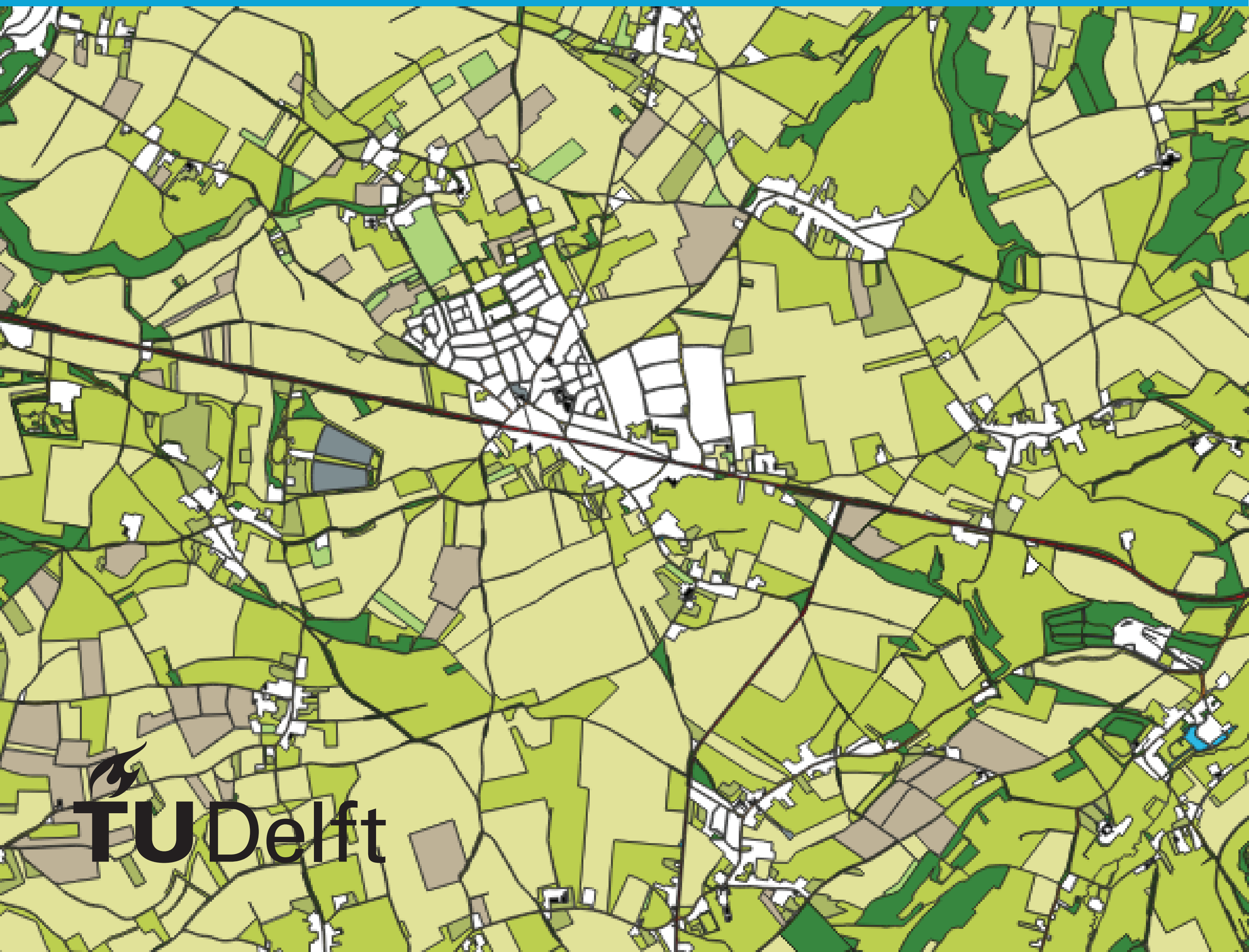
MSc thesis in Geomatics for the Built Environment

Parallel step assignment for continuous generalization constrained with target map.

Konrad Jarocki

2020



MSc thesis in Geomatics

Parallel step assignment for continuous generalization constrained with target map.

Konrad Jarocki

July 2020

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

Geo-Database Management Centre
Department of the OTB
Faculty of the Built Environment & Architecture
Delft University of Technology

Supervisors:	dr.ir. B.M. Meijers
	dr. Dongliang Peng
Co-reader:	prof.dr.ir. P.J.M. van Ooosterom

Abstract

Technological development changed the way maps are used. Digital representation makes them more interactive than ever before; even relatively low-priced smartphones are powerful enough to provide a truly interactive experience for zooming and panning maps while web viewers create a possibility for multi-scale representation. However, every operation requested by a user has impact on readability of data. Therefore, to keep readability on the same level for zooming, generalization of the map is necessary. However, besides a definition of generalization principles, the changes should be introduced to the user in a clear way to mare sure that they do not get confused if, after each zooming operation, they see a completely different map. This aspect has been addressed by many researchers many times, however, rapid progress in map generalization, especially development of smooth transition, makes the current solution relatively less attractive. The new idea requires more time to display each operation, which makes the current state-of-the-art approach too slow to be considered a good solution. Since the solution shows changes one-by-one, a natural conclusion is that to limit the display time, some of them should be shown simultaneously. In order to face that problem, a new solution for generalization operation presentation is proposed.

In this thesis a new method to show the generalization process is proposed and assessed - parallel step assignment. It investigates the feasibility and applicability of this method with respect to three proposed generalization approaches and tries to evaluate the possibility of vario-scale map representation. First, the generalization sequence is created for each of these approaches. It is then processed with a greedy algorithm in order to decide which generalization operations can be shown at the same time. After that, two aspects are assessed: the generalization of the map quality and the assignment itself. The results show that the method is feasible for the vario-scale maps, however, the proposed generalization techniques need to be improved and none of them can be considered as suitable at its current state of development. Nevertheless, the main aspect of the interaction is significantly improved and the solution can be considered feasible. At last, based on various observations and conclusions from the project, some ideas for future work are proposed together with an evaluation of the chosen methodology with underlined drawbacks.

Acknowledgements

In this chapter, I would like to give my gratitude to a number of individuals who helped me during my graduation project:

First I would like to thank Martijn Meijers for all the consultations and support. The topic of vario-scale map representation is very interesting, yet very challenging and it took me a significant time to understand it well enough to work with it in an efficient way. It would not be possible without the help of Martijn, who gave me big independence when it comes to development of the ideas, however he was also helping me to head to the right direction when lack of experience was leading me to obviously pointless solutions. He also provided me with much advice regarding the research work that I find as valuable for me as the acquired knowledge.

I would like to thank Dongliang Peng for all the technical support and research experience as well as for many consultations regarding the implementation of different stages of this thesis. I appreciate a lot in particular our online consultations as well as email conversations, sometimes late at night, even during weekends. I did not imagine such strong support from both of the supervisors and I am very grateful for that.

A lot of thanks to Bartosz Lada and Leszek Orciuch for editorial support and linguistic advice. They took time to read my thesis carefully, sometimes in a very tight timeline, when I was finishing the graduation plan or the thesis itself shortly before deadlines. It would be not possible for me to deliver a properly structured text without this support.

Finally, I wanted to thank my family and friends for all the remote support and conversations during the hard time of the pandemic confinement. It was the thing that kept me sane while being locked out alone for a very long time and I think it would have been not possible for me to finish the thesis on time without them.

Contents

List of Figures

List of Tables

Acronyms

SSC Space-scale Cube

SDK Software Development Kit

tGAP topological Generalized Area Partitioning

AAP accuracy assessment points

GIS geographical information systems

VMR variance to mean ratio

1 Introduction

Technological development has a significant impact on the ways digital maps are used. The digital maps are more interactive than ever before; even relatively low-priced smartphones are powerful enough to provide a truly interactive experience for zooming and panning maps while web viewers provide the possibility for multi-scale representation. However, every operation requested by the user has impact on readability of data – if too many elements of a map are presented on a small mobile phone screen, they can become useless for specific applications and inconvenient for the user [van Oosterom et al., 2014]. The zoom-out operation increases the extent of the map compared to the previous representation, which in turn leads to inclusion of much more information gathered from extended borders. Therefore, to keep readability on the same level for zooming, generalization of the map is necessary (Figure 1.1).

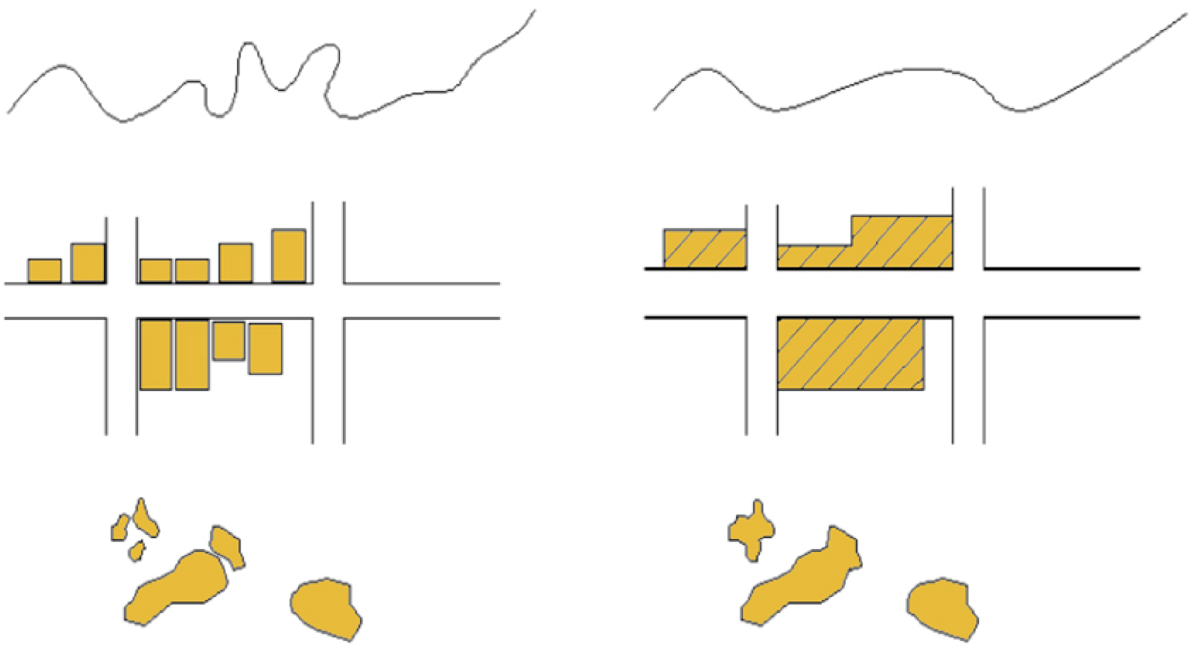


Figure 1.1: Simple generalization examples.

Current solutions are usually based on a traditional, discrete approach for generalization – maps at specific scales are being prepared in a process highly driven by the cartographer who is supervising their visibility and readiness [Frolov, 1975]. Every map is a set of objects constructed for a specific scale. In this case, zooming in or out to a scale which is not linked directly to any set of objects basically means choosing one of previously prepared maps so that it is in a scale most similar to the one of user's interest, and then graphically re-scaling it to make it suitable for the required scale. This approach is far from optimal; the discrete nature of content changes can make the experience anything but user-friendly and hinders the interaction. Meanwhile, increased technological possibilities have led to increased interest in one particular approach to map generalization, namely smooth continuous generalization, which makes it possible to generate maps automatically and follow changes resulting from the generalization process [van Kreveld, 2001]. The need for the possibility to show many maps, each designed for one of many scales, is leading to a simple conclusion – the continuous generalization needs to be automated. Many solutions have already been proposed (Sester and Brenner [2005], Burghardt et al. [2004], Cecconi and Galanda [2002], Thiemann et al. [2011], Oehrlein and Haunert [2017]), but so far no conclusion about the optimal solution has been reached.

1.1 Scientific relevance

As the first step to tackle the automation issue, van Oosterom [1993] introduced the vario-scale data structure, which was then upgraded to a smooth version by van Oosterom and Meijers [2011]. This structure makes it possible to prepare and store maps for many scales without causing huge redundancy – not only enlarged or shrunken picture of the map which can be considered the most accurate with a very limited number of representations prepared for specific scales. Its principles are based on a previously developed solution called topological Generalized Area Partitioning (van Oosterom [1993], van Oosterom and Schenkelaars [1995], van Oosterom [2005]). In short (a more detailed description of the structure is provided later on), it starts with a planar partition (which is a division of a two-dimensional space representing real-world entities) at the most detailed scale and merges the least important objects iteratively with their most suitable neighbours in order to create a new object with new importance based on its new characteristics. Every object's importance is linked to a specific scale in order to provide information as to which point of object needs to be removed due to the need to provide a readable map. Besides that, every object is linked to its so-called parent (an object created by merging) and children (objects existing before merge operation) to build a tree-like structure (Figure 1.2). The process is repeated until the only objects left are those with higher importance than the one linked to the map with the least number of details. This leads to an elegant solution: even with the smallest possible change of scale (usually linked to one discrete step of mouse scroll wheel rotation for better usability), a small change in the map is introduced. This makes it possible for the user to see and understand the way in which map morphing is done and, at the same time, to avoid abrupt changes when the zooming leads to a scale linked to a different map [Peng, 2019].

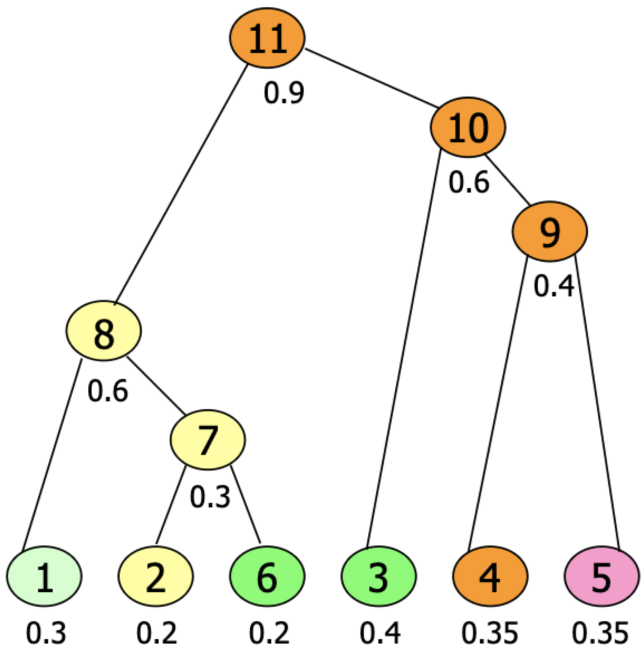


Figure 1.2: tGAP structure. The identification numbers of polygons are denoted by the numbers in the circles with importance value associated. Source: van Oosterom [2012]

However, this approach makes possible to improve user experience. The differences between specific scales are not limited to just two options which are possible for the traditional solution (which is no difference for up-scaled and descaled maps and a completely new setup of objects for maps linked to its different "templates"). A map created after the zooming operation can differ a lot from the previous extent which makes the user feel lost and limits their experience (Figure 1.3. This issue can be handled with the following solution: every object is linked to another object at a different scale to provide information to the server regarding how the generalization procedure should be followed, thus creating a topological relationship between them. The information on those links is stored as a structure called the topological Generalised Area Partition (tGAP). Therefore, the process of displaying the map is different in its principles – instead of choosing a specific map and, potentially, a scaling factor, the list of specific changes (in specified order) is provided, which makes it possible to decide on how it should be displayed to the user [de de Vries and van Oosterom, 2007]. However, this part of the method is still waiting to be solved. Peng [2019] in his thesis proposed a way of making and ordering specific changes in the map which makes it possible to show them to the user in a convenient manner. However, this approach is considered by the author as possible to be improved in a way that some of the changes would be shown at the same time in order to increase the display time for animation showing the change. This research is trying to tackle this problem and evaluate one of the proposed solutions in order to achieve a truly interactive, smooth, vario-scale map, namely parallel step processing.

Figure 1.3: Typical, old-fashioned generalization approach. The same area is presented on two different scales. Source: Jarocki [2018]

1.2 Research question

Displaying changes one by one makes the whole animation very long if there is a need to keep its readability on a constant level for every request. On the other hand, visualizing them simultaneously, but for a longer time, makes the whole process difficult to track and does not differ from traditional approaches – during the generalization process, the same object can be transformed many times, depending on the difference between the origin and target scales, which confuses the user. To solve this problem, a middle-ground solution needs to be developed [van Oosterom and Meijers, 2013] which would take into consideration the readiness of morphing animation while maintaining high quality of the results. A solution of this issue would not be to implement generalization in sequential steps, but to group them into several steps, which can be animated and shown in parallel to speed up the animation but still avoid the problem with abrupt changes (Figure 1.4). To assess that, different algorithms for initial sequence assignment will be evaluated: greedy (referred to as Option A in further sections) and A* (referred to as Option B) algorithms proposed by Peng [2019] and Haunert [2009], both constrained with the target map, as well as a greedy algorithm without any constraints which is used for comparison purposes (Option C). Since Options A and B require the use of target map which needs to be prepared, it is also the accuracy of this map that has to be evaluated in order to be able to assess the overall feasibility of that approach. Furthermore, the number of assigned objects per step needs to be examined as well as the number of steps – this thesis will evaluate the impact of target map provision on this distribution since it is expected that the geometrical constraint can have a positive impact on it. Besides that, it has to be taken into consideration that the proposed smooth vario-scale solution is still in the research phase and there is no clear conclusion about what would be the optimal solution for creating a Space Scale Cube and a web viewer necessary to show the results. All observations mentioned above are considered essential to answer the main question of this research, namely:

What are the possibilities for continuous generalization constrained with the target map by parallel step assignment and how do they perform?

Obviously, to answer this question there is a need to answer many sub-questions in the process of results derivation. The most important ones are listed as follows:

- Is it possible to see smooth transition of map with the proposed improvement?
- What is the accuracy of the result classifications with respect to the initial map?
- Does the target map make the parallel step assignment distribution closer to expected?
- Does the target map make spatial distribution of steps preserved locally?
- Does the choice of algorithm (greedy and A*) has an impact on parallel step distribution?
- What is the impact of the target map on accuracy? Is this solution feasible to be used in generalization process?
- What should the display time of every step depend on?
- How to process the data for the purpose of web viewing?

It is important to note that many elements of the common map generalization such as split and line simplification are not taken into consideration in this graduation project. The assumption is that it will not be possible to make a clear conclusion with many elements and at first the most simple approach should be tested as a proof of concept. Then, further research would allow to evaluate other aspects, such as different types of maps, optimization of server-client communication or large dataset processing. The results of the project will be available online and will be implemented in the form of a web viewer.

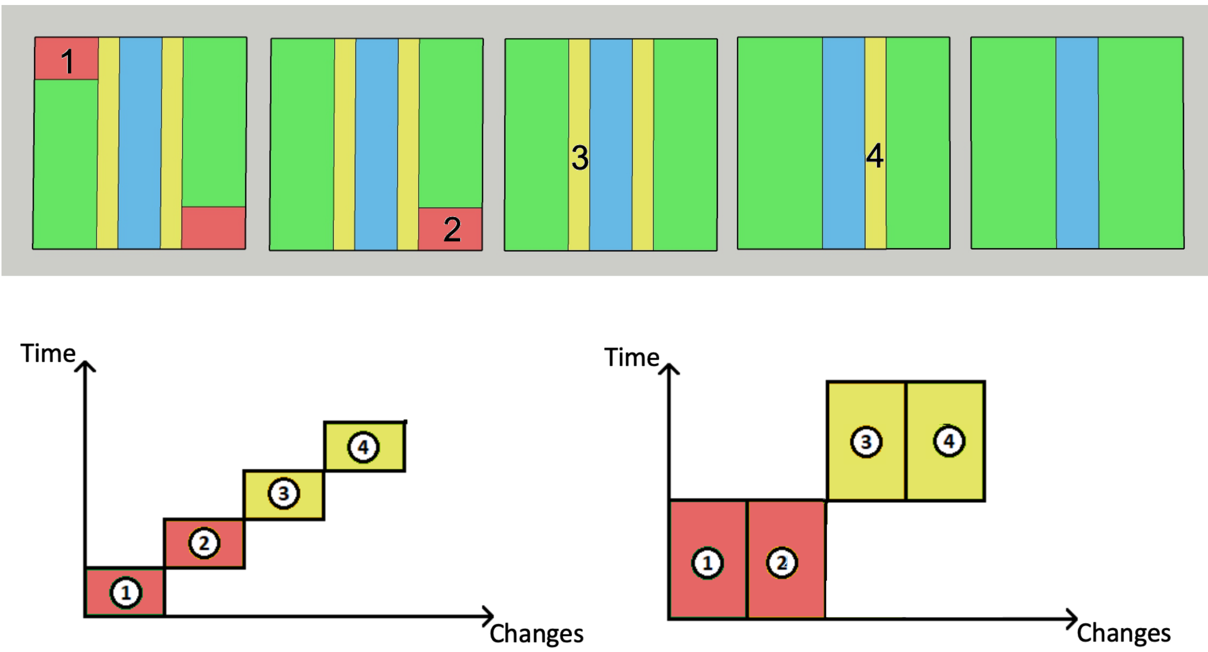


Figure 1.4: The general idea behind parallel step processing shown on the sample used for preliminary testing. In the bottom left image the changes are introduced in time one by one; in the bottom right image they are grouped for parallel processing in time. The top image shows the sequence of changes used for both of the solutions.

2 Related work

The section is structured as follows: it starts with a description of principles of map generalization in the context of the thesis, which is generalization of areas by merge operation. This description is followed by a summary of research results regarding tGAP structure and its smooth version (van Oosterom [1993], van Oosterom and Meijers [2011], Meijers and van Oosterom [2011]) – the thesis directly uses this solution and is trying to contribute to its further development. The next part is a description of two algorithms driving the crucial part of the project: the A* algorithm and greedy algorithm developed by Peng [2019]. Lastly, the thesis outlines the solution for a smooth SSC implementation proposed by Šuba et al. [2013], which is considered a suitable base for the purpose of this thesis as well as the solution for a web viewer is presented.

2.1 Map generalization

As discussed before, in order to keep the readability on the required level, maps at different scales should differ in the number of objects and elements (in other words, in the amount of information) presented as description of some physical entities [Haunert, 2009]. In general, the smaller the scale, the less detailed it should be or, in other words, the more general the representation should be (Figure 2.1). However, in the process of creating a map by generalizing the map at a bigger scale, there are two significant problems: firstly, how to describe the condition for the distinction of the elements which should be left untouched and the ones that should be generalized and, secondly, how should the elements be generalized. Due to the complexity of the map and variety of the ways to describe physical entities or phenomena, this problem is far from solved at the current stage of research in the area. In this thesis one part of the problem is tackled, namely the generalization of areal objects understood as the simplification of its spatial division by merges of objects representing areas.

(a) (b)

Figure 2.1: Comparison of maps presenting the same area at different scales. The map on the right is a result of generalization of the map on the left.

In the following section the principles driving operations in tGAP structure mentioned above are described. Obviously, there are various approaches for the purpose of generalization; however, this thesis does not evaluate this issue and takes the suitability of the already tested solution (which is the merge operation with given rules) as state-of-the-art.

2.2 Parcel merging

Among many different operations to achieve map generalization, two are taken into consideration when processing areas in order to obtain a tGAP structure. These objects are used for areal representation of information on the map. Every area is described by a polygon, and every polygon is related to polygons linked to it. Usually, the areas are also linked with classes using, for example, different colors of labeling. Every color describes a different type of land cover which is represented by the polygon – such as a forest, a road or a building. However, the class assignment process for a specific area will be a generalization of all information included in that place. In other words, in order to include a bigger extent of the map and to not overload the user with the amount of information visible to them, some of this information needs to be incorporated simultaneously. For example, on a large-scale map it may be worth to keep separate classes for different kinds of forest, while on a small-scale map it would be better to include them together as one class describing forest as a general phenomenon (Figure 2.2). Generally, during the generalization process, there is one crucial operation which involves processing of areas, namely the aggregation.

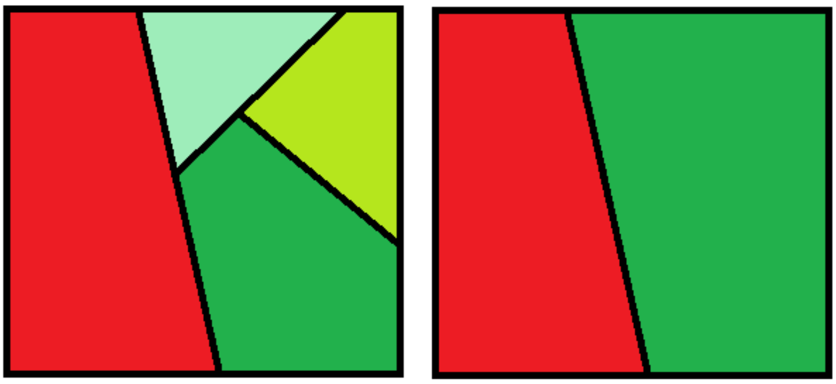


Figure 2.2: On the left different kind of forestation are represented by different shades of green. On the right all of them are represented as one forest class..

One criterion which is crucial to removing areas whose importance is lower than a given threshold for a given scale is the size of the area. While zooming to a smaller scale, there is a need to incorporate more information on the same size of the screen, which means that the smallest polygons have to be incorporated into bigger ones to keep readability at the same level [van Oosterom and Schenkelaars, 1995]. For every change of scale, all areas which do not fulfill the minimum size requirements are processed to find a neighbour which is the most suitable for aggregation. This criterion can be fulfilled using various combinations – the process might choose a neighbour with the longest boundary, a neighbour with the biggest area, a neighbour most similar by its class or even create a similarity matrix and many more options [Peng, 2019]. The resulting area is the area of the parent (the bigger of the pair) and the area of the initial polygon merged together with one boundary and – usually – the class of the parent.

2.3 Topological Generalized Area Partition

To address all the problems and issues driving the automated generalization of the map, topological Generalized Area Partition concept was chosen to perform the task defined in the research purpose. The original idea can be traced back to a GISDATA Specialist Meeting on Generalization held in 1993, when Generalized Area Partition (back then without the prefix referring to topology) was introduced by van Oosterom [1993] as a concept of on-the-fly map generalization which can handle the problem with data redundancy. The idea starts with a description of data as a topologically related structure of nodes, edges and faces, described in tables of, for example, an SQL database.

1. Nodes (also called 0-cells) would store the information about the point and list of edges associated with it.
2. Edges (also known as 1-cells) contain a polyline describing them (by coordinates), its length (useful in further steps) and its left and right face (oriented by the order of polyline coordinates).
3. Faces (named 2-cell respectively to the notation) would store a weight factor (also necessary in further processing), the area and the list of edges surrounding a given face.

The principles of generalization driven by this method were described in a few steps by van Oosterom [1993]. The following list outlines the procedure to generalize a map to reach a specific scale S_{goal} with assigned importance I_s. Note that in the described procedure, for simplification purposes, only the merge operation is provided

1. Assign every face to an unconnected node of tGAP structure.
2. Pick a feature (polygon) a with the lowest importance I_p (usually considered as a polygon with the smallest area), a so-called child.
3. Find a neighbour b of the removed feature which is considered as the most suitable one: the conditions for this process can vary; some of them were described in the subsection 2.2.
4. Merge polygons and assign resulting object to the class of the neighbour c. Link node a and node b to c in tGAP tree structure to describe the topological relationship between them. Note that the resulting feature has a different (increased) importance after merging operation due to a bigger area, which should be calculated and assigned to the object.
5. Repeat steps 2-4 until no features with $I_p < I_s$ are left without connection to any of the polygons in the tGAP structure. The result should be a set of tree-like structures with roots describing the polygons shown on the map at scale S_{goal}.

Obviously, the procedure described above can be repeated many times in order to obtain the results for all the possible scales. The resulting tree-like structure will have the root as the last object left on the map, representing all the information together as one general object. This way of description makes the generalization process simple and automatic – every aggregation can be automatically described in the topological structure of faces, edges and nodes and there is no risk of other problems arising in that matter. Note that the introduction of the split operation is changing the tGAP structure from a tree-like structure to an acyclically directed graph. However, this operation will not be introduced as part of this graduation project. The structure is shown on Figure 2.3.

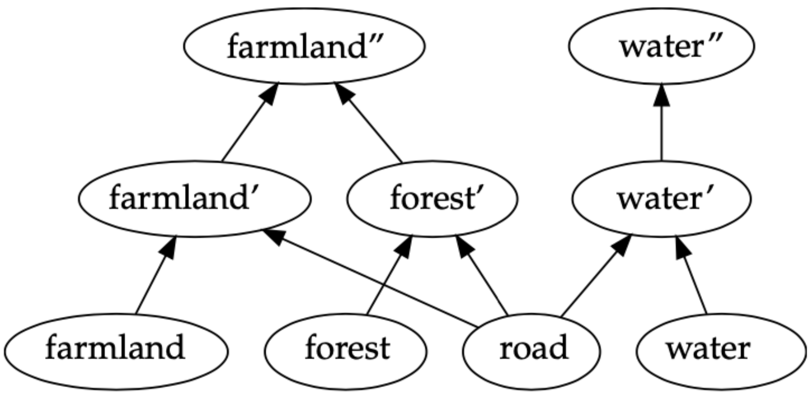


Figure 2.3: Representation of a tGAP structure. Note split and merge of the road. Source: van Oosterom and Meijers [2011].

Obviously, the automated topological structure of tGAP requires an automated solution for map visualization. To tackle that problem, 3D space-scale cube (SSC) was introduced. Its principles can be traced back to research conducted by Hägerstrand [1970]. The idea is relatively straightforward. The structure is represented by 3-dimensional prisms: point objects are represented as vertical lines, line segments as vertical faces (Figure 2.4) and area objects become volumes. The specific map is generated by providing a horizontal slice which leads to a 2-dimensional representation linked to a specific scale.

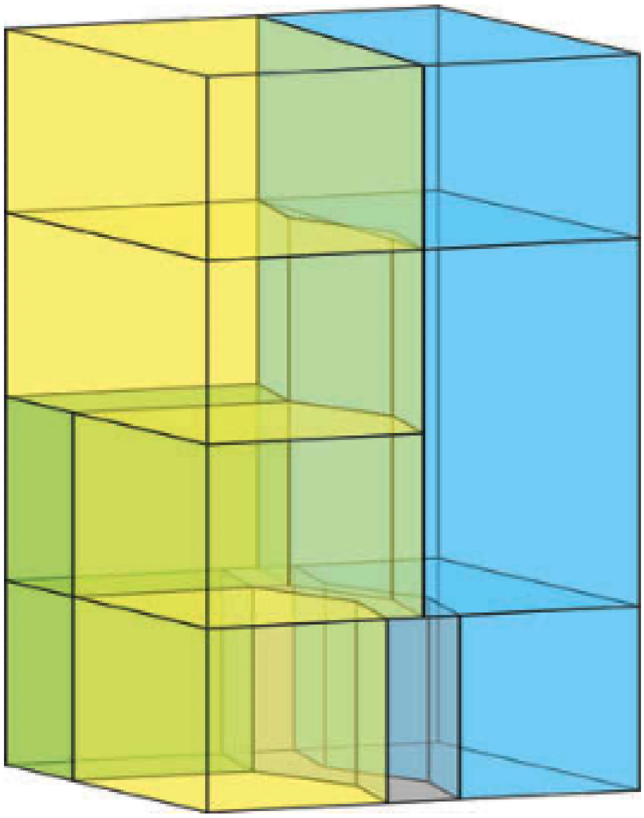
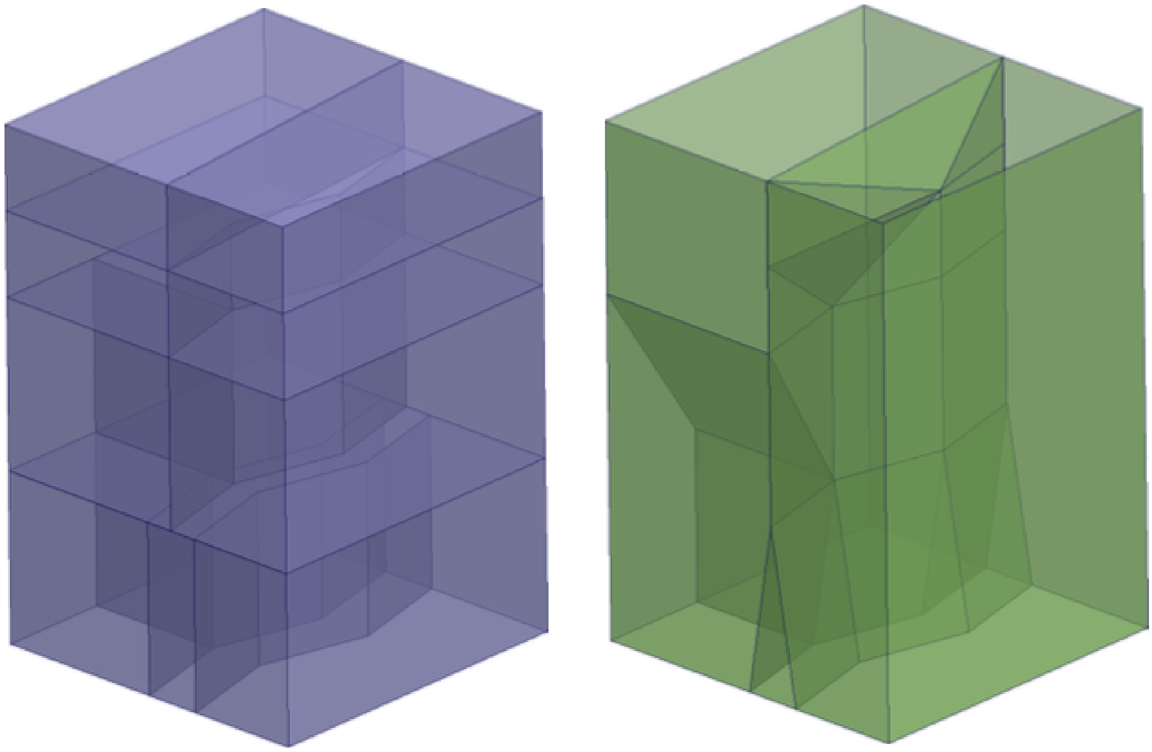


Figure 2.4: Space-scale cube as a representation of a tGAP-driven generalization in 3D. Source: van Oosterom et al. [2014].

However, this solution was able to solve the problem only partially – it proposed an automated solution for map generalization in a topological manner. The solution made it possible to track specific changes which can be used for the purpose of the project but did not allow for their subsequent visualization. For example, the merging operation is done by simply replacing a polygon: two areas become one big polygon with the extent described by the united boundary of the original polygons.

2.4 (Re)searching for the smoothness

Many years have passed since the tGAP structure was created and in the meantime it has been significantly developed. With increased possibilities from the technological point of view as well as increasingly high maturity of the idea facilitated by further research, the initial concept has been transformed into a more challenging task – a tGAP structure with smooth SSC (Figure 2.5). As opposed to the previous solution, this structure does not introduce changes in a discrete manner, in which, for example, the polygon simply could or could not be merged into another one, but it visualises this operation gradually, which could make the whole morphing process visible and possible to be tracked by the user [van Kreveld, 2001]. In other words, the difference between slices is introduced gradually, whereas in a typical structure, the differences were introduced by merging a specific polygon with its parent (with a simplified extent caused by line simplification) which lead to a shock change. The previous slice was not indicating that any change (such as a removal of the node in line simplification or merge) is going to happen to any extent and it is done in a blink of an eye [van Oosterom and Meijers, 2011].



(a) (b)

Figure 2.5: Comparison of the classic SSC (a) and its smooth version (b). Source: van Oosterom and Meijers [2011].

2.4.1 Smooth merge

To redefine merge, a crucial operation for polygon generalization, there is a need to alter the traditional way of description – instead of collecting the stacked prism, the SSC has to be represented by polyhedrons with non-vertical walls which make it possible to show gradual shrinking and growing of polygons. The horizontal planes are removed (besides the top and bottom ones), which makes the structure more compact in terms of storage [Šuba et al., 2014]. This is achieved due to removed nodes and faces which are not carrying any information that cannot be acquired in any other way. Every object is linked to its representation (which is described by a single polyhedron) going through all map scales. Simplified nodes lead to a smaller number of primitives, which has impact on the simplified topology – the structure is more compact in storage and, at the same time, more suitable for a true vario-scale representation. This results in a structure which can be considered more suitable for applications such as web viewers: less information necessary to reach the goal of showing map at a given scale makes the performance better compared to the classical tGAP solution. As can be seen in Figure 2.6, a small change of scale leads to a small change of areas. As opposed to the classical solution, the user receives a possibility to see which polygon is going to be incorporated with adherence to generalization principles.

2.5 Generalization sequence algorithms

One of the necessary steps to reach the goal of this research is to establish the order for further parallel step assignment for every tested approach. Among different approaches, two proposed by Peng [2019] were chosen for the task – the greedy algorithm and the A* algorithm. As part of the research conducted in this thesis, the results from both solutions will be traversed with greedy algorithm to assign specific patches to specific steps. Since development of both algorithms and its evaluation is not an essential part of this research, the following description should be considered as general. A more specific explanation can be found in Peng [2019].

2.5.1 Greedy algorithm

The first algorithm is comparable to the solution proposed by van Oosterom [2005] for establishment of the topological GAP structure. The algorithm proposed by Peng [2019] searches for patches with the smallest areas and evaluates their assignment to one of their neighbours. The choice of neighbour is concluded by finding the most compatible one; for that purpose a following cost function is introduced:

$$f(P_{s,i}, P_{s+1,j}) = (1-\lambda) f_{\text{type}}(P_{s,i}, P_{s+1,j}) + \lambda f_{\text{comp}}(P_{s+1,j}) \tag{2.1}$$

where

$$f_{\text{type}}(P_{s,i}, P_{s+1,j}) = \frac{A_u}{A_R} * \frac{d_{\text{type}}(T(u), T(v))}{d_{\text{type\_max}}}. \tag{2.2}$$

A_u is the area of patch u, $T(u)$ and $T(v)$ denote types of patch u and v and $\lambda \in [0, 1]$ refers to importance of f_{type} and f_{comp}. In his thesis Peng [2019] decided to use $\lambda = 0.5$, which is also the value chosen for the purpose of this research, as it was already tested and considered suitable for this type of task. The specific value for d_{type_max} is provided as input and d_{type} for a specific pair can be read from the matrix for a specific dataset. The second part of Equation 2.1 refers to the cost of compactness proposed by Frolov [1975]. It is based on compactness value of a specific patch u, such as:

$$c(u) = \frac{2\sqrt{\pi A_u}}{l_u} \tag{2.3}$$

where A_u and l_u are the area and the perimeter. In case of the specific subdivision $P_{s,i}$ (which denotes all the patches as a specific step) $C(P_{s,i})$ will refer to compactness of all patches. To arrive at the equation 2.1, Peng [2019] introduced the cost related to compactness. Assuming that there are $n - s + 1$ patches at the given time s, the function describing the cost of compactness is defined as follows:

$$f_{\text{comp}}(P_{s,i}) = \frac{1 - \frac{1}{n-s+1} \sum_{c \in C(P_{s,i})} c}{n-2} \tag{2.4}$$

Note that $n - s + 1$ and $n - 2$ are used for normalization purposes only.

2.5.2 A* search algorithm

Another approach to reach the goal of ordering the sequence of patches is the A* (pronounced "A-star") algorithm proposed by Hart et al. [1968] and altered for this specific application by Peng et al. [2017]. It is described as a refinement of Dijkstra's algorithm [Dijkstra, 1959] with $g(P_{t,i})$ as exact cost of the shortest path from P_{start} to $P_{t,i}$ and $h(P_{t,i})$ as the estimated cost for the path from $P_{t,i}$ to P_{goal}. The total cost is simply their sum and its equation is as follows:

$$F(P_{t,i}) = g(P_{t,i}) + h(P_{t,i}) \tag{2.5}$$

where

$$g(P_{t,i}) = (1-\lambda)\sum_{s=1}^{t-1} f_{\text{type}}(P_{s,i}, P_{s+1,j}) + \lambda \sum_{s=2}^{t-1} f_{\text{comp}}(P_{s,i}) \tag{2.6}$$

and

$$h(P_{t,i}) = (1-\lambda)h_{\text{type}}(P_{t,i}) + \lambda h_{\text{comp}}(P_{t,i}). \tag{2.7}$$

Estimated h_{type} and h_{comp} are defined as:

$$h_{\text{type}} = \sum_{s=t}^{n-1} f_{\text{type}}(P_{s,i'_s}, P_{s+1,i'_{s+1}}) \tag{2.8}$$

and

$$h_{\text{type}} = \sum_{s=t}^{n-1} f_{\text{comp}}(P_{s,i''_s}). \tag{2.9}$$

The purpose of Equation 2.5 is to reduce the searching space – the better estimation of h, the more reduction can be done by A*. A further explanation of the estimation of cost $h(P_{t,i})$ can be found in the PhD thesis written by Peng [2019].

2.6 Web viewer

A web viewer has to be implemented in order to enable and display interaction with the proposed solution. Huang et al. [2016] proposed three different versions of a client for the tGAP, however, only one of them was considered suitable for the smooth structure. The most challenging problem to solve here is the implementation of the SSC – a three-dimensional representation of the tGAP structure together with the information about objects which would enable to process it with slicing in order to generate a map which should be displayed. Since not all objects are convex, it can be challenging to find an automated way to support smooth, gradual degradation of the polygon which would look natural to the user in any type of situation [Šuba et al., 2013]. Suba [2017] in his thesis proposed three different solutions for that: single flat plane, zipper and eater, as well as a way to render the results, make horizontal intersections and map the textures. For the purpose of this thesis, current state-of-the-art solution (the closest to the 3rd option proposed by Huang et al. [2016]) was provided by Dr. Martijn Meijers and Dr. Dongliang Peng. However, it needs to be slightly altered to make the web viewer suitable for parallel step processing.

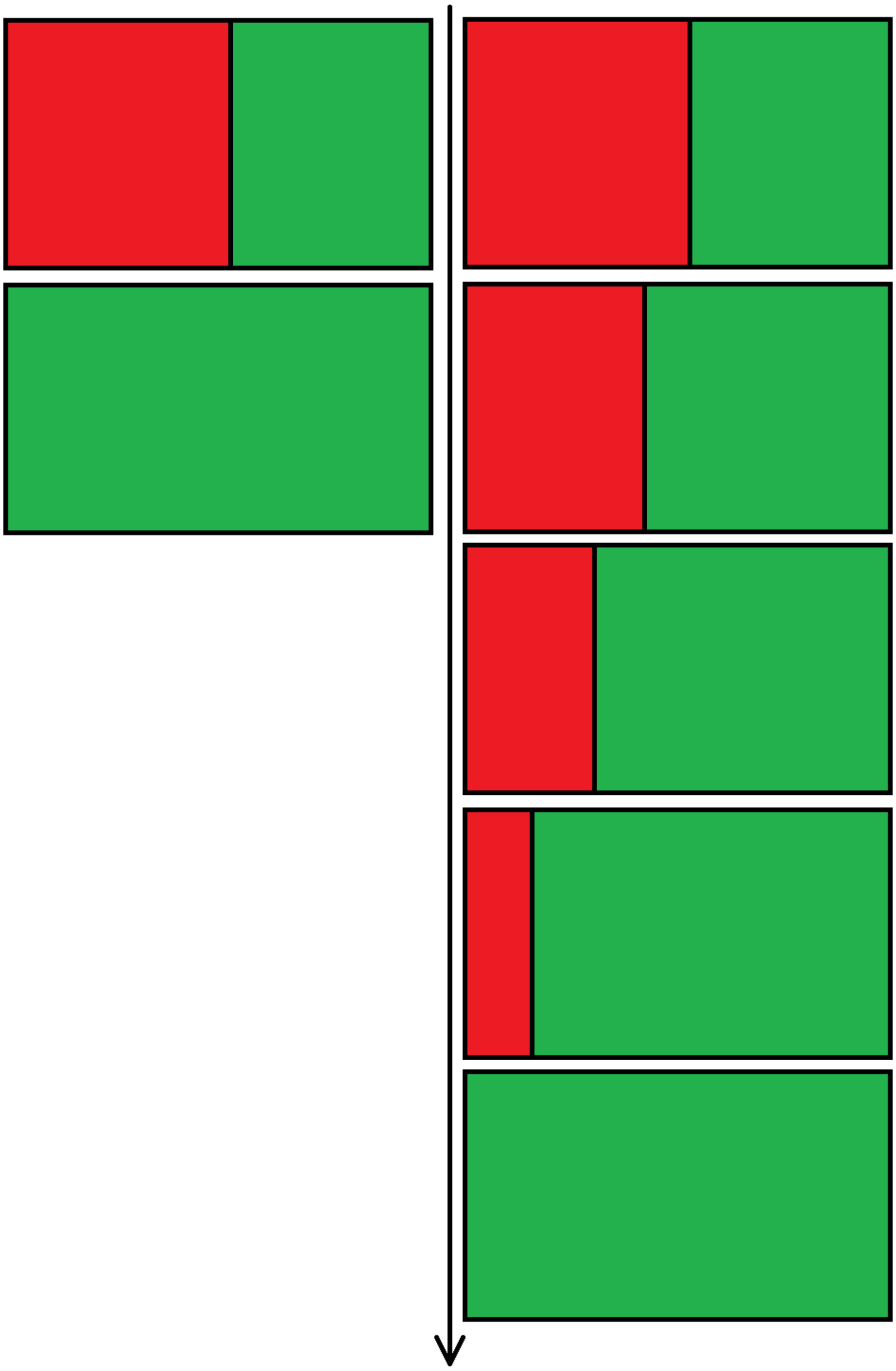


Figure 2.6: Principles of the merge operation described in the smooth SSC (right) compared to classic approach (left).

3 Methodology

The overall methodology of this research can be divided into a few steps. It starts with planning and conceptual development of the applied methods with assistance of literature review. The second part is an investigation into currently used solutions, since this thesis contributes to an already existing and on-going project. In the next part, it will require an implementation of the necessary functionalities. Lastly, it will provide an evaluation of accuracy and performance of all discussed options by comparing it with a proposed benchmarks. This part will also include an estimation of indicators such confusion matrix or steps distribution. The conceptual workflow is shown in Figure 3.1. A more detailed explanation can be found in Figure 3.2.

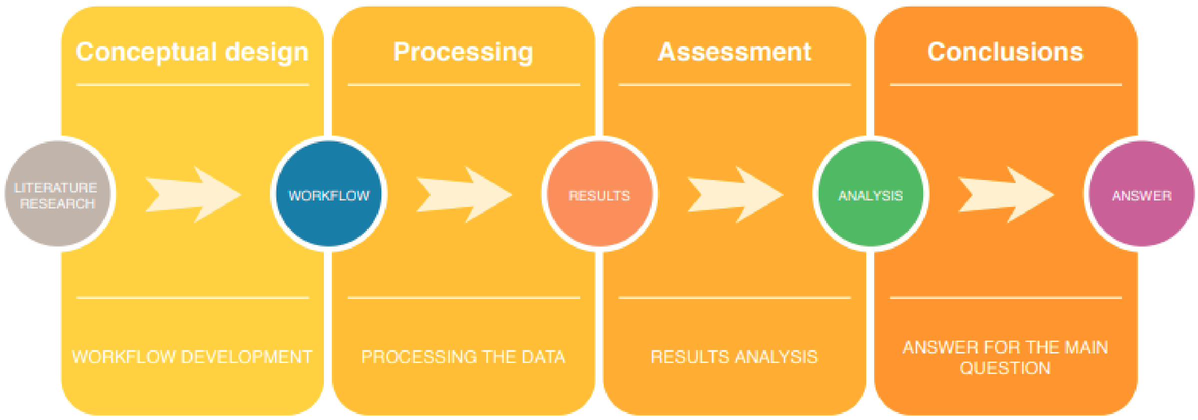


Figure 3.1: General workflow driving the thesis. Rounded rectangles describe the main steps, while the most important inputs and outputs are shown in the circles. The key tasks to be done to derive the results are listed under the horizontal line.

3.1 Data preparation

One of the most important tasks was preparing the data (part of TOP10NL dataset which will be described in a further part of this thesis) so that it fits the requirements of used algorithms. First, several limitations of the A-star algorithm were taken into consideration, namely:

- Simple geometries of polygons. Implementation of the algorithm used for the purpose of this graduation project required no inner holes, only exterior rings were accepted as valid objects.
- Limited number of objects. The nature of graph search algorithms makes them inefficient and not able to deliver results for very large graphs created with a too high number of objects.
- Tolerance of spatial placement of nodes. The dataset was created in an automated way and because of that it contains small polygons (so-called slivers) with nodes so close to each other that the algorithm considers them as one point with a calculated tolerance assumed as proper by a developer. It leads to overlaps which make the algorithm confused as well as to degradation of

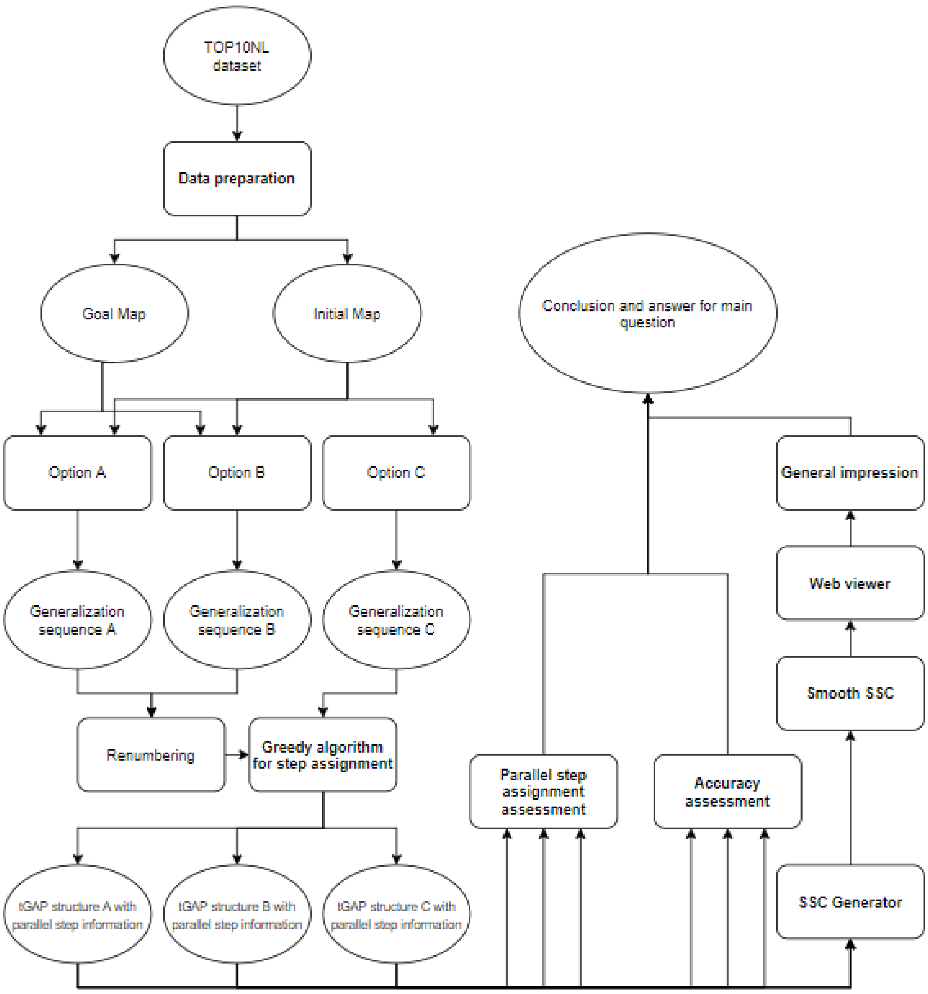


Figure 3.2: Detailed workflow driving the thesis.

triangular geometries to lines in the cases of closely positioned pair of nodes of an object (Figure 3.3).

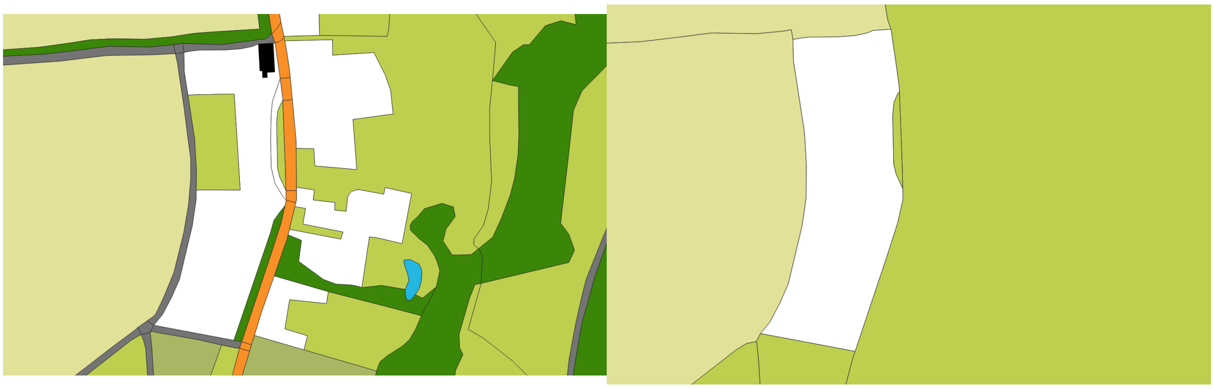
- Exclusion of "islands". The approach chosen in order to prepare the target maps can lead to objects being left unchanged in the transition from initial to target map. Those objects had to be excluded from the algorithm input since it is also confusing for the implemented algorithm and unified with results of generalization later (Figure 3.4).

From the initial dataset of 13238 objects 8091 were created as the initial map, including islands (this category is excluded from processing and will be incorporated to the results separately later on).

3.1.1 Road rings

Besides all the requirements mentioned above the nature of the A-star algorithm requires preparation of a target map. In other words, the algorithm needs the information as to which objects should be finally merged together and what is the class of the final object representing all of them on the target map. In order to provide this information, the so-called road ring approach was implemented. The whole dataset was divided by a network of roads which were also included in the map as spatial objects. Every object was assigned to the group of other object included in the same enclosed area, where enclosed area

Figure 3.3: Examples of polygon which were not eligible to be processed by the current implementation. On the left, all objects are shown and on the right only degraded objects are visible (probably because of poor reconstruction of buildings). The yellow polygon was left for easier orientation.



(a) Before

(b) After

Figure 3.4: Examples of polygons not changed in the process of transition from the initial map to the target map with the road ring approach, because of detailed road network.

denoted the space fully surrounded by the set of neighbouring polygons with the class "road". The final object is represented by geometry created by merging all the objects assigned to a specific group and the class of the final object is the class with the largest sum of the areas of the objects grouped by classes assigned to them. The workflow of target map preparation is shown in Figure 3.5.

3.2 Algorithms – generalization sequence

As mentioned before, different algorithms are evaluated in this research in order to answer its main question. The differences among them can be observed by comparing two aspects: the way of ordering the patches in order to obtain a generalized map and the way of assigning ordered patches to specific parallel steps. Each of these aspects needs to be tested with specific parameters for parallel steps assignment and compared with one another. For the purpose of this thesis, various approaches to establishing the sequence of generalization with different solutions were chosen. They are listed as follows and, later, described in respective subsections:

- Option A – greedy algorithm with information about the target map that should be achieved.

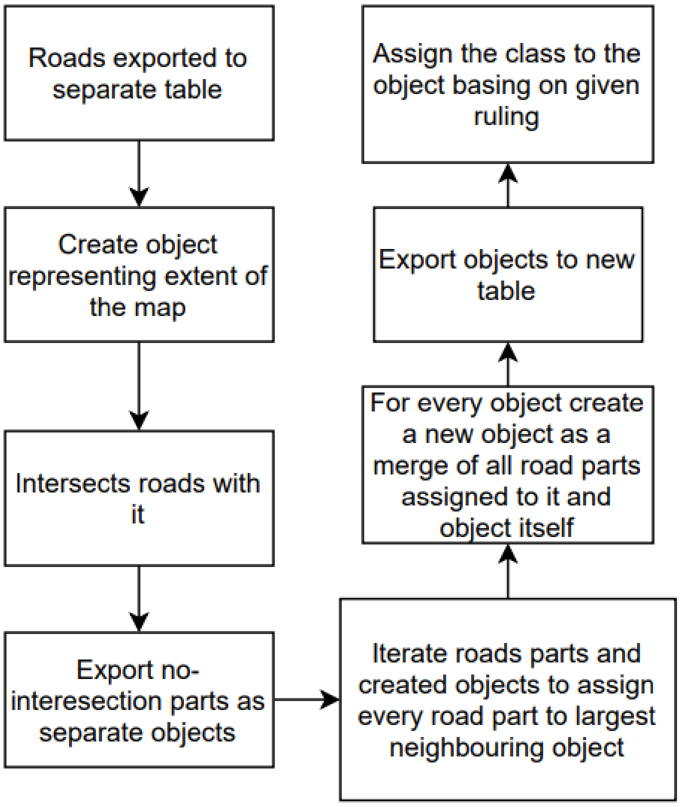


Figure 3.5: Workflow describing target map preparation.

- Option B – A* search algorithm with the same information about the target map that should be achieved.
- Option C – greedy algorithm without any information about what the the target map should look like.

3.2.1 Greedy algorithm with target map – Option A

This approach, described as Option A in this thesis, uses a greedy algorithm implemented by Peng [2019] to establish the order of patches within given geometrical constraints of an object on the target map and the information about the final class of this object. The algorithm's principles were described in the previous section. The simplest approach was chosen for the purpose of this thesis, which means that the most suitable neighbour is simply the neighbour with the longest boundary with the object. In order to make sure that the class of the object on the final map will be the same as the class of the object on the target map, the original criterion for the class of the parent object was changed. Instead of choosing the class of the bigger polygon in the part, the algorithm will compare similarity of classes of both objects and choose the class with a higher similarity to an object on target map from a given table. The values in the table represent the distance from class A to class B which is expressed by the number of nodes necessary to visit to get from class A to class B on a given tree structure representing all classes. The structure is shown in Appendix A. In a situation in which the similarity value is the same for both objects, the class of the bigger object is chosen as the class of the parent.

3.2.2 A* algorithm with target map – Option B

This option is based on the A* algorithm implemented by Peng [2019] for creating and ordering patches. This algorithm also uses geometrical constraints of the target map as well as information about its class. It will search for possible graphs created based on the initial map and the target map in order to find the solution considered as optimal. A more detailed description of this algorithm has been provided in the previous chapter.

3.2.3 Greedy algorithm without target map – Option C

In this option, a greedy algorithm is used just as in option A, but it does not have any geometrical constraints or information about class of a given object on the target map. The assumption is that the information acquired from the target map can be valuable for parallel step assignment: the algorithm knows in advance how many changes are involved in order to reach the object on the target map starting from the initial map (assuming that every merge operation involves two polygons and results with one, the number of operations is simply the number of objects on the initial map minus the number of objects on the target map). However, the impact of this approach on accuracy of the map still needs to be tested and because of that the most common algorithm with no constraints is proposed as a benchmark.

3.3 Parallel step parameterization and assignment algorithms

One of the sub-questions crucial to answer in order to answer the main question is referring to parallel step parameterization. The idea behind parallel step assignment is straightforward: its goal is to improve the interaction between the map and the user by extending the presentation time for a specific change shown to the user. Assuming that the overall time for the whole transformation from initial to target map is fixed, this approach requires a parallel presentation of those changes in order to not extend the whole presentation time. Obviously, the assignment to a specific step cannot be done without fulfilling some basic requirements i.e. regarding the problem of shock changes, which leads to the conclusion that the expected (optimal) number of changes which we want to assign to a specific step can be different compared to the actual number of changes that the algorithm will be able to assign. On the other hand, it should be also taken into consideration that the map should not change too much at once, even if the changes are not geometrically related to each other – for that purpose, a maximum area (a percentage of the whole map) of the parallel step has to be proposed.

At the initial stage of the research done for the purpose of this thesis, one of the ideas was to take into consideration what the display time should depend on. However, the state-of-the-art implementation of the web viewer makes it possible for the user to set it according to their personal preferences and so it is considered pointless to evaluate what should this time depend on besides this aspect. On the other hand, it was also considered if the time should vary among different steps, but this idea was also discarded since the case study is conducted on a relatively small part of data and in real-world application, especially with small-scale maps, when many tiles are shown, this can lead to inconsistencies in the way the SSC is sliced by the web viewer.

3.3.1 Parallel assignment of unconstrained generalization

In Option C the result of data processing is one single sequence of operations, ordered by importance of polygons involved. In that case the greedy algorithm for parallel step assignment is proposed. Assuming that the initial step number is X, algorithm starts with the first pair of objects from the operation which were not taken before and which are going to be merged in order to create the object representing both of them. The algorithm marks both of them as locked, as shown in Figure 3.6. If none of those objects were locked before, the operation is assigned to step X. If the opposite is true, the algorithm changes the step number to $X + 1$, marks each object as unlocked and continues with the assignment process. It is possible to constrain the algorithm by providing a maximum number of operations that can be assigned to specific steps. In that case the algorithm will change the step number to a higher one and clear the locks, similarly to a situation with already locked objects described above. The example of possible results is shown in Figure 3.7. However, this approach does not take into consideration the size of the part of the map which changes in one specific step. In order to prevent the situation when too many objects are assigned to specific steps, the algorithm will check the area of assigned objects after every assignment and will terminate assignment to the current step if the total area of the assigned object is higher than 1%, 5% and 10% of the overall area of the map. The results will be examined in a web viewer in order to evaluate the visibility of a smooth transition for every option and the most suitable one will be chosen by the user.

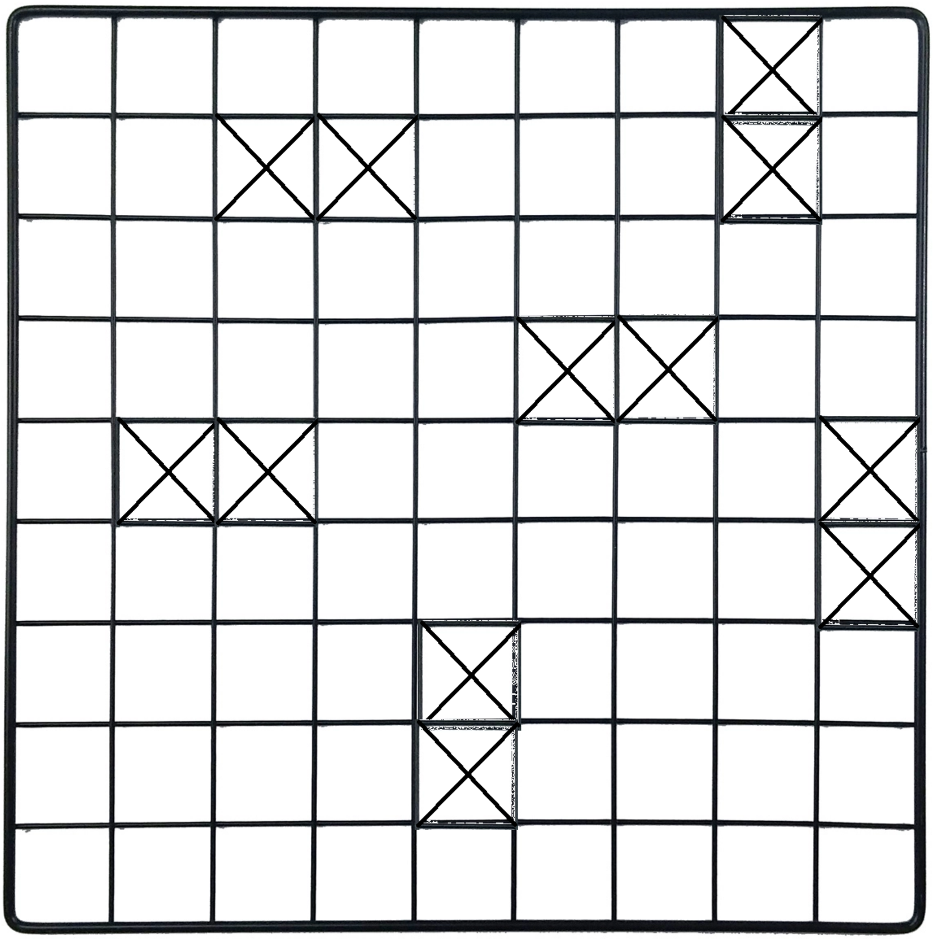


Figure 3.6: Principle of locks. Each rectangle denotes a polygon and X in the middle denotes polygons locked for the actual step.

3.3.2 Parallel assignment with target map

Because of a substantial difference in the general approach used in Options A and B compared to Option C, it is possible to test a different approach for step assignment. It is important to mention that every parameter of the parallel step assignment, such as the number of events in one step, is estimated in order to achieve one final goal – parallel step assignment which is considered as the most suitable for a specific application - in that case, interaction with the map. By changing the number of steps or number of operations in every step it is possible to adjust the amount of information which the user should receive at some specific point in time. This is, however, based on a hidden, bold assumption that

Figure 3.7: Example of objects (marked yellow) assigned to the same parallel step.

provided global parameters (for the whole map) are preserved locally. For example, the assumption that for a specific map extent the user should not be shown more than 1% of the area at once and the implementation of that assumption does not guarantee that every quarter of the map will be covered with changes in 1% in every step as well as that every quarter of the map will have any changes at all (Figure 3.8).

Provision of the target map is, in other words, information about the sub-tree structures in the whole tGAP structure which represent generalization of the map, where a sub-tree refers to the sequence of operations performed in order to transform a set of objects on the initial map to one object on the target map. In cases when, instead of checking the geometrical relation of the objects with respect to other objects to avoid shock changes, it is possible to simply check to which sub-tree a specific operation belongs and choose one operation per each sub-tree for a specific parallel step X to be sure that there is no problem with shock changes in that step. The proposed algorithm for initial step assignment looks as follows.

1. Renumber the sequence of operation for each sub-tree (referring to a specific object on the target map) to $1000 - X + Y$, where X denotes the total number of operation in sub-tree and Y denotes the initial parallel step assignment acquired from generalization sequence (one by one). The value of 1000 was chosen to make sure that there will be no negative numbers in numbering (the expected number of operations for every sub-tree is expected to be significantly smaller). The example of results is shown in Figure 3.9. This approach will allow to make sure that it is possible to create a list of operations which involves every polygon present on the target map (except for the objects which are not changed in the process of generalization, i.e. when an object is not changed between

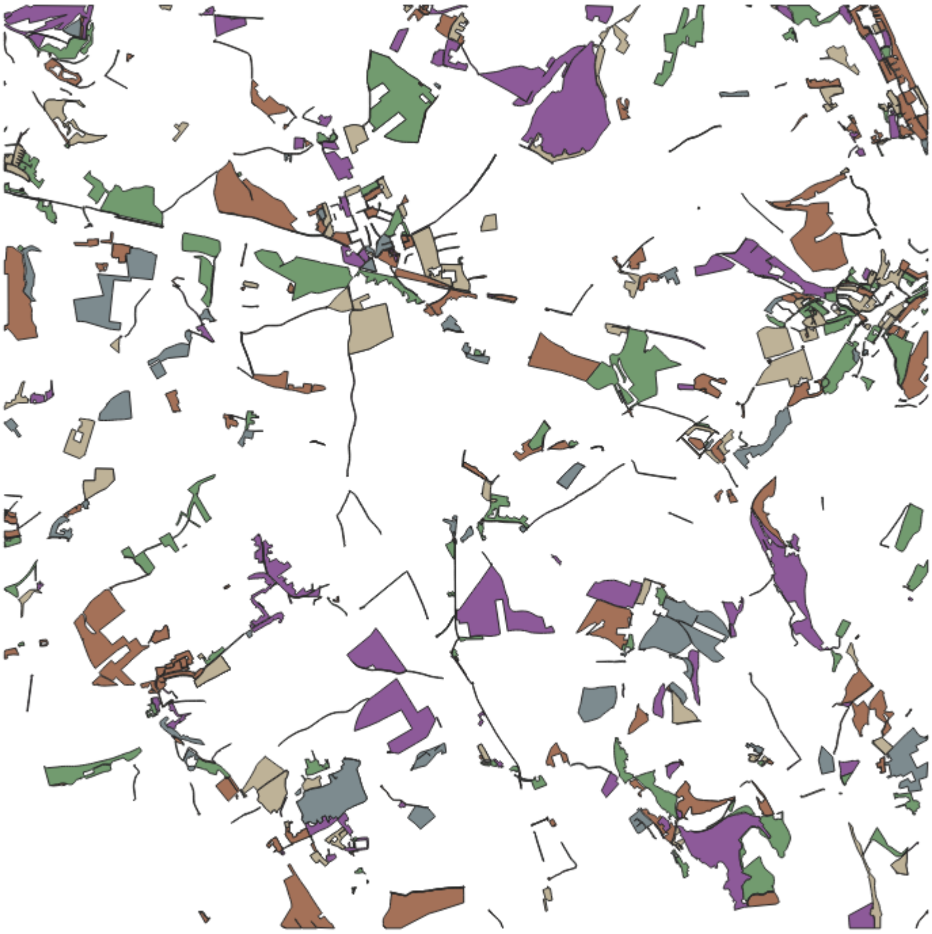


Figure 3.8: Example of preliminary results with several parallel steps marked with different colors. It is clear to see that some parts of the map are almost not involved at all, which will make the user wait during the processing of steps which are not assigned to the area of his interest.

initial and target map), since the last operation for every object on the target map will be assigned to the same step by equation given above (it will be the number 999 or 1000, depending on whether it starts from 0 or 1 within the original numbering of every sequence).

2. Process the results with the greedy algorithm used for Option C, but instead of using importance value for consecutive ordering, the algorithm will base the order on value Y (initial parallel step assignment). Obviously, for many values there will be multiple objects with the same numbers (for $Y = 1$ there should be 590 of them). Because of that, an additional second order is proposed – the total number of element in sub-tree that the object belong to. Parameterization will be the same as for Option C (1%, 5% and 10% of total area of the map) to limit number of parallel operations.

At the early stage of the research, one considered approach was to not renumber the sequences. However, it was concluded that the theoretical difficulty of the parallel step assignment (measured by how often shock changes condition and how often maximum area is reached) is increasing during the process of generalisation. The chances that it will be possible to include several changes related to the same object on the target map in the same step is higher at the beginning of generalization.

subtree1	1	2	3	4						
subtree2	1	2								
subtree3	1	2	3	4	5	6	7	8	9	10

subtree1							7	8	9	10
subtree2									9	10
subtree3	1	2	3	4	5	6	7	8	9	10

Figure 3.9: General idea of renumbering in case of 10 instead of 1000 as the last step. Sequences shown at the top were renumbered to sequences shown at the bottom of the picture.

3.4 Benchmark and evaluation

In order to be able to evaluate the applicability and performance of each approach, the solutions need to be assessed. First, the accuracy of the results needs to be checked. The old-fashioned map generalization is considered to be a labour-intensive process for many reasons – the need for supervision by the cartographer being a big part of it – but it also guarantees a high quality of results. Because of that, it is not possible to draw a conclusion that the automated approach will be suitable for the application without any doubt and so its performance needs to be checked. In this thesis, in order to create the target map, the road ring approach is used. Obviously it is not possible to achieve accuracy of this simple, automated approach which can be comparable with a map created in a user-driven process. However, its relative accuracy of map presentation in comparison to the most common approach for generalization for vario-scale map can still be calculated and can provide valuable information about further research in that direction. In order to evaluate the accuracy and quality of this generalization, the procedure shown in Figure 3.10 is proposed and explained in the following section.

3.4.1 Accuracy assessment

The accuracy assessment will be performed by comparing the maps prepared in the process of generalization with the initial map in order to evaluate how accurately the information about the class of a specific point of the map was preserved. This test will be conducted using maps at various scales, using the definition proposed by Huang et al. [2016]. Assuming that the initial set represents the map at the scale with denominator D_{init} with number of elements equal to N_{init} and that the number of objects presented should be on average as written byHuang et al. [2016] and the window viewer size is constant, it is not complicated to derive the conclusion that the number of objects representing a given map at scale D_x is as shown in Equation 3.1:

$$N_x = (D_{init}/D_x)^2 * N_{init} \tag{3.1}$$

Then, deriving a specific scale from tGAP structure requires counting the initial number of objects and the number of operations which should be performed: each operation reduces the number of objects by one, which leads to Equation 3.2 describing the number of operations N_o:

$$N_o = N_{init} - N_x = N_{init} - (D_{init}/D_x)^2 * N_{init} = N_{init} * (1 - (D_{init}/D_x)^2) \tag{3.2}$$

Proposed scales to test the accuracy of dataset were shown in Table 3.1 together with the number of elements on the whole map with respect to the initial map.

The idea is to generate a pseudo-random distribution of Accuracy Assessment Points (AAP) over the extent of the map and intersect the points with a whole structure of objects representing maps at all scales stacked on each other. Since the same place on the map is represented by many objects in many scales, it will create a list of objects with assigned classes for every pseudo-random point, representing

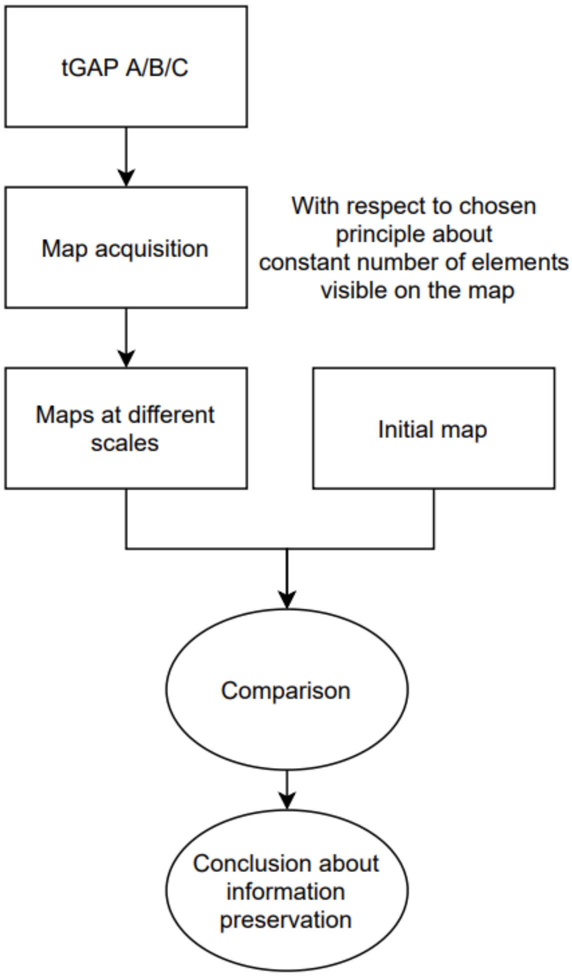


Figure 3.10: Workflow describing accuracy assessment of results.

the specific point of the map in every scale possible. In order to derive the results for the scale of interest of the user, the number of objects presented on the map at a specific scale is calculated with equation 3.2, and subtracted from the initial number of objects in order to obtain information about the number of operations necessary to perform to consider the map as the map at a specific scale. Since the sequence of generalization is ordered, it is possible to conclude which objects are still visible at specific scale and choose the object with lowest possible importance among visible objects from every list of objects linked to every AAP.

It is important to remember that with parallel step assignment it is not possible to show the map in every of it states – in the old-fashioned approach, every change of the map was provided separately. In case of parallel assignment the number of objects considered as suitable for given scale has to be snapped to the closest number of objects possible to show with parallel steps.

This information will be used to build a confusion matrix which is a common tool for evaluating classification. Details about the construction of such matrix are shown in Chapter 4.

Scale	Objects [%]
1:10000	100
1:15000	44.4
1:20000	25
1:25000	16
1:30000	11.1
1:40000	6.25

Table 3.1: Scales proposed at the benchmark of the dataset.

3.4.2 Parallel step assignment assessment

The second part of the evaluation process will try to evaluate the parallel step assignment performed on all the results. With conclusions about display time already drawn (it should be equal for every step), it will assess how equally it was possible to distribute the operations among steps with respect to initial assumptions. Besides that, assuming that the user is interested in the fragment of the map related to the position he is searching information about, the assumption is that it should be also examined if the parallel step assignment is preserved locally on the map. With not-constrained greedy approach it is expected that there is no control over the distribution of changes and it is possible that some part of the map might not be involved in some of the steps at all. With this approach, there is no guarantee that it is possible to achieve the same results as with the constrained approach – that in the last parallel steps all the objects are involved in operation (or half of them in case of division to sub-steps as described above). To assess that, parallel step assignment will be also checked locally, on parts of the map, to evaluate preservation of step spatial distribution. For comparison variance and variance-to-mean ratio (VMR) will be calculated. For that purpose total area of objects assigned to every step will be used.

4 Implementation

4.1 Data preparation – initial processing

As it was mentioned before, preparation of the initial dataset (which is a part of TOP10NL dataset as it was described before) was crucial because of limitations of the implementations which make it impossible to process the dataset in the original form. Firstly, because of a limitation of A* algorithm implementation (it cannot proceed polygons with complicated geometries such as polygons with inner rings), all the polygons lying inside other polygons were removed and geometries of polygons were adjusted to exterior rings describing it. This operation also removed from the dataset objects which would definitely have a significant impact on the number of steps possible to reach – as can be seen in Figure 4.1, there are many objects from building class inside of one object from a different class. To avoid shock changes, every incorporation of every building should be shown separately, making it necessary to perform a few parallel steps just because of very small polygons and their unfavorable spatial distribution.

In order to perform the A* algorithm on the dataset, the current implementation requires the provision of initial and goal map and automatically iterates objects on initial map in order to relate them to objects on goal map. In that process, the algorithm also makes sure that the geometries of polygons are correct (no overlaps, no degraded polygons) and adds polygons which were not possible to relate to a list called Attention List. Around 50 polygons from the initial dataset where considered as not possible to proceed. The reason for that is the lack of consistency between tolerance for spatial position used in the algorithm (it was used for a different dataset prepared with different methods, thus its parameterization does not always fit perfectly the characteristics of the dataset used for the purpose of this thesis) – some of the smallest polygons with nodes were placed relatively close to each other and the algorithm considered them to be the same point, changing its geometry, causing overlaps or sometimes (in cases of triangular polygons) it was degrading it to linear objects making it impossible to proceed (Figure 4.2). Since the purpose of this thesis is a comparison of different approaches, for two other options the same dataset was used for the initial map as for option B (with the same limitations).

After these two operations, the dataset can be described as can be seen in the Table 4.1. As it was mentioned above, the class with the most changes was the urban class, especially the class describing the buildings itself. Besides that, a few water-class objects were removed – those describing ponds, small lakes, features which are commonly placed inside of other objects.

Class	Before	After
Road features	3775	3770
Water features	218	65
Urban features	5211	453
Forestation	2928	2715
Farmland	1086	1086

Table 4.1: Details concerning the datasets used for the experiments.

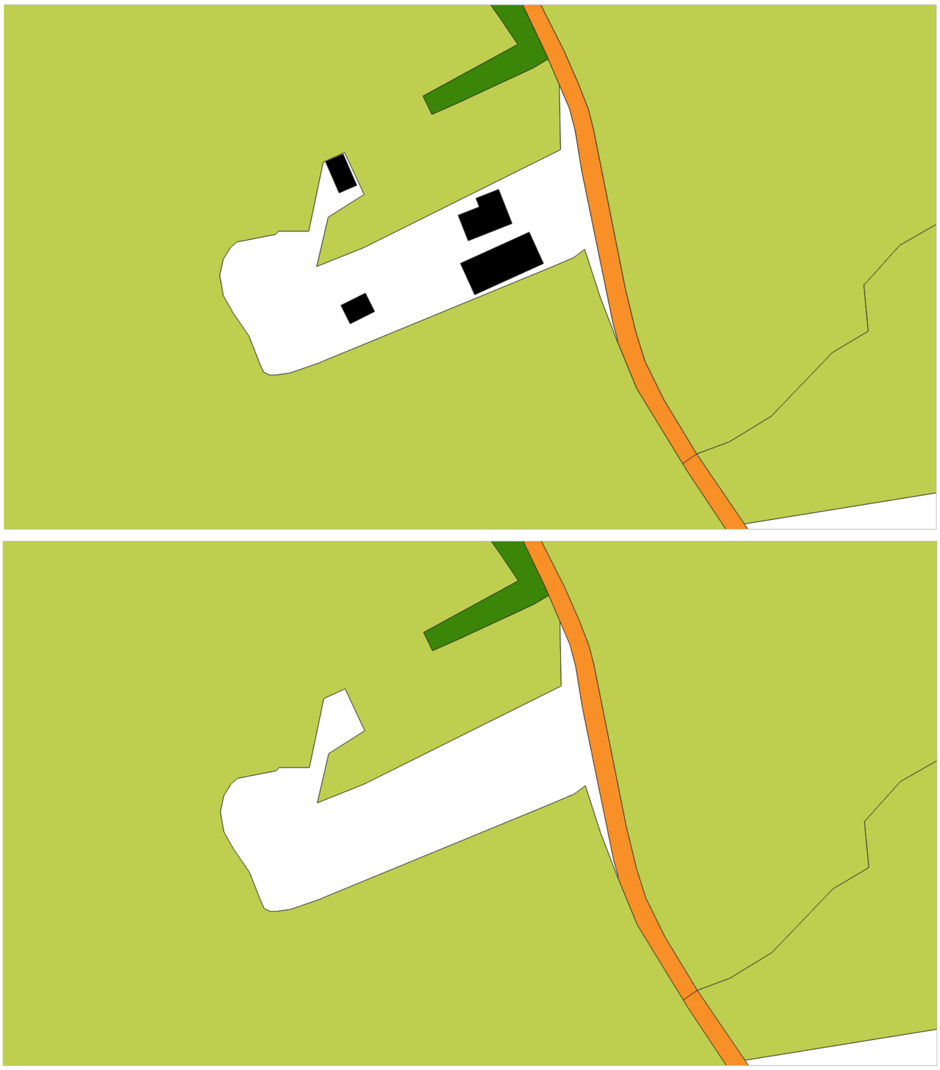


Figure 4.1: Example of data transformation in the preparation process. The most significant difference is a substantial degradation of objects representing buildings, since a lot of them were entirely surrounded by a single polygon representing an urban area around it.

4.2 Data preparation – preparation of the goal map

As it was described in Chapter 3, Options A and B, which use the A* algorithm, require preparation of a goal map to provide the information necessary to build the graph which the algorithm searches for. The resulting number of polygons to proceed in order to achieve an object on the final map vary from 2 to 163 objects between an object on the initial map and a related object on the goal map. The classes for every object were assigned using a simple approach:

1. For every object O_x on the goal map a list of objects on the initial map was created.

2. On every list, all the areas of objects were calculated and areas referring to the objects with the same class were summed up. A list of classes L_x and related areas for every object O_x on the goal map is created.

3. The class of the object O_x on the goal map is the class with the largest area on the list L_x among areas linked to object O_x.

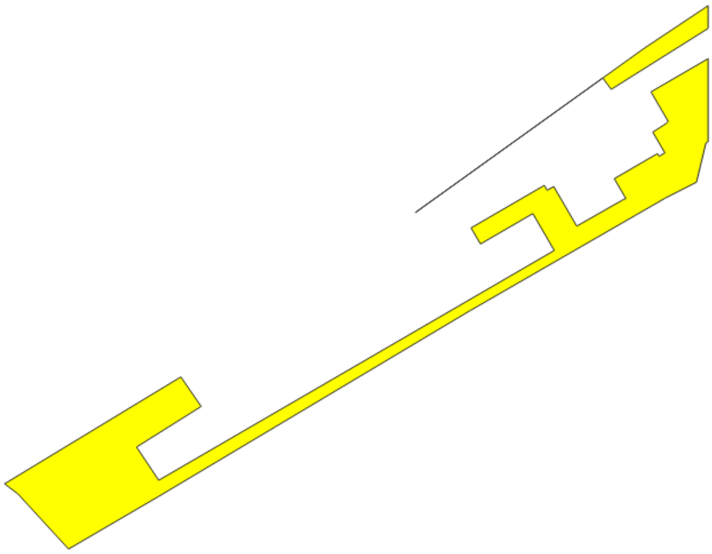


Figure 4.2: Example of polygons not eligible to be processed by the algorithm. Especially in case of the smaller polygon it is clear to see that the geometry is not correctly represented because of the dangling line.

Class	Initial	Goal
Road features	3770	0
Water features	65	1
Urban features	453	190
Forestation	2715	206
Farmland	1086	193

Table 4.2: Comparison of classification of an object on initial and goal map.

This approach makes it possible to take a look from the perspective of the whole dataset instead of following simple, local rules such as it is in case of Option C, where the algorithm decides on the specific class assignment based on information about only those polygons which are involved in the operation. Especially in cases when the largest polygon does not represent the class which is the most common class in some specific area of the goal map, it can lead to a situation in which the whole area is assigned to the class which is not an accurate representation of phenomena occurring in this specific place. The resulting map was related to the initial map in order to establish groups of objects which should be merged together. The resulting classes, together with a quantitative assessment of the created dataset, were shown on Table 4.2 and the final goal map is presented in Figure 4.3.

4.3 Generalization sequence – processing the data with algorithms

With the whole dataset prepared, the research can move to the next step, which is establishment of generalization sequence within each tested approach: for Option A with the constrained greedy algorithm approach, for Option B with the A* approach and for Option C with a unconstrained greedy approach.

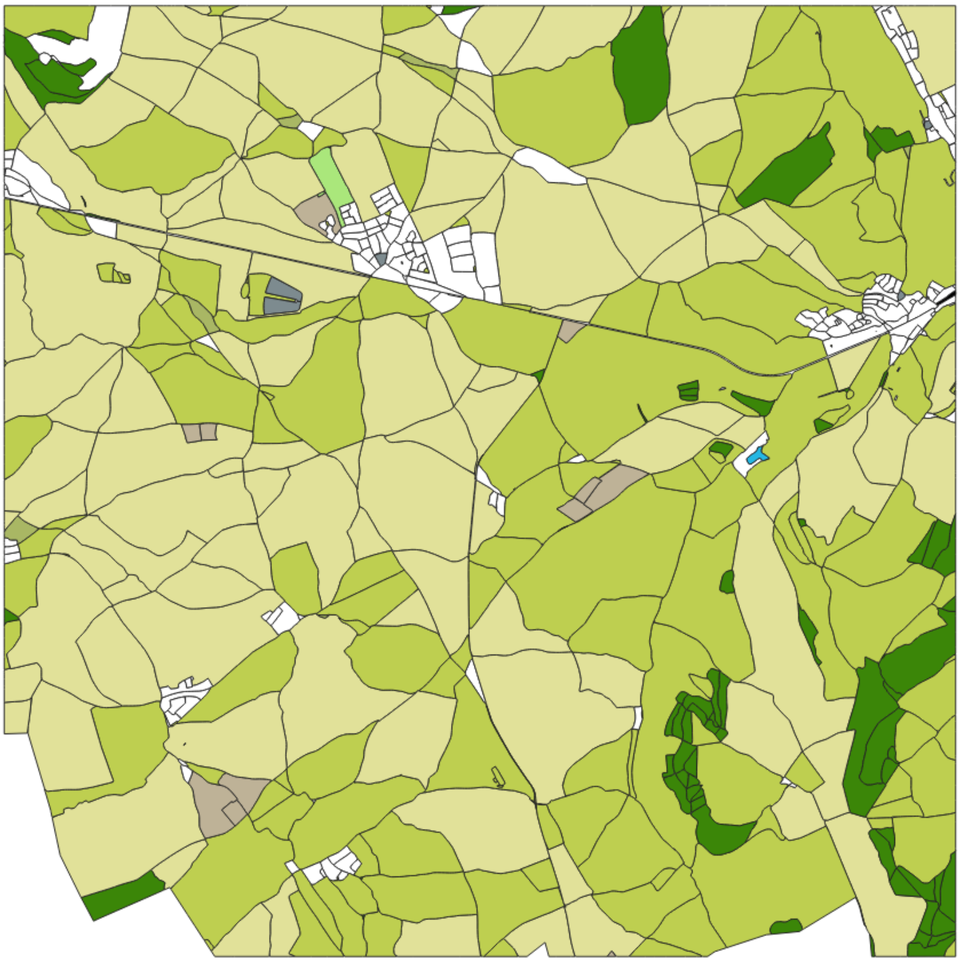


Figure 4.3: The final result presenting the goal map prepared for the purpose of Option A and Option B.

4.3.1 Generalization sequence – Option A

For that purpose vario-scale maps Python Software Development Kit was used, including the solution which was considered as state-of-the-art for the whole project. It makes it possible to create a generalization sequence for given data, however, it does not offer a possibility to define geometrical boundaries of the goal map in order to use it as constraints during processing. This problem was solved by an additional script which was added to the existing solution which makes iterative provision of specific objects on the goal map paired with objects on the initial map and builds the same area as well as creates the sequence separately for each of them. The final structure was acquired by unifying the results with the results of the process of renumbering objects to new identification numbers to keep them unique. It was done by summing the identification number of each created polygon multiplied by 1000 with an identification number of the paired polygon on the goal map as shown on the Figure 4.4 – the number of object on the goal map is expected to be significantly smaller than 1000. In order to provide information about class on the goal map, two tables were provided, one with information about region and linked class and the other with information about class similarities.

For every pair of coupled objects with linked classes (children), the algorithm chooses a class with a higher similarity to the class of linked object on the goal map. This approach will allow to make sure that the final map created by generalization will look the same as the goal map – every object on goal map is linked to the class which was the most common in the area which this object represents. This means that there is at least one object with the same class as the object on goal map.

	face_id integer	imp_low bigint	imp_high bigint	step_low bigint	step_high bigint	step_high_sub bigint
1	8072065	90	135	45	46	46
2	8185065	92	138	46	47	47
3	8214065	94	141	47	48	48
4	8381065	96	144	48	49	49
5	8382065	98	147	49	50	50
6	8548065	100	150	50	51	51
7	8557065	102	153	51	52	52
8	8559065	104	156	52	53	53
9	8708065	106	159	53	54	54
10	8723065	108	162	54	55	55
11	8847065	110	165	55	56	56

Figure 4.4: Part of a table with results for Option A related to object 65 on the goal map, renumbered to keep identifiers unique. Object 65 is the object which involves the most of objects from the initial map in generalization sequence.

4.3.2 Generalization sequence – Option B

For Option B, the currently existing implementation made by Dr. Peng in C was used. It uses ArcGIS Engine as well as ArcGIS Objects SDK. In order to process the data the user needs to prepare it as layers included into ArcMap project file and insert the path to the file into the software. Every geometry has to be exported to shapefile format as well. The GUI implemented by Dr. Peng makes the operation trivial, limiting it to a simple choice of a specific functionality by clicking on its name. First, the initial map has to be referred to the goal map by a simple spatial join and its results are saved as an additional column in the table of content of the initial map layer. With the prepared relation, the algorithm was able to process the data. It creates a CSV file containing information about the sequence, as shown in Figure 4.6. In order to produce the map, tGAP model still needs to be created. For that purpose a Python Language script was proposed which is reconstructing the geometries and the topological relations between objects in a similar manner as the solution already implemented in Vario-scale maps Python Software SDK – the resulting tables need to be similar since this software is also necessary to produce the Space Scale Cube in further steps. The software connects with the existing database and creates the proper tables with the tGAP structure using the table with information about the initial map as well as the CSV file created by applying the algorithm. In Figure 4.5 it is possible to see the example of the resulting sequence (one sub-tree, a generalization sequence leading to one specific object on goal map) for Option B.

4.3.3 Generalization sequence – Option C

For Option C, which did not require provision of a goal map, the generalization sequence was acquired by using Vario-scale maps Python Software Development Kit, similarly to Option A. In this case there was no need to provide additional scripts in order to provide additional information. The approach used here (which is generalization with compactness taken into consideration) required the preparation of distance matrix to assess the compactness between two object based on their classes, in a similar way as the table used for Option A. It is important to note that in the generalization procedure the tGAP created in Option C should not include the whole sequence created in that step – for Option A and Option B the algorithm stops generalization with 590 objects left. The same number of objects will be left with Option C in order to be able to compare them more accurately.

4.4 Parallel step assignment

With every generalization sequence acquired, it is now possible to move to the next part of this research – parallel step assignment. At first, Option A and Option B were taken into consideration. In order to renumber objects with given rules, simple Python Programming Language script was proposed, to alter information about step assignment in a PostgreSQL table. Option C does not require this step.

In order to acquire final step assignment, additional processing needs to be done on the given data – this is done with the greedy algorithm. As described in Chapter 1, the algorithm needs to take into consideration the problem with shock changes. In order to do that, a Python Language algorithm was proposed. At the beginning of assignment to step X, an empty dictionary is created to store information about the features already assigned to the step or discarded from candidates for step X assignment because of geometrical relation to object assigned to specific step and shock change rule. The algorithm connects to the PostgreSQL database and iteratively traverses the list of objects in a given table, ordered by the initial event assignment, which were not assigned yet to any final step. For every object, a list of objects is created describing each object and the algorithm checks if any of them appear in the dictionary mentioned above. If not, the algorithm assigns the object to step X, adds all the objects from created list to the dictionary and moves on to next object. If yes, step number is increased by one, the dictionary is cleared and the same steps are followed – assignment to step $X + 1$ and locking objects by addition to the dictionary. As it was mentioned in chapter 3, in order to avoid the situation with too many changes being shown (animated?) on the map, various criteria for maximum number of steps will be proposed and after the examination of results in a web viewer a conclusion regarding this aspect will be drawn.

4.5 Smooth Space Scale Cube

Current solutions for Space Scale Cube and its smooth version are implemented and working well for simple, non-parallel approach of tGAP and its current implementation. However, it requires the provision of both edges and faces in a topologically consistent form, which is a challenging problem for Option A and Option B. Because of the iterative approach, every object on the goal map is represented separately and there is inconsistency in the edges table which makes it impossible to process the data in the current implementation. For example, every edge describing the boundary of a specific element on the goal map is not considered as the edge describing the boundary of the whole map, which means that one of the faces describing the left or right side of the edge will be equal to 0 (Figure 4.8). This confuses the algorithm. The current implementation is experimental and even minor differences in input details can lead to problems which are challenging to recognise.

However, a consistent topological data structure is only needed to generate a geometrical object representing the Space Scale Cube – this information is no longer important once the cube is created. With that assumption, it is possible to simply create separate cubes for every object and glue them together to create one consistent structure. This task was also carried out in the Python programming language, since data structure of .obj files is very straightforward – it contains textual information about faces, vertices and lines. Then the merge of objects is done by appending the same elements together, renumbering them to keep the indices unique and saving in a new file. The resulting file will contain redundant information about edges which are doubled on the borders of object on goal map and there is no need to keep them, however it should have no impact on the way the results are assessed in this graduation project. Example of this merge is shown on Figure 4.9.

4.6 Benchmark

4.6.1 Accuracy assessment

As described in Chapter 3, there is a need to assess accuracy of maps created with every option proposed. Provision of the goal map can help with step assignment, however accuracy of used solution is not known and its suitability for practical application needs to be checked. Current state of art, greedy algorithm with compactness criteria, is considered as valid solution for generalization of areal objects. This assessment will be focused on a similarity of approaches which use a goal map and those which use no goal map, where similarity is described with respect to maps used as benchmarks.

For the part of assessment related to information preservation, the initial map will be used as a ground truth. It was considered to use raw dataset which was used to create initial map, but this idea was discarded because it does not help to answer the question. There were many operations performed between the raw input and the initial map, with a high impact on that matter. With the assumption drawn in Chapter 3 and Equation 3.2, it is possible to calculate the number of operations necessary to create a map at a specific scale. Note that the number of objects should be lowered by the number of objects lost in the process of initial map preparation. The resulting values are shown in Table 4.3. The *Operations* column already takes this information into consideration. Note that the values refers to the optimal number of objects and it is unlikely that generalization with step assignment will be able to reach the specific number given in the table. Once the parallel step assignment is finished, the most suitable value will be chosen and a conclusion about the actual number of objects at the specific scale will be made.

Scale	Objects [%]	Operations
1:15000	44.4	2207
1:20000	25	4781
1:25000	16	5972
1:30000	11.1	6620
1:35000	8.16	7010

Table 4.3: Scale proposed as benchmark of the dataset. "Objects" refers to the number of objects left with respect to raw data and "operations" denote the number of operations necessary to perform in order to transform the initial map into the map at given scales.

With all the information mentioned above it is possible to make a conclusion about both of the assessments in confusion matrix, which is the structure used to summarize the classification performance of a given classifier – in that case, road rings approach in Option A and Option B and the greedy algorithm in Option C. It is a two dimensional matrix with true classes of objects (in that case, Accuracy Assessment Points with classes from benchmark maps), indexed in the first dimension and in the second dimension by the class acquired with the classifier. In every cell C_i its value C_v refers to the number of AAPs which should be represented by class given by row number index but it resulted in class given in column number index. This approach makes it possible not only to make a conclusion about the accuracy of the classification, but also helps to identify misclassification patterns by providing specific information about the number of misclassifications for every possible pair of classes.

4.6.2 Parallel assignment assessment

As described in Chapter 3, parallel step assignment needs to be assessed on two ways – globally and locally. Obviously, the first idea for parallel step assessment is to compare the number of steps required by the algorithm to put all the objects in, and this is the value that is going to be assessed and compared. However, this approach will make it impossible to assess if the results will be compatible with conclusion about the same time for display for every step. In both cases, the final analysis will result in a histogram describing the number of objects and the total area of objects in every step. With the assumption that in optimal situation every step should contain the same number of information, the variance of the resulting histogram can be checked. This is defined as the squared deviation of the number of objects in the step from its mean. Comparing this value, especially between Option A and Option C, will help to give an answer about the impact of the goal map on parallel step distribution, both locally and globally.

Three places were chosen for local assessment, fairly distributed over the extent of the map. For each of the points, two circles describing the area of interest of the user were defined at a distance of 1.5 km ($c1$) and 3 km ($c2$) from the point of interest (POI). A histogram will be built for each of the circles, describing the parallel step assignment, with the constraint that the objects needs to intersect with the circle, with a boundary given by the rings geometries. Initial map with selected places and circles is shown on Figure 4.10;

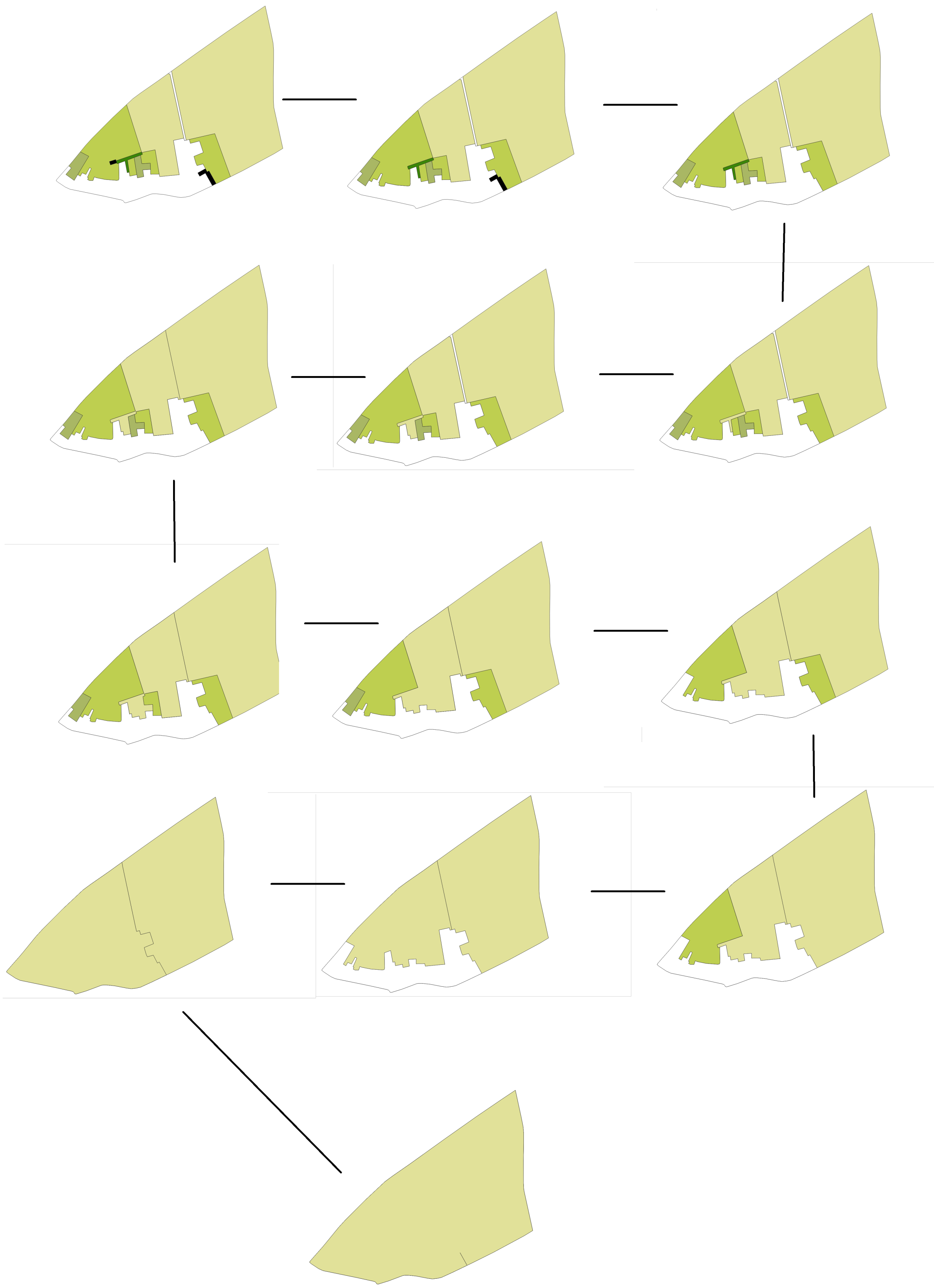


Figure 4.5: Generalization sequence for Object 502 on the goal map acquired in Option B (A* algorithm)

ID	child 1	child 2	passive	active	class	last
2414	3764	3750	3764	3750	14010	not_end
2415	1194	1205	1194	1205	14130	not_end
2416	250	214	250	214	14130	not_end
2417	735	634	735	634	14130	not_end
2418	192	210	192	210	14010	not_end
2419	2034	2695	2034	2695	14130	not_end
2420	3459	3410	3459	3410	14130	not_end
2421	3369	3342	3369	3342	14130	end
2422	1047	1116	1116	1047	14010	not_end
2423	2553	2668	2553	2668	14010	not_end
2424	2696	2678	2696	2678	14010	not_end
2425	2330	3845	2330	3845	14010	end

Figure 4.6: File describing the generalization sequence acquired with A* algorithm. ID denotes the operation number, child 1 and child 2 refer to candidates for a merge and "active" denote the object whose class will be assigned to a newly created polygon. Column *last* denotes if the polygon is the final polygon on the goal map. Note that the algorithm does not provide numbering for parents and keeps the ID of an active child as the ID of a parent. This needs to be taken into consideration in order to avoid problems with data consistency in further parts of the research.

	10310	10750	10410	10411	10510	10600	10700	10710	10720	10730
10310	0	0.33333	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10750	0.33333	0	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10410	0.66667	0.66667	0	0.33333	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10411	0.66667	0.66667	0.333333333	0	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10510	0.66667	0.66667	0.666666667	0.66667	0	0.66667	0.66667	0.66667	0.66667	0.66667
10600	0.66667	0.66667	0.666666667	0.66667	0.66667	0	0.66667	0.66667	0.66667	0.66667
10700	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0	0.66667	0.66667	0.66667
10710	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0	0.66667	0.66667
10720	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0	0.66667
10730	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0
10740	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10741	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10760	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
10780	0.66667	0.66667	0.666666667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667

Figure 4.7: Part of the distance matrix used for compactness criteria for the greedy algorithm used in Option B and Option C. For Option A, the largest neighbour rule will be used (since the initial classification will be changed later and because of that the initial choices for the most suitable neighbour could be invalid). The values describe similarity between them.

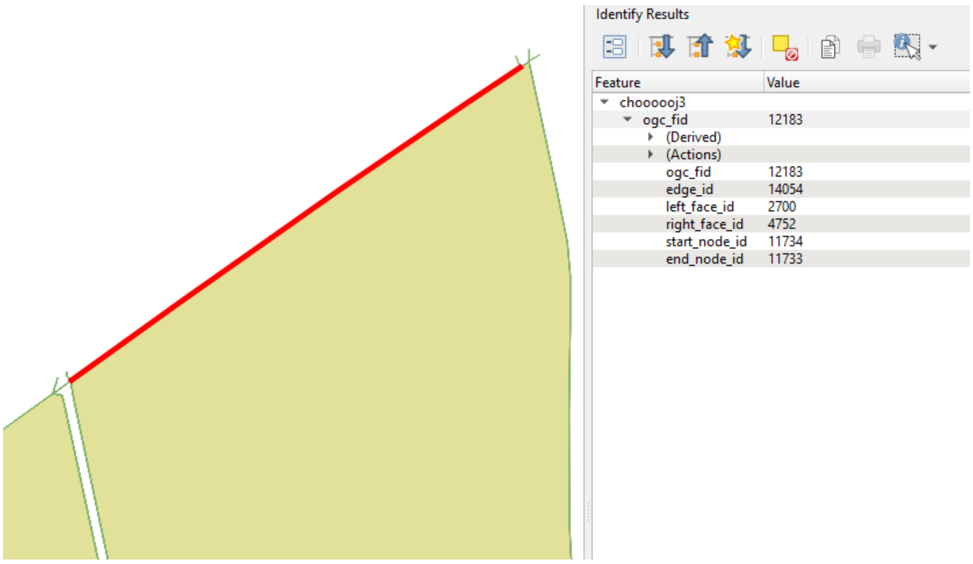


Figure 4.8: Example of inconsistency of edges caused by an iterative provision of the goal map objects for algorithms. It is possible to see that in the table the edge is related to the object which is not present in this part of the dataset. Since part of the implementation is the iterating table with edges, it is very important to keep this aspect consistent while using this information.

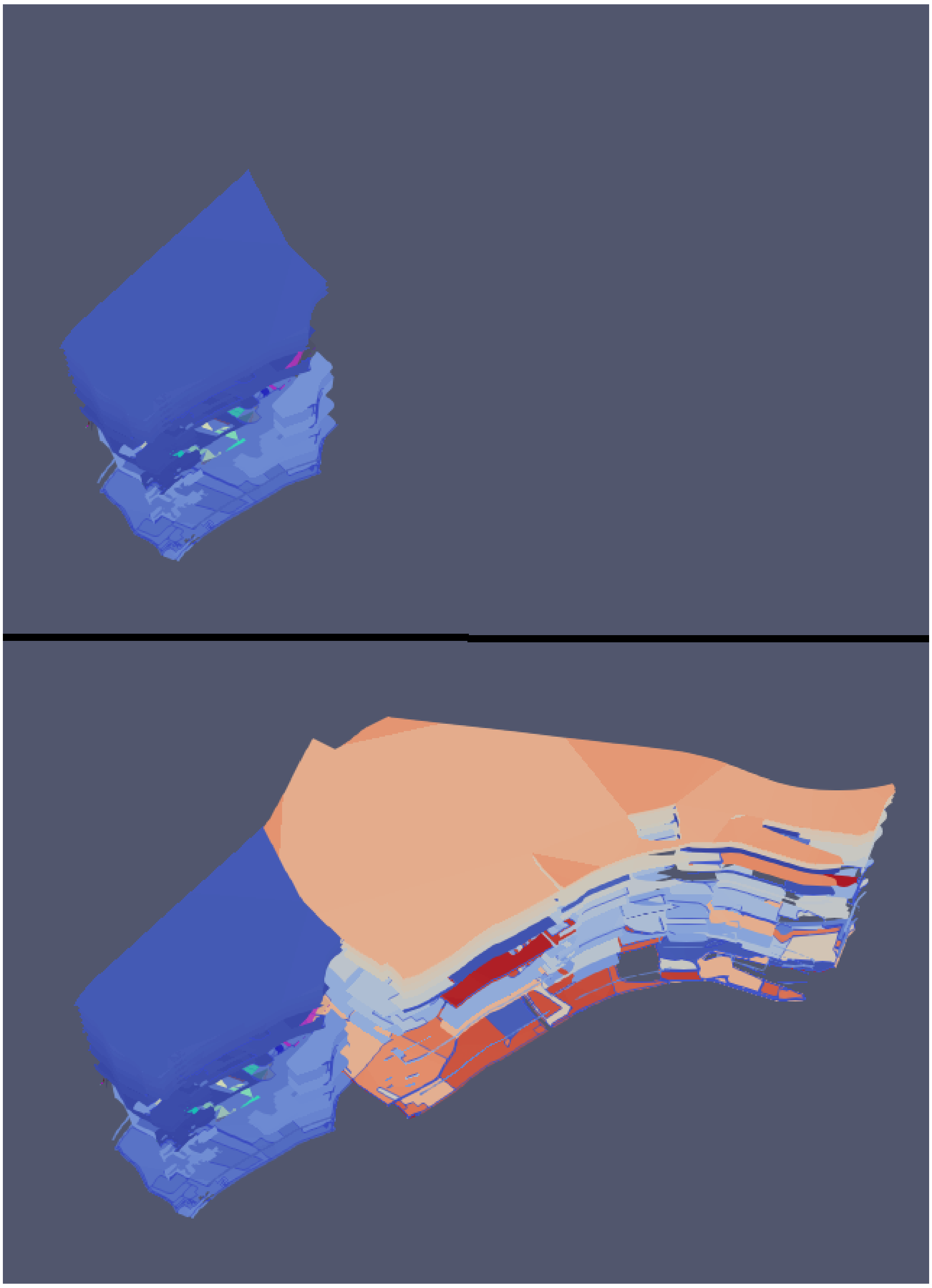


Figure 4.9: Example of a merged SSC representing two objects on the goal map.

Figure 4.10: Chosen places for the local part of parallel step assessment shown over the initial map extent together with circles denoting the range of interest.

5 Results and analysis

In this chapter the results acquired in whole graduation process will be presented and evaluated in order to find answers for research questions. At first, parallel step assignment will be presented for Option A, Option B and Option C on every step of results acquisition. In the second part, the assignment process will be evaluated in order to conclude about applicability of chosen methods. In the third part the results necessary to evaluate the accuracy of presenting information will be shown, since it is an essential application for the map and parallel step assignment cannot be prioritized as a more important functionality. The final part will evaluate the results in areas mentioned above as a whole to find the answers to the questions from Chapter 1. Final results showing actual map created from the given dataset will be available online in the form of a web viewer.

5.1 Results – parallel step assignment

5.1.1 Initial conclusion regarding parameterization

For each option proposed in this graduation project all the steps discussed in Chapter 3 and Chapter 4 were applied successfully. Each of them was tested in a web viewer in order to answer the question about the optimal choice of the biggest possible area which can be assigned to a single step and the conclusion was that for 1% it is almost not possible to see the smooth transition in generalization process. At the same time, for 5% and 10% the results were comparable from the perspective of the user: it was possible to see smooth transition in both of the options. The user was not getting confused in any of these cases which leads to conclusion that both 5% and 10% can be proposed as values which make it possible to see the transition without confusing the user. In order to answer all questions regarding this graduation project, 5% was chosen as a suitable parameter.

5.1.2 Option A

The first tested option, which is the greedy algorithm with the goal map provision, resulted in 590 sub-trees describing the sequences of generalization for every object on the goal map. The maximum number of objects linked to a single object on the goal map was 163, which is equal to the number of initial parallel steps because of the chosen methodology. With the proposed greedy algorithm and rules about maximum area, the final number of steps is 155. Histogram describing distribution is shown in Figure 5.1 and the total areas of changed objects in every step are shown in Figure 5.2.

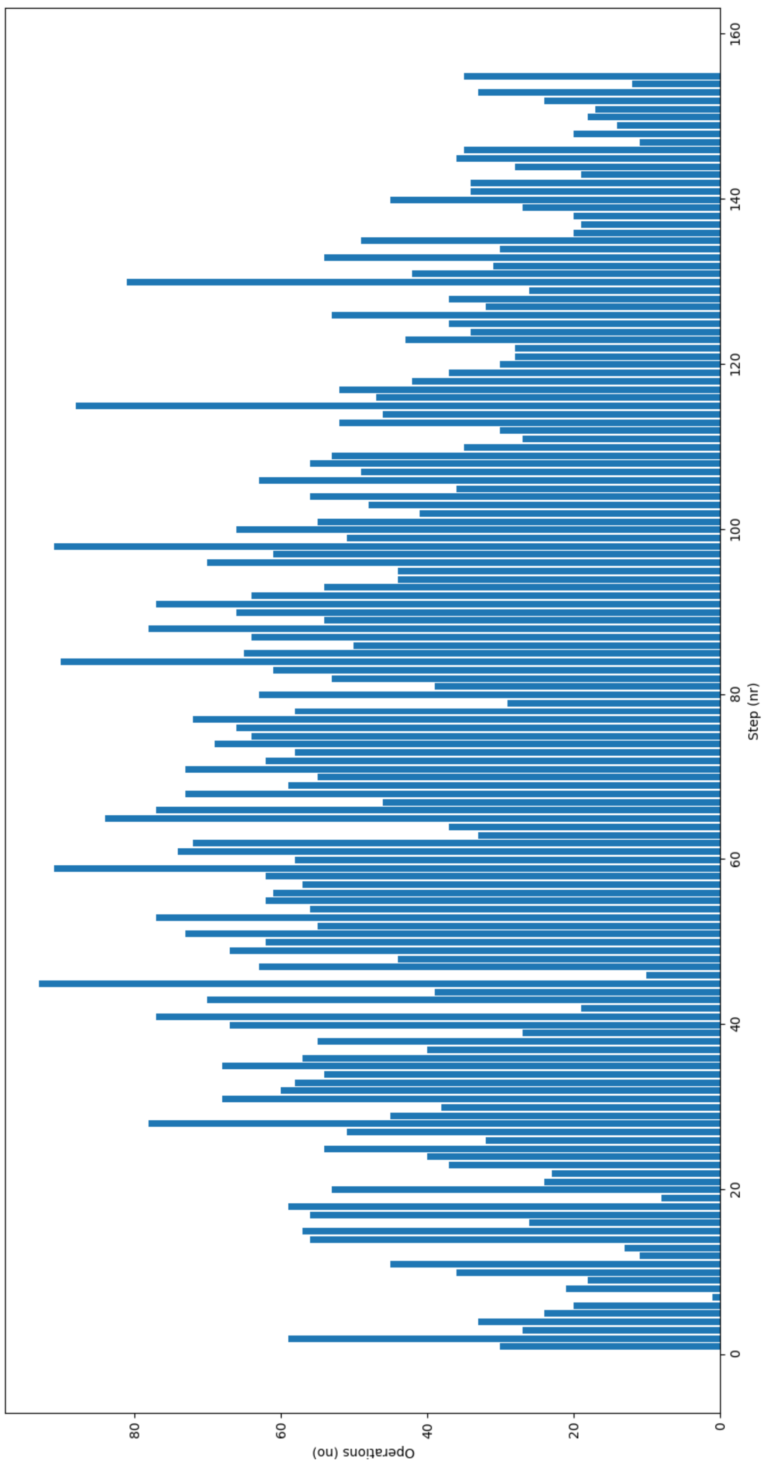


Figure 5.1: Histogram describing number of objects assigned to every step after the application of the greedy algorithm in Option A.

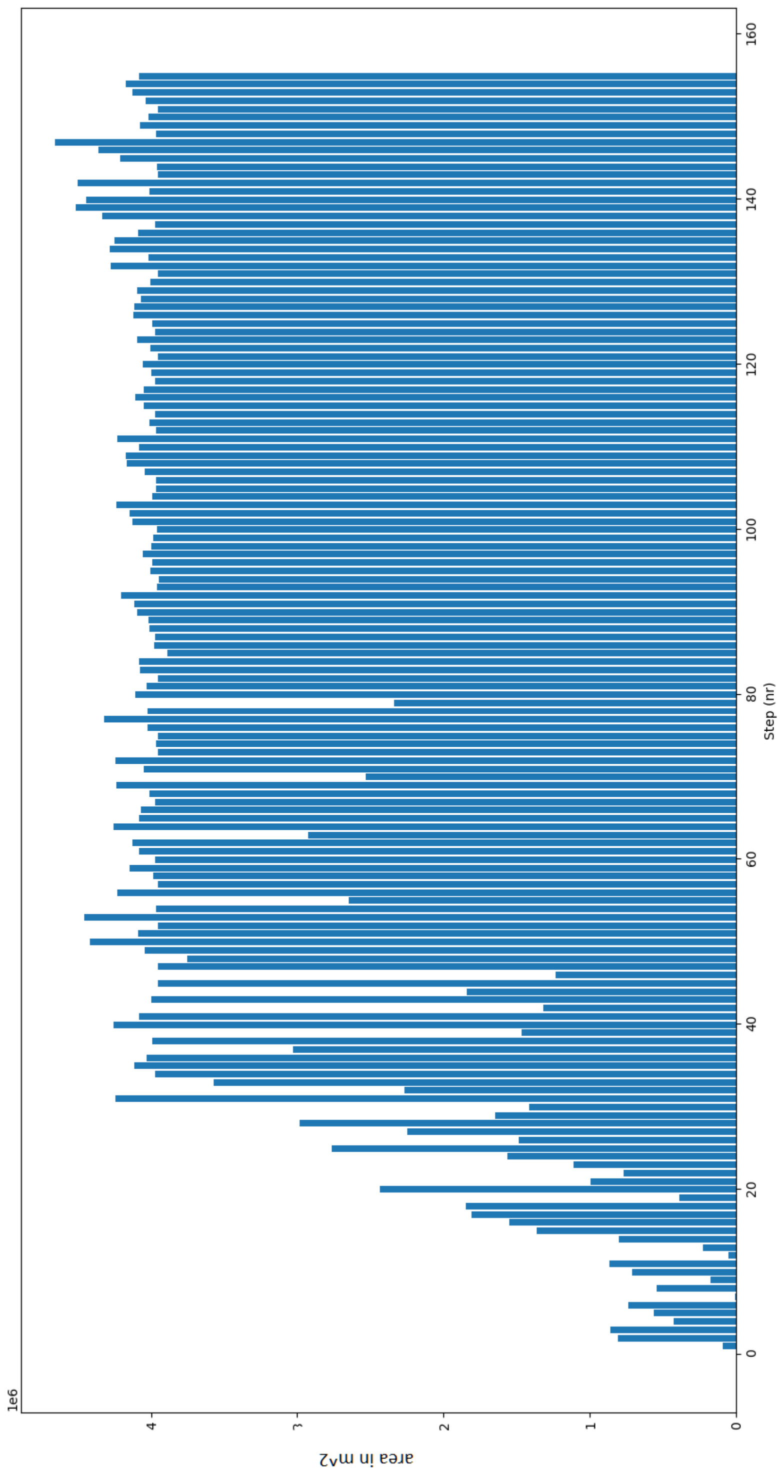


Figure 5.2: Total areas of objects for every step in Option A after the application of the greedy algorithm.

5.1.3 Option B

Since Options A and B had the same principles regarding provision of goal map, which means that the number of objects on the initial map related to specific objects on the goal map will be the same. Therefore, the initial number of parallel steps is the same and equal to 163. The sub-trees of tGAP structure were processed by greedy algorithm with 5 % parameter, resulting in 197 steps. A comparison between histograms describing the distribution of parallel step assignment from the initial assignment and from the final assignment is shown in Figure 5.3. In Figure 5.4 it is possible to see the total area of objects shown in every step.

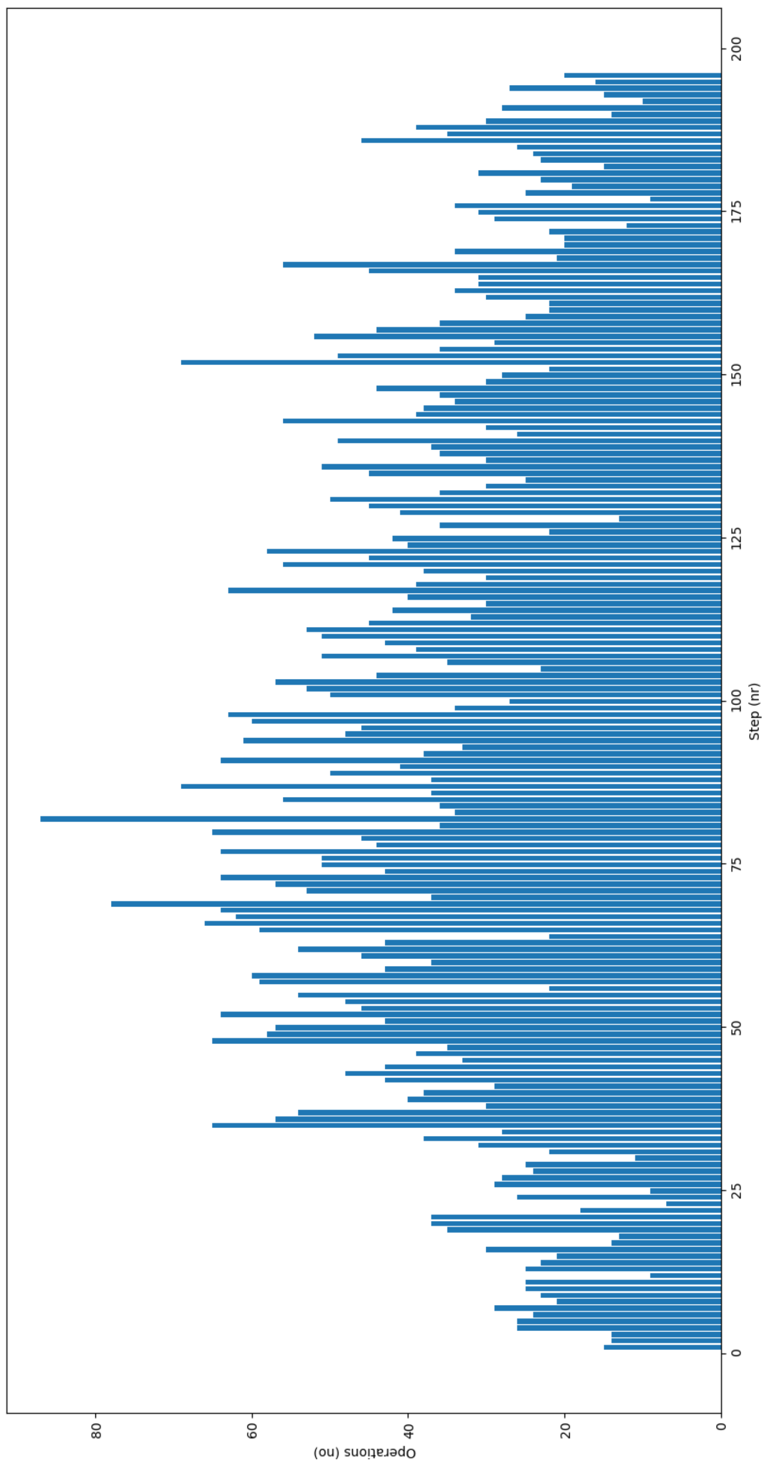


Figure 5.3: Histogram describing the number of objects assigned to every step after the application of the greedy algorithm in Option B.

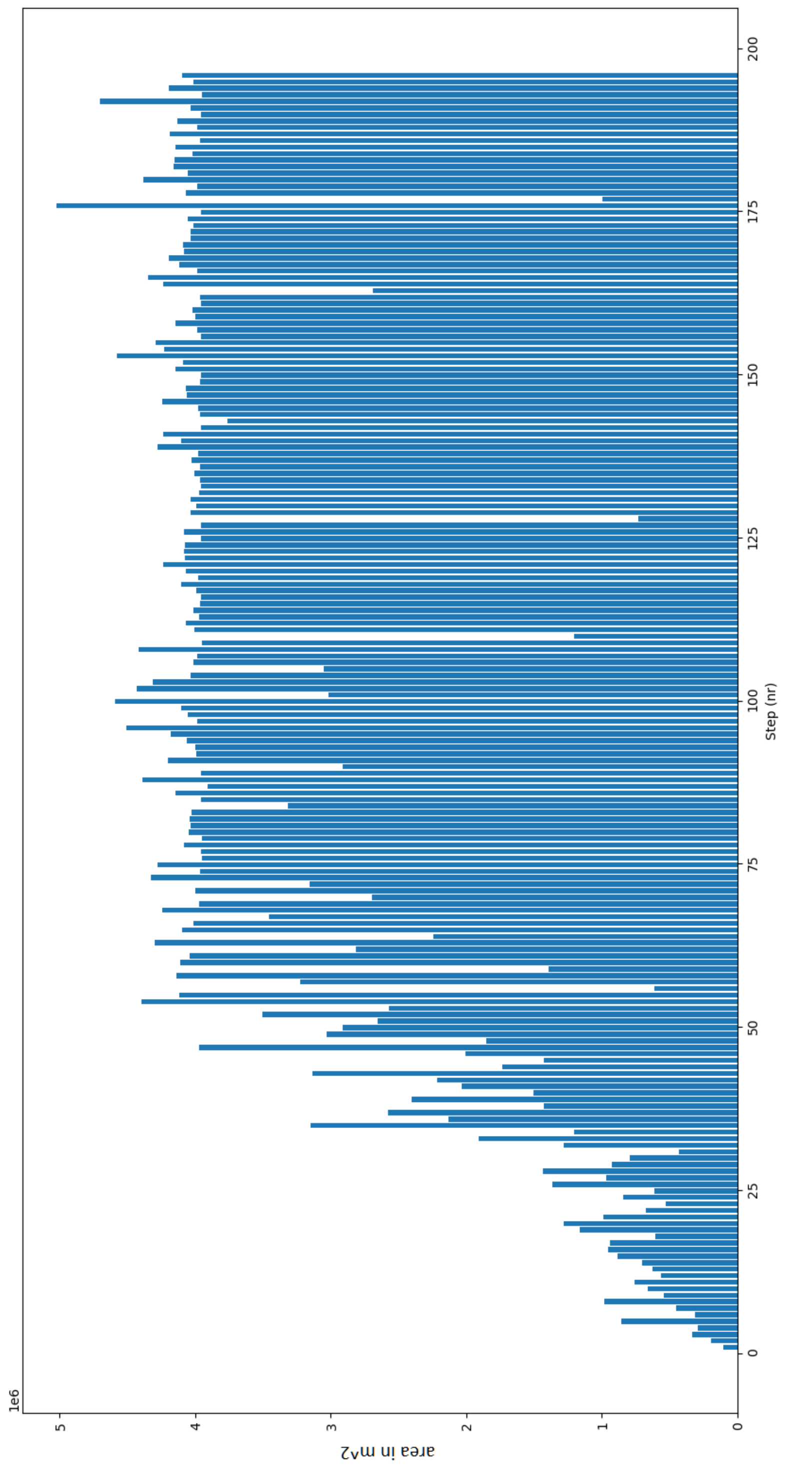


Figure 5.4: Total areas of objects for every step in Option B after applying the greedy algorithm.

5.1.4 Option C

For Option C the initial sequence of 7501 operations leading to 590 objects left on the map (the same as on the goal map used for Option A and Option B) was processed with the same algorithms as for the previous options, resulting in 183 parallel steps. The resulting distribution as well as the total areas of objects assigned for specific steps are shown in Figure 5.5 and Figure 5.6.

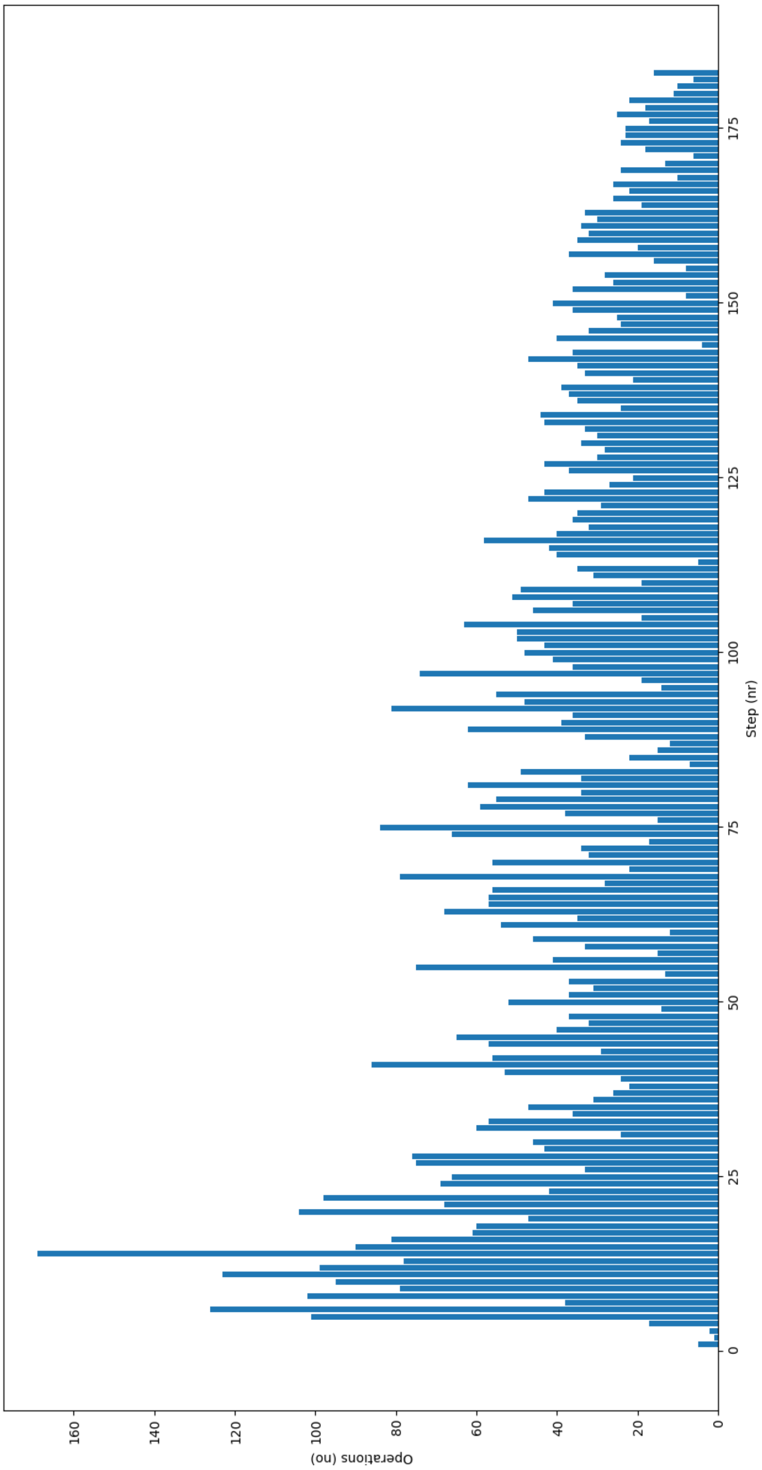


Figure 5.5: Histogram describing the number of objects assigned to every step after applying the greedy algorithm in Option C.

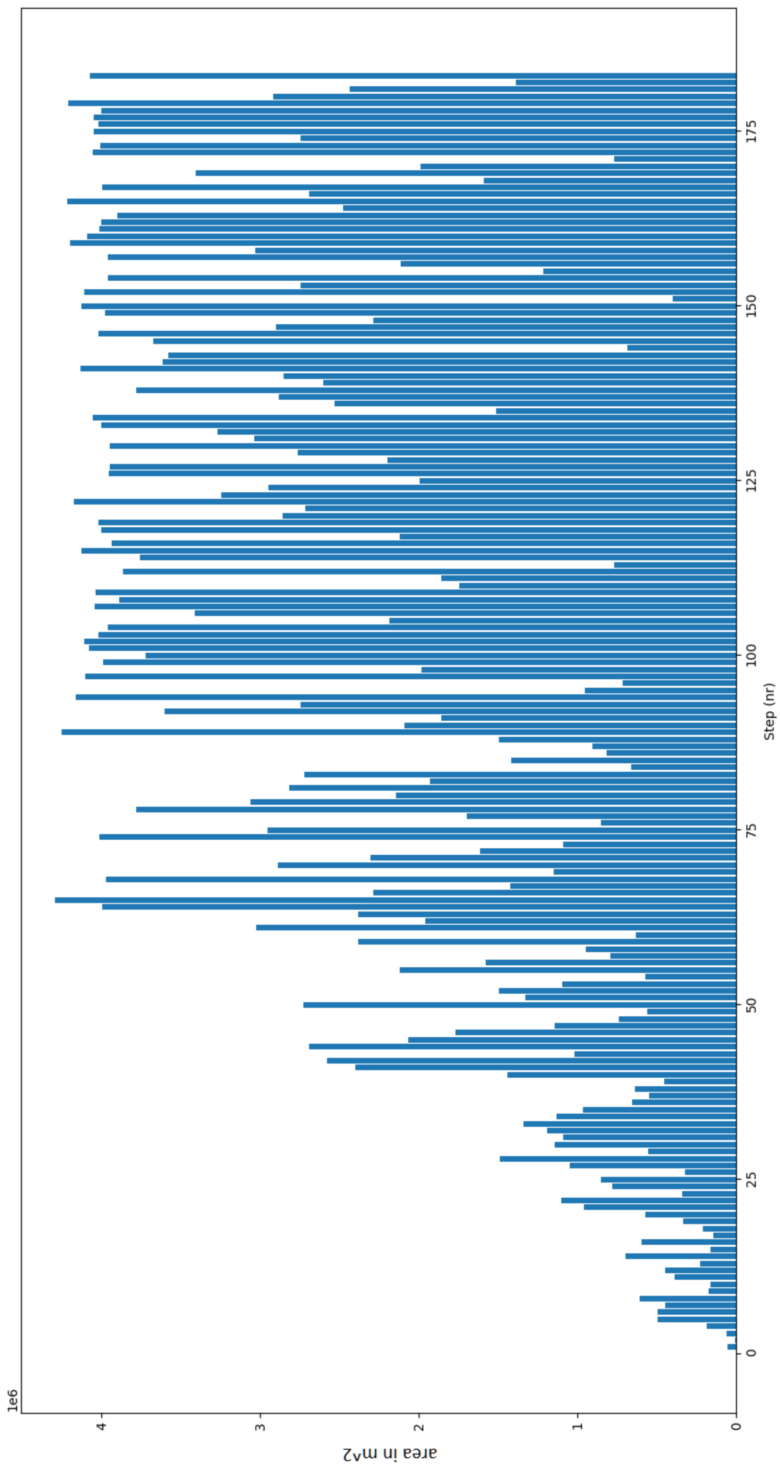


Figure 5.6: Total areas of objects for every step in Option C after applying the greedy algorithm.

5.2 Results – accuracy and assignment assessments

With number of steps assigned to each proposed option, it is possible to perform the assessment of this graduation project, which is crucial to answer the main question of the research. In this subsection two aspects will be taken into consideration, namely the parallel steps assignment for every of them and the accuracy of the maps created in each of tested approaches.

5.2.1 Parallel step assignment assessment

First, variance was calculated for every option. It is shown in Table 5.1 and since the number of steps differs among the analyzed options, the value was normalized to variance-to-mean ratio (VMR) by dividing it by mean area of objects assigned to a single step in each option. For better readability the values were divided by 1000 and rounded to 1.

Option	Variance	VMR
Option A	1609553	476
Option B	1845368	576
Option C	1439192	761

Table 5.1: Variance and VMR for all every tested option.

The second part of the assessment was to evaluate preservation of spatial distribution of assigned objects in different parts of the map. For every point two results were acquired (one for each circle) and a plot with the total area of objects in specific steps was created for every tested option, as shown in Appendix **??**. Variance and referring VMR values are shown in Table 5.2.

Point	Circle	Option A		Option B		Option C	
		Var	VMR	Var	VMR	Var	VMR
Point 1	c1	23.45	3.99	19.63	4.43	27.27	7.93
	c2	88.12	4.03	78.29	3.06	35.13	6.45
Point 2	c1	22.7	3.92	24.63	4.47	56.28	9.81
	c2	75.18	3.62	74.04	3.17	43.7	5.97
Point 3	c1	25.53	4.22	32.11	6.54	44.80	10.56
	c2	74.45	3.62	88.47	3.98	29.7	5.98

Table 5.2: Results of local parallel step assignment assessment. Note that for Option A and B the total area of the assigned object is larger – the variance values are significantly higher but the ratio to mean is usually lower.

.

5.2.2 Accuracy of information presentation

First, using the information from Table 4.3 and information acquired in the process of parallel step assignment, Table 5.3 was created to show the specific number of operations and number of steps that are necessary to complete in order to receive a map at specific scales:

With this information it is possible to create confusion matrices for each scale and option and, based on these matrices, to calculate the overall accuracy. All the matrices are shown in Appendix **??**. The overall accuracy for each option and scale was shown in Figure 5.7 and Table 5.4.

Scale	Option A		Option B		Option C	
	steps	obj.	steps	obj.	steps	obj.
1:15000	51	2246	65	2216	33	2233
1:20000	92	4810	117	4769	96	4787
1:25000	114	5938	148	5966	127	5962
1:30000	130	6633	166	6601	150	6619
1:40000	141	7004	182	7002	180	7005

Table 5.3: Scales with number of objects and steps for every option proposed in the research. The number of objects is different in every step because the number of events in various steps is not equal.

.

Scale	Option A	Option B	Option C
1:15000	94	91	99
1:20000	88	80	97
1:25000	78	69	93
1:30000	72	63	89
1:40000	65	62	84

Table 5.4: Accuracy of results described with a percentage of correctly classified sample points.

.

5.3 Analysis

This section will evaluate all the presented results in order to answer all questions of this research. First, the parallel step assignment will be analyzed in order to evaluate its performance and in the second part the information presented on the map will be evaluated in order to assess the applicability of the proposed solution for the vario-scale project.

5.3.1 Parallel step assignment

First, details regarding the parallel step assignment will be examined.For every tested approach it was possible to achieve a significant compression C of generalization sequence among steps, as shown in Table 5.5 – the smooth transition is possible to observe in each of the tested cases and from the perspective of the user, the shock change rule was preserved. The compression is expressed as ratio between the number of generalization steps in step by step procedure (7501) and the number of parallel steps in each tested option.

Option	C
Option A	48.4
Option B	38.1
Option C	40.99

Table 5.5: Compression of generalization steps in each tested option.

.

As it was concluded in Chapter 3, the optimal distribution of objects per step is the equal area (represented as the percentage of the whole area of the map) distribution which is preserved both locally and globally. As it was expected, providing information from the goal map made it possible to distribute the objects more equally among different steps. The difference is significant, especially between Options A and B, for which VMR values of respectively 476 and 761 are describing 37 % and 62% of mean as average deviation from mean total area of objects for a step. This result is understandable since without any

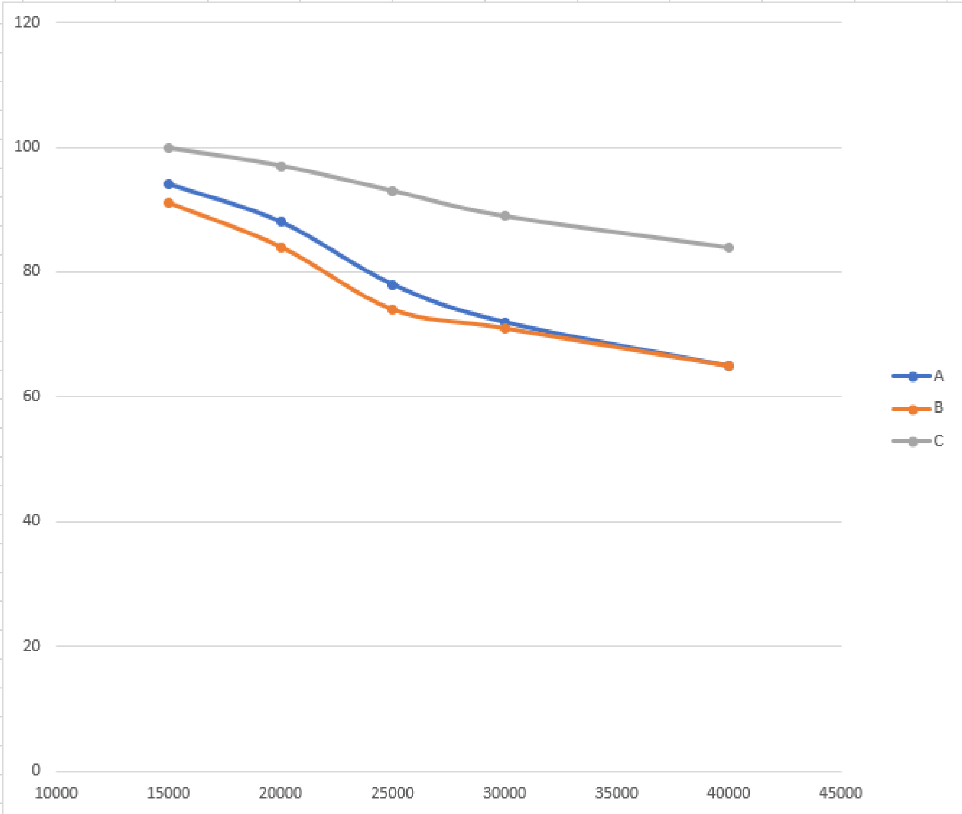


Figure 5.7: Accuracy of information presentation with respect to the initial map for every option tested for the purpose of the graduation project.

geometrical constraints it is possible that even in the most unfavorable spatial distribution one specific object and its consecutive parents can be used in every operation on the map if it is sufficiently large and complex. Besides that, it is also possible to see that such an unfavorable situation from the point of view of parallel step assignment which is involvement of a parent as a child in two operations close to each other in generalization procedure - tends to happen more often in case of Option C – the histogram describing the total area of objects per step shows significant drops in the second part of the distribution and as a result some steps are missing a big part of the objects (Figure 5.8).

Besides general distribution of objects, providing the goal map has impact on preservation of that distribution – in both of the Options A and B it was preserved to a bigger extent compared to the unconstrained approach. In case of point 2 $c2$ the results are not that favorable for Option A and Option B, however it can be justified by and unfavorable road distribution next to city centers – the more dense the road web, the smaller the objects trapped in every road ring.

Another difference that can be observed is that, because of initial renumbering of steps for Option A and Option B which makes them stacked at the end of generalization, the approach will not start with the smallest polygons (slivers) in first steps – it depends more on the size of the sub-tree to which the objects are assigned. Because of that, a simple pattern can be observed – in steps at the beginning of the generalization process, Option C is processing even more than 100 objects in one step with the total area which cannot be considered a significant part of the map. However, these objects can cause a significant degradation of distribution because many of them are incorporated to very big polygons in the generalization procedure, which makes them locked for the step. However, during the analysis of a related histogram (using a web viewer) it was observed that the shock change rule is not valid in those cases because it is not possible to see these polygons. The reason for this is that some of them are the size of 0.15 m^2 and so they have no impact on the shape and size of resulting parent object in the web viewer.

The difference between Option A and Option B is also noticeable, however it is not as significant as the difference between the two of them as a pair and Option C. It is still possible to observe that the greedy algorithm performed better in that matter and had lower variance just as it was in the case of global and local distribution analysis.

5.3.2 Accuracy of results

In order to evaluate the applicability of the proposed solutions, the assessment focused on the most important application – the presentation of spatial information. As it was explained in Chapter 3, a newly proposed method using road rings to help to constrain spatial distribution of objects can improve parallel step assignment and cartographic quality, however it changes the principles of map generalization which are currently considered to be state-of-the-art and because of that its accuracy needs to be checked.

The test conducted here was a comparison of the maps at different scales created for each tested option with the initial map. The aim of this comparison was to check the preservation of classification in the generalization process. It is clear that the newly proposed method used in Option A and Option B is in general significantly less accurate than the typical solution which was implemented in Option C since resulting accuracy is significantly lower. The most important reason is the change of the main criteria for a candidate for the next consecutive generalization operation, which is the lowest importance of the object based on its size among objects left on the map. In case of Option A and Option B this principle was only preserved locally in each of the sub-trees representing generalization of specific object on the goal map. The impact of this is the biggest at the beginning of the generalization procedure because instead of slivers, normal polygons are considered as the ones that should be incorporated first. This can be seen in Figure 5.7 – the accuracy dropped significantly more between bigger scale maps among proposed scales.

It is important to mention that, through the interaction with the map, it can be clearly concluded that some parts of the map are preserved better with a goal map than with no goal map, especially in the parts with very dense road structure such as cities or other populated areas (Figure 5.9). In those cases the algorithm helped to preserve the structure which is also possible to notice on confusion matrices – Option A and Option B preserve the information about urban areas better than Option C.

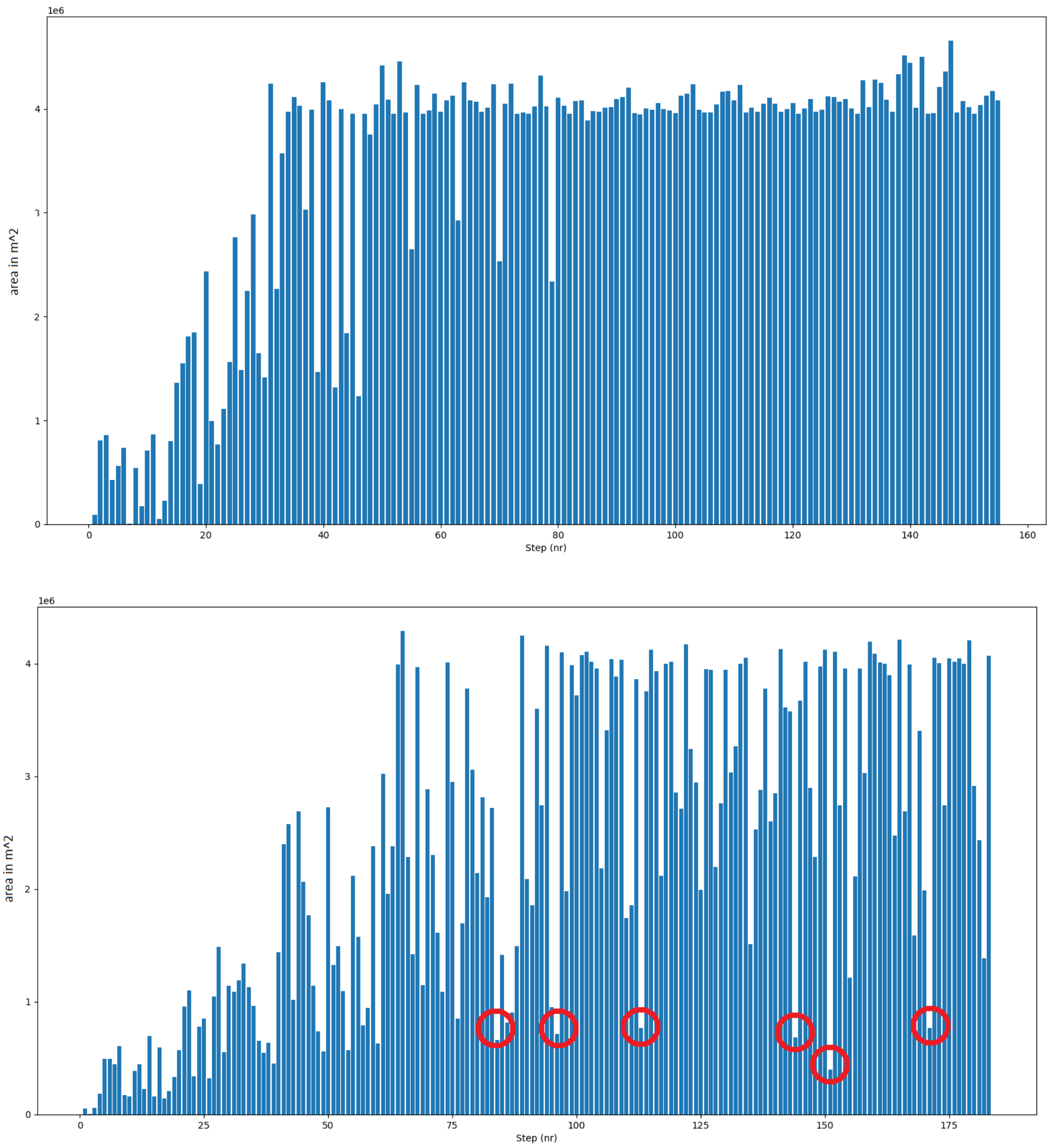


Figure 5.8: Comparison of histograms between Option A and Option C. Note that the drops appear more often in case of Option C because of lack of geometrical constraints. Some of the parallel steps are almost skipped.

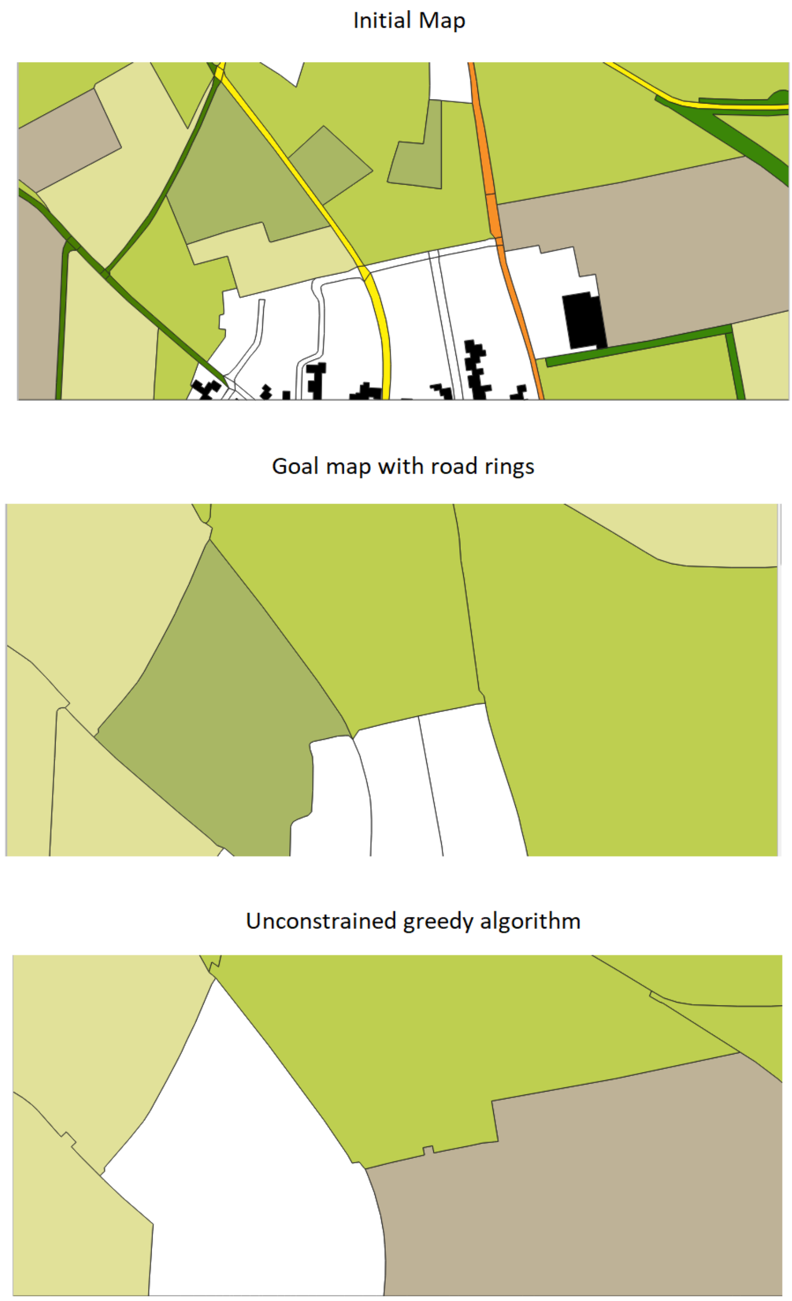


Figure 5.9: The difference in preserving information between the provision of the goal map and no provision of the goal map. The result acquired from the unconstrained algorithm is considered to be worse than the prepared goal map.

6 Conclusions

In the final chapter of this graduation project all the research questions are answered or reviewed in order to assess to what extent it was possible to address them. After that, the contribution of this graduation project is evaluated with respect to the current state-of-the-art solution and a recommendation is provided for further work and research. All the animations as well as the web viewer used for answering the question are available online as software repository.

6.1 Research overview and discussion

At the beginning of this research, after the topic of it was defined, the list of sub-questions was created as an essential part of the problem description. This process made it possible to find a clear answer for the stated problem. For each of these sub-questions, an answer will be provided in this chapter together with a justification and corresponding conclusions acquired in this research. These answers are supported by the results and analysis acquired and evaluated in Chapter 5. At the end of this section the main question of this research will also be answered.

Question 1 Is it possible to see smooth transition of a map with the proposed improvement?

Yes, it is possible to see smooth transition with all the tested solutions, and for all of them it can be concluded that the proposed solution improves the interaction between the map and the user in the predicted way. In every approach, the display time of the transition in the merge operation was around 40 times longer compare to original approach. Thanks to applying the shock changes rule the user does not get confused and the interaction is significantly more satisfying compared to the traditional approach – the objects are not suddenly appearing on the screen and the user is not getting disoriented by them. The proposed restriction to avoid the shock changes problem seems to prevent the user from getting lost on the map.

Question 2 Is the goal map making the parallel step assignment distribution closer to expected?

Yes, provision of the goal map has a significant impact on the parallel step assignment and makes it possible to distribute the objects more equally among specific steps based on the chosen ruling. The difference is significant and the number of ”drops” caused by consecutive changes of geometrically related objects is also lower. It can be clearly concluded that the geometrical constraint equalizes the distribution of objects among steps. Nevertheless, it is important to mention that this difference is not very significant from the perspective of the user and this part of the experience was overall considered comparable in every tested solution from that perspective.

Question 3 Is the goal map making spatial distribution of steps preserved locally?

It is possible to see that the goal map is preserving it to some extent – clearly, the spatial distribution was better in both of the approaches involving a goal map compared to the classic approach. However, in one part of the map, many objects on the goal map were represented by the same object as on the initial map because of a dense web of roads in the city center and in that spot the distribution was generally distorted. It should, however, be considered as a consequence of the limitation of the chosen approach to generalization of the goal map rather than as a drawback of using a geometrical constraint in general.

Question 4 What is the accuracy of the result classifications with respect to the initial map? Is there a difference between tested approaches?

There is a significant difference in the accuracy of information preservation between tested options, as shown in Figure 5.7, however it is also possible to notice a positive impact of the geometrical constraint on that matter – especially in the areas with dense road network. Regardless of that aspect, it is still concluded that the chosen ruling for class of an object on the goal map cannot be considered as suitable for that task and a different method should be developed.

Question 5 What is the impact of the goal map on accuracy? Is this solution feasible to be used in the generalization process?

As written above, this question has to be answered from two perspectives. The first one one is the geometrical constraint of the dataset and the other is a rule for establishing classes of the goal map. It can be concluded that the geometrical constraint approach and its principles can be considered as suitable at this stage of development, however the ruling for the optimal choice of the final class needs to be changed.

Question 6 What should the display time of every step depend on?

Two main conclusions were drawn in order to answer that question. Since the map is processed tile-wise it was concluded that every step x should have the same weight in every tile and it should not depend on, for example, number of elements in it or the total area of objects – such an approach could cause inconsistencies in the way the SSC is traversed by the viewer.

The second conclusion is that the weights should be also the same for every step in every tile – in the optimal situation all steps should have an equal area of objects assigned to them. The algorithm can change the step to the next one only in two situations: when the total area is higher than a given limit or in case of a problem with intersection. The second situation is considered to be happening randomly and there is no way to predict in which steps exactly it will happen. If this value has to be fixed, this approach will make sure that the results will be as close to the actual distribution as possible.

Question 7 How to process the data for the purpose of web viewing?

One of the challenges driving this project was to establish a suitable solution for presenting parallel steps with the use of the current state-of-the-art solution developed for the purpose of a traditional, non-parallel approach. Because of the conclusion about the display time and the approach chosen for parallel step definition, it can be concluded that the current state-of-the-art method is suitable for processing SSC with parallel steps and there is no need to alter it. However, during the research many problems regarding consistency of the data were noticeable, such as the inconsistency of edges definition in case of Option A and Option B and it has to be taken into consideration if the user wants to use the currently implemented solutions.

Main research question What are the possibilities for a continuous generalization by parallel step assignment and how do they perform?

For years, efficient map presentation has been a challenging problem, but more recently, technological development supported by relevant research made it possible to proceed further in this aspect and test different ideas and proposed solutions. Starting with maps drawn on paper, through a simple presentation of its digitized image on the computer screen we reached the point when, instead of reading the map, it is possible to interact with it. This changed the picture completely and opened numerous possibilities for improvement. In this graduation project, an idea for parallel step assignment was presented as a proposition to improve the smooth interaction experience and several suggestions for the solution of that problem were shown and evaluated.

As written above, all the presented approaches were assessed in various ways in order to find the answer for the main question of the research and provide foundations and propositions for parallel step assignment as a way to improve the interaction with the map. The main conclusions refer to the display time, the use of a geometrical constraint as a support in equalizing spatial distribution and parallel step assignment as well as a proposition to apply a greedy algorithm which was tested for both constrained and not constrained solutions. Besides that, the current state-of-the-art approaches were investigated in order to evaluate the applicability of the proposed solutions in a related environment. It can be clearly concluded that parallel step assignment should be considered as a suitable way to improve the experience of a smooth transition for the user. The difference was significant for every tested option and there is no doubt that it is possible to present the changes in parallel with no confusion from the user perspective.

Moreover, it can be concluded that the geometrical limitation which makes it possible to distinguish separate sub-trees of tGAP structure helps to preserve the changes more equally among steps. This aspect can be particularly important in further research, in which many more aspects of generalization would be taken into consideration. Peng et al. [2020] proposes to exclude also the neighbouring polygons during step assignment which can have a big impact on its distribution. In this thesis it was not observed that this is necessary to keep the user more comfortable during the interaction, however it can change with the introduction of various elements and different operations such as line simplification or a split.

The accuracy of the proposed generalization method in Option A and Option B does not make it possible to be considered suitable for the task. However, Option C seems to be surprisingly effective in information preservation and further investigation about the applicability of this method for establishing goal map classes should be made. Obviously, preserving information is not the only aspect of the generalization that should be taken into consideration, and development of a suitable method of generalization assessment should be also considered at some point of the vario-scale project development. Together with the geometrical constraint, it can be possible to address the current drawbacks and find a method that can be considered as suitable for generalization and parallel step assignment in general.

6.2 Future work

This thesis proposes and evaluates a solution to improve interaction with vario-scale map. It analyzes different approaches and answers the main question of the research. However, there are several recommendations for the future which are considered as a good direction for research aimed at improving this experience.

- The shock change rule should be redefined in order to make it possible to proceed the data with many polygons included in one big polygon. The current state-of-the-art definition states that all of the operations which incorporate the smallest polygons into the big ones surrounding them need to be done in separate steps if the changes involve the same big polygon. However, this kind of changes cannot be considered as changes which make the user confused and it unnecessarily restricts the parallel step assignment.
- For real application, another method for target map acquisition should be developed, since the current one cannot be considered accurate when it comes to information representation.
- Specifically, it should be evaluated whether the proposed solution (especially a greedy algorithm used for parallel step assignment) is also suitable for various operation such as split and line simplification. For example Peng et al. [2020] proposed the exclusion of neighbours to make sure that morphing map will not confuse the user, however this solution has not yet been assessed with the operation of line simplification.
- Moreover, there is a need to find the best method of road representation in the SSC (Space Scale Cube). Currently, road objects are processed in the same way as other objects and it is clear that this solution should be adjusted to make it possible to show the road (for example as line segments with different thickness related to road type) on maps in small scales.

Bibliography

Burghardt, D., Purves, R., and Edwardes, A. (2004). Techniques for on the-fly generalisation of thematic point data using hierarchical data structures. In *Geographical Information Systems Research – UK*.

Cecconi, A. and Galanda, M. (2002). Adaptive zooming in web cartography. *Computer Graphics Forum*, 21(4):787–799.

de de Vries, M. and van Oosterom, P. (2007). Model generalization and methods for effective query processing and visualization in a WebService/client architecture. In *Spatial Data on the Web*, pages 85–106. Springer Berlin Heidelberg.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Frolov, Y. S. (1975). Measuring the shape of geographical phenomena: a history of the issue. *Soviet Geography*, 16(10):676–687.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Haunert, J.-H. (2009). *Aggregation in map generalization by combinatorial optimization*. phdthesis, Leibniz Universität Hannover, Germany.

Huang, L., Meijers, M., Šuba, R., and van Oosterom, P. (2016). Engineering web maps with gradual content zoom based on streaming vector data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:274–293.

Hägerstrand, T. (1970). What about people in regional science? *Papers of the Regional Science Association*, 24(1):6–21.

Jarocki, K. (2018). Land use recognition and ndvi analysis for monitoring crop fields in kuyavian-pomeranian voivodeship.

Meijers, B. M. and van Oosterom, P. J. M. (2011). 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.

Oehrlein, J. and Haunert, J.-H. (2017). A cutting-plane method for contiguity-constrained spatial aggregation. *Journal of Spatial Information Science*, (15):89–120.

Peng, D. (2019). *An Optimization-Based Approach for Continuous Map Generalization*. PhD thesis, University of Würzburg.

Peng, D., Meijers, M., and van Oosterom, P. (2020). Paralleling generalization operations to support-smooth zooming: Case study of merging areaobjects.

Peng, D., Wolff, A., and Haunert, J.-H. (2017). Using the a* algorithm to find optimal sequences for area aggregation. In *Advances in Cartography and GIScience*, pages 389–404. Springer International Publishing.

Sester, M. and Brenner, C. (2005). Continuous generalization for visualization on small mobile devices. In *Developments in Spatial Data Handling*, pages 355–368, Berlin, Heidelberg. Springer Berlin Heidelberg.

Suba, R. (2017). *Design and development of a system for vario-scale maps*. PhD thesis, A+BE — Architecture and the Built Environment.

Šuba, R., Meijers, M., Huang, L., and van Oosterom, P. (2014). An area merge operation for smooth zooming. In *Connecting a Digital Europe Through Location and Place*, pages 275–293. Springer International Publishing.

Šuba, R., Meijers, M., and van Oosterom, P. (2013). 2d vario-scale representations based on real 3D structure. In *Proc. 16th ICA Workshop on Generalisation and Multiple Representation (ICAGM)*.

Thiemann, F., Warneke, H., Sester, M., and Lipeck, U. (2011). A scalable approach for generalization of land cover data. In Geertman, S., Reinhardt, W., and Toppen, F., editors, *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography, chapter 10, pages 399–420. Springer.

van Kreveld, M. (2001). Smooth generlization for continous zooming. In *20th International Cartographic Conference*.

van Oosterom, P. (1993). The gap-tree, an approach to "on-the-fly" map generalization of an area partitioning. In *GISDATA Specialist Meeting on Generalization*.

van Oosterom, P. (2005). 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.

van Oosterom, P. (2012). Challenge the future delft university of technology. In *STW User Committee meeting,*. Oracle.

van Oosterom, P. and Meijers, M. (2011). Towards a true vario-scale structure supporting smooth-zoom. In *Proceedings of the 14th Workshop of the ICA Commission on Generalisation and Multiple Representation & the ISPRS Commission II/2 Working Group on Multiscale Representation of Spatial Data, Paris*, page 19.

van Oosterom, P. and Meijers, M. (2013). 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.

van Oosterom, P., Meijers, M., Stoter, J., and Šuba, R. (2014). Data structures for continuous generalisation: tGAP and SSC. In Burghardt, D., Duchêne, C., and Mackaness, W., editors, *Abstracting Geographic Information in a Data Rich World: Methodologies and Applications of Map Generalisation*, Lecture Notes in Geoinformation and Cartography, chapter 4, pages 83–117. Springer, Cham.

van Oosterom, P. and Schenkelaars, V. (1995). The development of an interactive multi-scale GIS. *International journal of geographical information systems*, 9(5):489–507.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.