Master's Thesis - GEO 511

Road Network Selection for Small-Scale Maps Using an Improved Centrality Approach

Roy Weiss

Matrikelnr: 06-914-196

Email: roy.weiss@geo.uzh.ch

September 27, 2013

Faculty Representative: Prof. Dr. Robert Weibel
Advisor: Prof. Dr. Robert Weibel

University of Zurich UZH

GIScience: Geographic Information Systems
Department of Geography
University of Zurich

Acknowledgments

I would like to thank my advisor, Prof. Dr. Robert Weibel, for his valuable input and continued support throughout the work on my thesis. In addition, thanks goes to the experts at swisstopo, in particular Cédric Métreaux, who was always available when needed and supported me during my work, but also to Adrian Böhlen, for the help during the evaluation phase.

I owe special thanks to my parents and my significant other, Christie, who showed their support not only during the work on this thesis, but also in every other part of my life.

Thanks is also due to Stefan A. Benz, who worked on a related thesis. Together, we were able to develop the roundabout detection algorithm. In addition, many valuable inputs resulted from the hour-long, insightful discussions regarding the programming part of this work. The same goes to Oliver Zihler, who was always available for language- and programming-related questions.

Abstract

The Federal Office of Topography of Switzerland (swisstopo) has the vision to ultimately derive their complete series of map products from a single, high resolution topographic landscape model, namely, the TLM3D. With a nominal scale between 1:5,000 and 1:25,000, depending on the area and feature class, TLM3D also serves as a basis to derive small-scale maps of 1:200,000 and smaller. Due to the vast scale difference, the generalization process is currently still carried out manually in an interactive system.

The greater objective of this work is to develop a methodology to automate this process. The stroke-based, graph-theoretic centrality approach by Jiang and Claramunt (2004a) seemed promising for road network selection in small-scale maps, as it incorporates the importance of roads. In the first part of this thesis, this approach was implemented in Java and evaluated on four different test areas based on requirements worked out in collaboration with swisstopo. While the approach successfully extracted many of the most important roads, several problems could be identified which hinder the approach from fulfilling all requirements satisfactorily.

In the second and core part of this Master's thesis, several enhancements to the basic algorithm were developed and evaluated in order to rectify these problems. The first enhancement, a roundabout collapse algorithm, is able to successfully detect roundabouts in the test regions. A subsequent collapse operation results in a better continuity of the strokes. The second enhancement uses an extended set of rules for the concatenation of strokes. Next to an adaptive angle threshold, the algorithm ensures that only segments with a similar road class are concatenated. As a third improvement, an adaptive centrality threshold, based on road density is introduced. It successfully corrects part of the pre-selected road network, which remained in a too dense state, leading to a more balanced selection result. Because the creation of new dead-ends and disconnected parts is inherent to all stroke-based selection algorithms, the last improvement consists of the intelligent reconnection of dead-ends and disconnected parts.

The results of the improved approach were quantitatively and qualitatively evaluated on four test areas. The qualitative evaluation was conducted by swisstopo cartographers. The results indicate that the improved approach is able to rectify many of the problems inherent to the basic approach, leading to a significantly better fulfillment of the requirements.

Zusammenfassung

Das Bundesamt für Landestopografie der Schweiz (swisstopo) hat die Vision, ihre Serie von Landeskarten von einem einzelnen, hoch aufgelösten topografischen Landschaftsmodell (TLM3D) abzuleiten. TLM3D, welches einen Massstab zwischen 1:5,000 und 1:25,000 aufweist, dient auch als Basis für Karten im Massstab von 1:200,000 und kleiner. Da der Sprung auf ein kleinmassstäbliches Modell sehr gross ist und die Generalisierung einen sehr komplexerten Prozess darstellt, wird der Generalisierungsprozess bei swisstopo derzeit noch vorwiegend manuell an einem interaktiven System durchgeführt.

Das Ziel dieser Arbeit ist die Entwicklung eines Algorithmus um diesen Prozess zu automatisieren. Der auf Graphen und Strokes basierende Zentralitätsalgorithmus von Jiang and Claramunt (2004a) erschien nach eingehender Literaturrecherche als sehr vielversprechend für die automatische Selektion von Strassennetzwerken für kleine Massstäbe, da er die Wichtigkeit von Strassen berücksichtigt. Im ersten Teil dieser Arbeit wurde dieser Ansatz deshalb in Java implementiert und anschliessend anhand von vier verschiedenen Testregionen, basierend auf Anforderungen welche in Zusammenarbeit mit swisstopo erarbeitet wurden, eingehend evaluiert. Währenddem der implementierte Basisalgorithmus viele der wichtigsten Strassen aus dem Ursprungsdatensatz extrahieren konnte, wurden im Zuge der Evaluierung eine Reihe von Problemen identifiziert, welche den Algorithmus daran hindern, die Anforderungen zufriedenstellend zu erfüllen.

Im zweiten Teil, welcher den Kern dieser Arbeit darstellt, wurden deshalb eine Reihe von Erweiterungen eingeführt mit dem Ziel, diese Probleme zu lösen. Die erste Erweiterung, ein Kreiseldetektions und -eliminations-Algorithmus, detektiert und eliminiert erfolgreich Kreisel in den verwendeten Testgebieten. Die zweite Erweiterung verwendet einen erweiterten Satz an Regeln für den Strokeverbindungsalgorithmus. Neben einem adaptiven Winkelgrenzwert, garantiert der Algorithmus, dass nur Strassensegmente, welche eine ähnliche Klasse aufweisen, miteinander verbunden werden. Die dritte Erweiterung, ein adaptiver Zentralitätsgrenzwert basierend auf der Strassendichte, korrigiert Teile des vorselektierten Strassennetzwerks, indem dichte Regionen in einem zweiten Durchlauf weiter ausgedünnt werden. Dadurch kann eine bessere Struktur des Strassennetzwerks erzielt werden. Da die Erstellung neuer Sackgassen und vom Hauptnetzwerk getrennten Segmenten ein Problem aller Stroke-basierter Selektionsalgorithmen ist, wurde als vierte Erweiterung ein Algorithmus entwickelt, welcher diese Elemente in geschickter Weise mit dem Hauptnetzwerk verbindet.

Die Resultate des erweiterten Ansatzes wurden abschliessend quantitativ und qualitativ ausgewertet. Die qualitative Evaluation erfolgte dabei durch Experten von swisstopo. Die Resultate der Evaluierung zeigen auf, dass der erweiterte Ansatz die meisten Probleme des Basisansatzes erfolgreich beheben kann, wodurch die Anforderungen an das Endresultat deutlich besser erfüllt werden können.

Contents

List of Figures

List of Tables

List of Abbreviations

BFS	Breadth-First Search
CRITIC	Criteria Importance Through Intercriteria Correlation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCM	Digital Cartographic Model
DFS	Depth-First-Search
IDE	Integrated Development Environment
MST	Minimum Spanning Tree
NMA	National Mapping Agency
RL	Radical Law
RQ	Research Question
SOM	Self-Organizing Map
TLM3D	Topographic Landscape Model 3D
UML	Unified Modeling Language

1 Introduction

1.1 Context and Motivation

One of the ultimate goals of many National Mapping Agencies (NMAs) is to derive smaller-scale data sets from a single, detailed database (Beard, 1987; Foerster et al., 2010). Thus, the Federal Office of Topography of Switzerland (swisstopo) has the vision to ultimately derive their complete series of map products from a single, high resolution model, the TLM3D. TLM3D is the large-scale topographical landscape model, which includes natural and artificial features and is the most extensive and accurate 3D vector data set of Switzerland (swisstopo, 2012b). With a nominal scale between 1:5,000 and 1:25,000 (depending on the area and feature class; swisstopo, 2012b), TLM3D serves as the basis to derive smaller scale data sets, such as the VECTOR200 product, which features a scale of 1:200,000 (swisstopo, 2012d). Owing to the vast scale difference and the complexity of map generalization involved, the derivation process is currently still carried out completely manually in an interactive system.

Figure 1.1 shows the planned derivation and map production process at swisstopo. It shows the two landscape models, TLM3D and VECTOR200 in green and the digital cartographic models (DCMs), which serve as the basis for the production of the actual national map, in yellow. The TLM3D data set is updated constantly, with an updated product being released every six years (swisstopo, 2012b). The VECTOR200 data set is updated and released annually: because an annual complete generalization from the TLM3D to the VECTOR200 data set would require too much work, the data set is simply updated with the changes made to the TLM data set (swisstopo, 2012d,e). Therefore, in order to render the map generalization process more repeatable and efficient, swisstopo is interested in automating the selection of certain feature classes, in particular the road network. Because road networks provide important contextual information, they can be considered as essential components of maps (Touya, 2010). While automated generalization and selection of road networks has undergone much research in the last decade (Kreveld and Peschier, 2000; Jiang and Claramunt, 2004a), it is still considered as a very complex process due to the fact that the linear road segments are connected to each other in a coherent network (Chaudhry and Mackaness, 2005). In addition, they have a certain shape, angle, orientation and length instead of just a location and a value of importance, which makes the generalization of roads a very complex process (Chaudhry and Mackaness, 2005). Map generalization is a very broad, multifaceted concept (see Section 2.1) and the complete generalization of a road network incorporates many different operators. This thesis focuses on the selection operator, that is, choosing the relevant information in relation to the target map or database specifications, which can be considered as a key step in the generalization process (Touya, 2010).

While much research has been done regarding the automatic selection of road networks in the recent past (see Section 2.2), the problem has not yet been solved satisfactorily. As has been mentioned before, the manual or interactive generalization, in particular of road networks, is a very expensive and

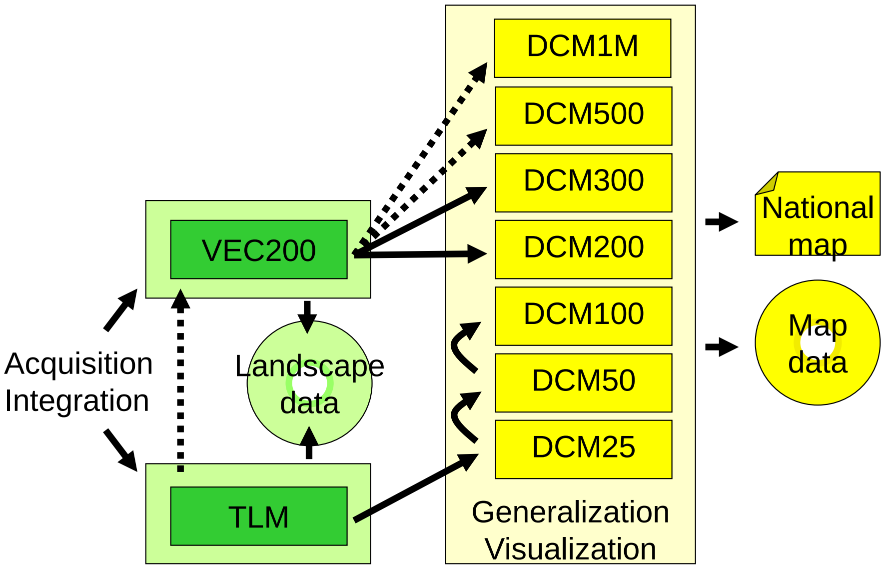


Figure 1.1: The acquisition and generalization process at swisstopo, which has already been partly implemented. The picture shows the two main landscape models, TLM3D and VECTOR200 in green. TLM3D is the accurate, large-scale landscape model, with a scale of 1:5,000 to 1:25,000. VECTOR200 on the other hand, is the small-scale landscape model with a scale of 1:200,000. The landscape models serve as a basis for the digital cartographic models (DCMs), from which the national map is produced (based on swisstopo, 2012a).

time consuming task. However, it also produces data sets and maps of very high quality. Therefore, the main question that arises is whether there exists an approach to the automatic selection of road networks, which is able to generalize the large-scale road network of TLM3D to the target scale of 1:200,000 with sufficient quality, such that costs and working hours could be saved.

1.2 Objectives

The overarching objective of this thesis is to find an algorithm for the automatic selection of the road network of the TLM3D data set for the target scale of 1:200,000. Because the transition from 1:5,000 – 1:25,000 to 1:200,000 is rather large (see Figure 1.2), the first part of this thesis consists in testing whether there exists an algorithm which is able to cope with the vast scale difference and the associated necessary large areas.

The first objective of this thesis is to implement and quantitatively and qualitatively evaluate the stroke-based centrality approach presented in Jiang and Claramunt (2004a). As a basis for the evaluation, constraints and requirements, which have been worked out together with swisstopo, shall be used. The solution should adhere as closely to these constraints as possible.

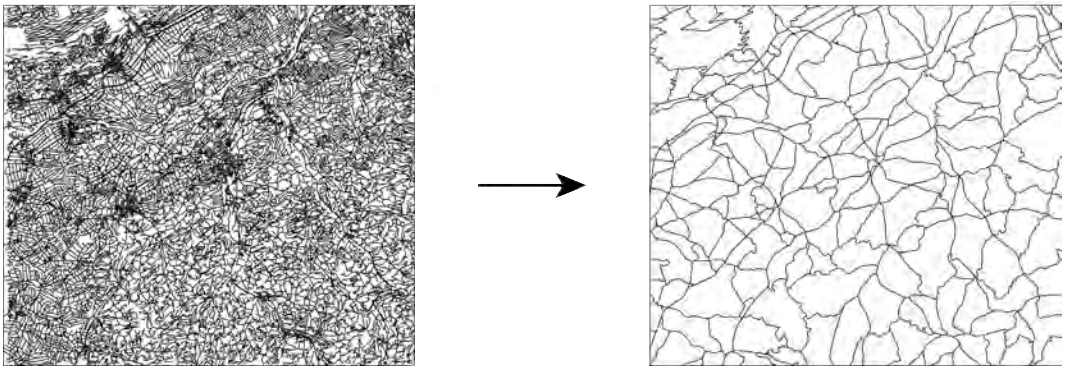


Figure 1.2: The main objective of this thesis is to find an approach for the automatic selection of road networks for the target scale of 1:200,000. To the left, a snippet of the base data set (TLM3D) with a scale of 1:5,000 – 1:25,000 is shown. To the right, one can see the same extract of the VECTOR200 data set, the official 1:200,000 landscape model of swisstopo. Because of the large scale difference, the algorithm needs to be able to cope with a large road network and large areas (in this case 26 km x 23 km). (© swisstopo)

The second and main objective of this thesis is to enhance the basic algorithm in such a way, in order that it better meets the requirements and constraints. The final solution shall then be evaluated quantitatively and qualitatively. The qualitative evaluation of the results should include expert opinions from swisstopo cartographers.

The research questions of this thesis are presented in Section 2.3, following a thorough literature review in Section 2.2.

1.3 Thesis Structure

After the introduction, the necessary background, related work as well as the research questions are introduced in Chapter 2. Afterwards, the data used as well as the different test areas are introduced in Chapter 3. This chapter also contains the constraints and requirements to which the target solution should adhere. In Chapter 4, the methodology of the basic approach is introduced and its results evaluated. Chapter 5 reports on the various improvements and their influence on the road network. Subsequently, the results and a thorough discussion of the improved approach are presented in Chapter 6. Finally, concluding remarks are made in Chapter 7, by reviewing the achievements and insights this thesis offers and giving an outlook for future work.

2 Background and Related Work

2.1 Generalization

Generalization is a fuzzy concept and not well defined (Brassel and Weibel, 1988). There exist several conceptual models which explain the overall process of generalization (e.g. Brassel and Weibel (1988), McMaster and Shea (1992) or Spiess et al. (2002)). Generalization in a very broad sense can be seen in direct relation to the term *abstraction*, as it emphasizes the removal of the unimportant in order to concentrate on the important (which itself is always dependent on the purpose of a map; Brassel and Weibel, 1988). An important aspect of generalization is also the preservation of the structural characteristics of the original road network (Zhang, 2004). Thus, dependent on purpose and scale, the important information is visualized and the unimportant ignored or omitted (Brassel and Weibel, 1988). According to Spiess et al. (2002), generalization mainly consists of the adequate selection and aggregation of objects and the positional accurate, representative and distinct graphical representation. Figure 2.1 shows nine crucial objectives which come into effect during the generalization process. When the goal is to derive a new database and not a map, as it is the case in this thesis, it is called model generalization and is not constrained by cartographic symbols (Weibel and Dutton, 1998; Touya, 2010). Generalization of a map is usually achieved by applying different types of generalization operators, such as selection, smoothing, displacement or merging (Yaolin et al., 2001; Foerster et al., 2007), which relate to the generalization objectives shown in Figure 2.1. As the title of this thesis suggests, only the selection of objects (in this case roads) is considered. The selection operator selects individual objects which should appear in the target dataset (Foerster et al., 2007) and will be discussed in more detail in the next section. To be more precise, also the reclassification and collapse operators are used in the improved approach discussed in Chapter 5.

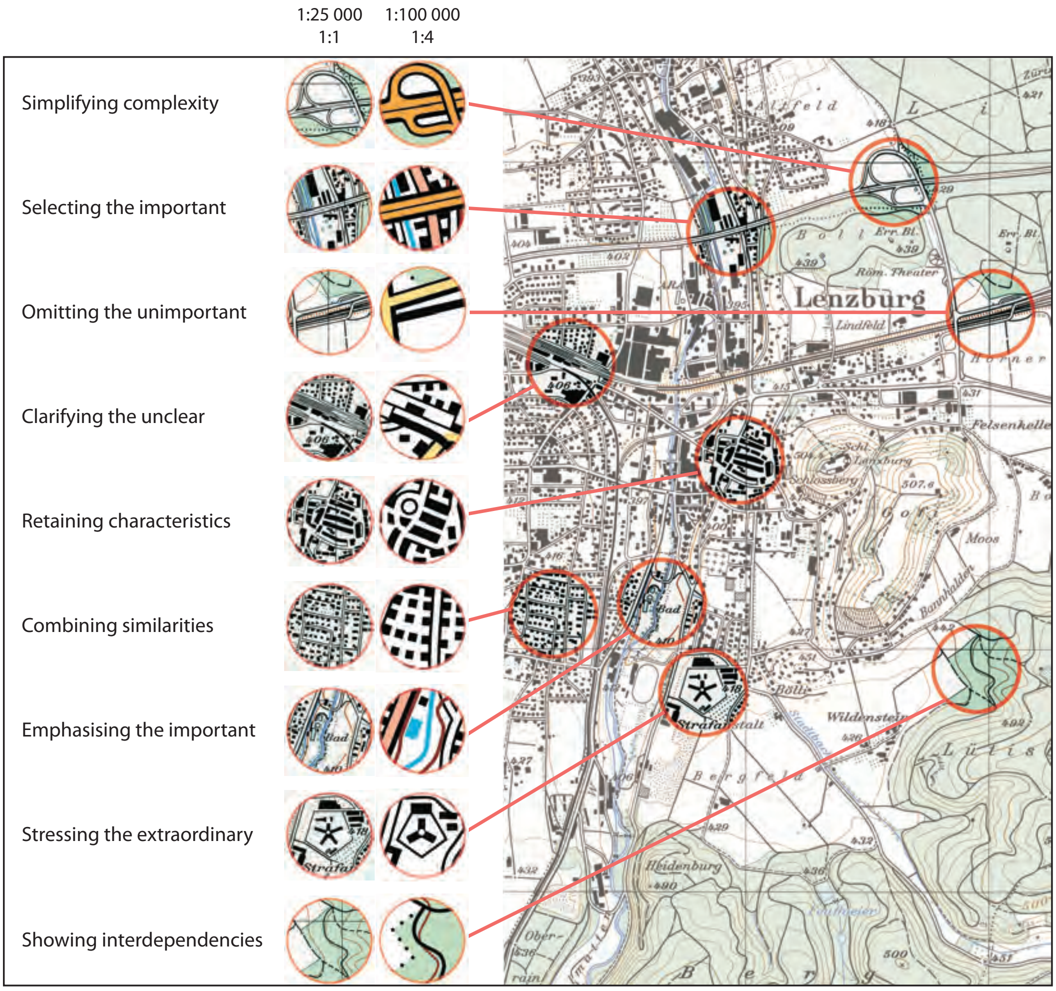


Figure 2.1: Nine crucial objectives of map generalization (based on Spiess et al., 2002).

2.2 Selection: Related Work

Selection can be considered as an operator concerning the abstraction of the spatial database and is a key step to model generalization (Touya, 2010). It seeks to choose the essential and relevant elements of a geographic database but also to maintain the main characteristics of geographic information while reducing the level of detail (Touya, 2010). Because roads are very important features of maps, road network selection has been a key topic of generalization for several years. Liu et al. (2010) categorizes the various algorithms that have been proposed into three major groups: (1) **semantics-based selection**, (2) **graph-based selection** and (3) **stroke-based selection**. In the next part of this section, a short literature review of approaches belonging to (at least) one of the listed groups is presented.

Semantics-based selection. Semantics-based selection is the simplest of the three group of methods. It is primarily based on semantic attributes, such as street types and number of lanes. Streets are selected in a ranked order according to their relative importance of attributes (Liu et al., 2010). Such methods, however, do not produce good results for most scales, as geometrical relationships are largely ignored. For extremely small scales or very small scale differences, where either all roads of a specific type are retained or where the selection is based on very specific details for a special purpose map, such a method may yield acceptable results. However, such methods are not sufficient for a general selection algorithm for a general purpose map.

Graph-based selection. Graph-based methods manipulate road networks as connected graphs (see Section 4.3.1). As Mackaness and Beard (1993) point out, map generalization is known to be complex, and its application to each map requires understanding and modeling at the geometric, topological, and attribute levels. They also introduce the concept of graph theory, which enables one to characterize topological relationships among objects. In addition, they used a so called minimum spanning tree (MST) to select the most important streets in a network while maintaining connectivity between cities, which is a major advantage over a simple semantics-based approach. If, for example, the only connection between two important cities is a road of a lower type, this road is still included in the result as it is the only way to maintain the connectivity.

Thomson and Richardson (1995) use a graph-based approach that employs shortest path algorithms between nodes in the network that provides an initial solution to the problem of deriving measures of the functional relevance of road segments, given a context defined in terms of a set of points of interest. Hence, for a given context, a set of rankings reflecting the importance of the segments is created, which can be used in map density reduction and generalization (Thomson and Richardson, 1995). In addition, spanning trees can be used to maintain connectivity.

While tackling only one specific problem, Mackaness and Mackechnie (1999) introduced an approach based on graph theory and spatial clustering to simplify road junctions. However, it is only useful for specific, relatively simple junctions. An isolation of such junctions is not done automatically, which is why the algorithm cannot be used for an entire region without additional modifications.

Jiang and Claramunt (2004a,b) introduced a graph-theoretic approach, which uses centrality measures (see Section 4.3.4) to determine the importance of roads. A prerequisite for this algorithm is that each segment in the database has a name attached to it, as Jiang and Claramunt (2004a) first concatenate segments with the same name together, forming complete roads in the process. This differs from pure graph-centric approaches in that the algorithm does not work on separate segments, but on roads as a whole, which is why this approach also contains elements of stroke-based selection (as strokes are, essentially, a replacement for roads). The algorithm calculates for each road several centrality measures and performs a simple threshold-based selection afterward: roads, which exhibit a centrality value above the threshold are retained, the ones which feature a value below the threshold are omitted. The overall density of the road network can be varied by simply decreasing or increasing the threshold. Jiang and Claramunt (2004a) note, however, that the approach also exhibits several shortcomings and that the centrality measures should be considered with other geometric and semantic properties. For example, the algorithm does not guarantee that the road network remains connected after the selection. Nevertheless, the results show that the general structure of the road network can be retained, which is an important aspect of generalization. Centrality measures were also used in conjunction with Self-Organizing Maps (SOMs) (Jiang and Harrie, 2004). Using this approach, it was possible to include additional parameters, such as road length, the number of lanes and the speed limit as selection criteria.

More recently, Yang et al. (2011) improved upon the basic centrality approach of Jiang and Claramunt (2004a) by eliminating several of its shortcomings. The presented approach can be seen as a combination of graph-based and stroke-based selection, as instead of street names, they used the classic stroke approach of Thomson and Richardson (1999). In addition, they introduced methods to detect dual carriageways and complex street junctions as well as ways to simplify such structures. As a result, the continuity of the strokes could be maintained more successfully across the road network. Similar to the reasoning behind Jiang and Harrie (2004), who used SOM to use several criteria simultaneously to assess the importance of roads, they used the *Criteria Importance Through Intercriteria Correlation* (CRITIC) method of Diakoulaki et al. (1995) to use a combination of several centrality measures as well as the stroke length to determine the importance of a stroke. To eliminate the aforementioned problem of the centrality approach that it does not prevent the road network from becoming disconnected, a MST was used which ensured a connected network.

Stroke-based selection. Strokes were introduced by Thomson and Brooks (2000) and will be discussed in more detail in Section 4.3.2. The stroke approach is based on the perceptual grouping principle of good continuation (see Figure 2.2), which means that at each intersection in the road network, the segments are grouped together such that each pair of segments forms the smallest angle possible (Thomson and Richardson, 1999). This results in potentially long, linear elements which extend through junctions Thomson (2006). These elements were termed *strokes*, prompted by the idea of a curvilinear segment that can be drawn in one smooth movement and without a dramatic change in line style (Thomson and Brooks, 2000; Thomson, 2006). The generated strokes can be sorted afterward according to some attributes, like length or road type (Thomson, 2006). A selection can then be performed by simply selecting only the longest few strokes, for example.

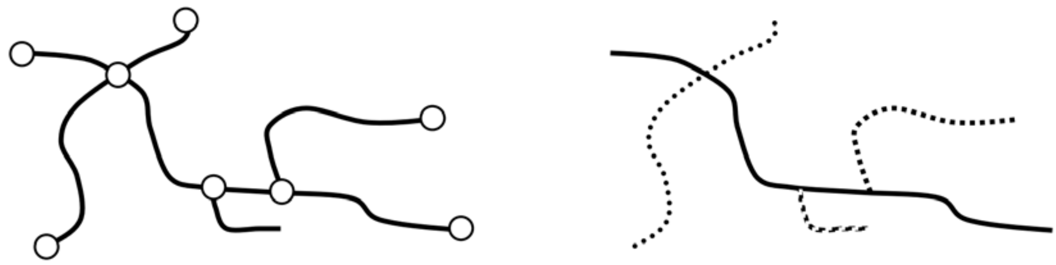
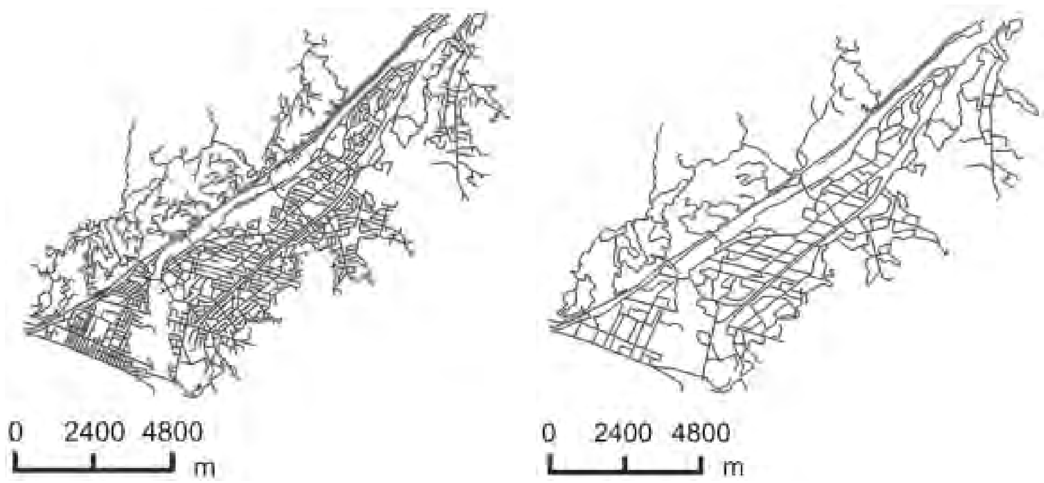


Figure 2.2: Resulting strokes (right) of a road network (left). This technique can be used to concatenate several individual segments to form a road (based on Thomson and Brooks, 2002).

Next to the initial work of Thomson and Richardson (1999), Thomson and Brooks (2000) and Thomson and Brooks (2002), who have shown promising first results using this approach for the generalization of road and hydrological networks, more researchers have focused on this approach in the recent past. Chaudhry and Mackaness (2005) presented a framework for the generalization of both rural and urban roads over a large scale change using a stroke-based approach. They have shown that simple attributes, such as length, can be used to determine the order of importance on which the process of generalization can be based (Chaudhry and Mackaness, 2005). Liu et al. (2010) used the stroke-approach in combination with Voronoi partitioning, which enables the authors to incorporate road density into the algorithm. The algorithm produced reasonable results with a potential to be adopted in road selection for map generalization (Liu et al., 2010). Touya (2010) improved upon the basic stroke algorithm using several additional algorithms and enhancements. First of all, he used various road structure detection algorithms to isolate crossroads, complex junctions and dual carriageways. Furthermore, additional information such as points of interests were used to further enrich the dataset with information. Note that most of the *stroke-based* algorithms also belong to the category of *graph-based* algorithms. Similarly, *graph-based* algorithms may also belong to the group of *stroke-based* algorithms.

Other approaches. While the classification of road selection algorithms into the three aforementioned classes used by Liu et al. (2010) is well suited to group the approaches of the last decade, it is by no means exhaustive or the only possible way to group the different types of approaches. Another notable group are the so-called **mesh-based** algorithms. The crucial element in these algorithms are the areas which are formed between roads. The areas in between roads (i.e. so-called meshes) represent the overall density of an area more accurately than simple line-based density (Chen et al., 2009). The approach presented by Chen et al. (2009) does not use the stroke-based selection explicitly, but incorporates the principle of strokes in the elimination phase of the algorithm. The basic principle is to dissolve the dense meshes by eliminating one boundary segment (Hu et al., 2007; Chen et al., 2009). As a result, the most dense meshes become progressively larger and the road network less dense. Li and Zhou (2012) and Zhou and Li (2012) improved upon the approach of Chen et al. (2009) by using the mesh-based algorithm in conjunction with a stroke-based algorithm to handle areal and linear features separately, which eliminates one shortcoming of purely mesh-based approaches. An example of a result of the integrated stroke- and mesh-based approach can be seen in Figure 2.3.



(a) (b)

Figure 2.3: Example of the combined stroke- and mesh-based approach with the original road network on the left and the selected one on the right (based on Li and Zhou (2012)).

Another approach, which was developed in a multinational research effort, is the AGENT model (Lamy et al., 1999). Based on so called agents, which can be described as *"a self contained program capable of controlling its own decision making and acting"* (Lamy et al., 1999, pg. 1), the project seeks to integrate a multitude of generalization algorithms and operators which are executed autonomously by the agents. Such an agent-based approach was used by Morisset and Ruas (1997) for road selection.

2.3 Research Gaps and Research Questions

While reviewing the literature, the approach based on Jiang and Claramunt (2004a), which has not been used widely, looked promising for the aims of my thesis for various reasons. First of all, it proved to generate good results in the recent work by Yang et al. (2011), who were able to eliminate several shortcomings of the original algorithms. Secondly, the principle behind the approach seemed to fit the aims of my thesis very well. At a target scale of 1:200'000, which can be considered (especially for Swiss dimensions) as a small scale, the *importance* of a road within the road network becomes the most relevant criterion: the algorithm needs to evaluate whether a road, if the whole road network is considered, is important enough to be retained in the pruned network or not. However, the approach has never been evaluated on large areas. This is also true for some of the other approaches, which have only been evaluated on relatively small road networks. If an approach were to be used for a small scale, it should produce meaningful results not only for small, experimental datasets but also for larger ones. This lead to the following research questions (RQs):

RQ1: Is it possible to generalize the road network of the TLM dataset for a target scale of 1:200,000 in such a way that the requirements of swisstopo (listed in 3.3.2) are fulfilled?

RQ2: If the the basic algorithm does not fully meet the requirements, how can it be improved such that it performs to specifications?

3 Data and Requirements

3.1 Swisstopo TLM3D

Swisstopo has been developing and updating the Topographic Landscape Model TLM3D since spring of 2008 (swisstopo, 2012c). The TLM3D covers the whole of Switzerland as a comprehensive basic landscape model in three dimensions with a very high accuracy. Swisstopo estimates the geometric accuracy for well-defined landscape features, such as roads, at 0.2 - 1.5 m (swisstopo, 2012b). While the first products were released in autumn 2010, the integration of feature classes into the database is still underway. Several different feature classes, such as land cover, hydrography and administrative boundaries or buildings exist in the TLM3D. For the purpose of this work, we are only interested in the roads and tracks. Table 3.1 shows each object type of the road feature class. It becomes evident that some filtering is required beforehand, as we are not interested in some of the object types for the targeted scale. To reduce the complexity, the following object types were removed from the data set before the main algorithms presented in chapter 4 and 5 were executed:

- **Roadhouse.** In small-scale environments, this type is either not displayed at all or represented as a point.
- **Service-only access-road.** We are only interested in roads or paths which are open to public traffic.
- **Plaza.** Same as above.
- **Car-carrying train.** We are only interested in streets and paths.
- **Ferry.** Same as above.
- **Climbing route.** Same as above.

This leaves a total of **15 different object types**. Except for the *connection* object type, the meanings of all of the types should be clear and do not require further explanations. The connection object type is used in the TLM3D dataset where a segment connects itself to another segment asymptotically. This happens very often at highway exits and entries, for example. For several reasons, this is not wanted in the dataset and thus the segments which connect asymptotically are truncated and a segment which connects in a square angle is inserted in the TLM3D database. An example can be seen in Figure 3.1. While this does not change the topology of the road network, it poses some problems to the algorithms, as will be discussed in a later stage in this thesis.

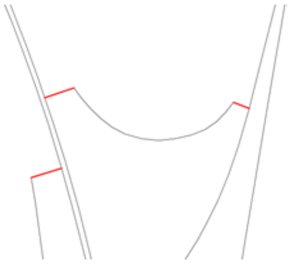


Figure 3.1: The 'connection' object type, shown in red, prevents segments from asymptotically connecting together. (© swisstopo)

Table 3.1: Definition of all classes used in the road feature class of the TLM3D data set. The classes which were removed from the dataset before executing the selection algorithms are listed at the bottom of the table (translated from swisstopo, 2012e).

Object Type	**Width**	**Comments**
Connection	N/A	Connection between two axes which do not intersect. Necessary to guarantee a connected network.
Freeway	> 10m	-
Expressway	> 10m	-
Entry	N/A	Direction separated entry to a freeway or expressway.
Exit	N/A	Direction separated exit from a freeway or expressway.
Access-road	N/A	Road segment between entry/exit and the connection to a main road.
10m street	>10.20 m	-
6m street	6.21 - 10.20 m	-
4m street	4.21 m - 6.20 m	-
3m street	2.81 m - 4.20 m	-
2m path	1.81 m - 2.80 m	-
2m path fragment	1.81 m - 2.80 m	-
1m path	<= 1.80 m	-
1m path fragment	<= 1.80 m	-
Marked track	N/A	Non-visible path, but access possible.
Service-only access-road	N/A	-
Square	N/A	-
Car-carrying train	N/A	-
Roadhouse	N/A	Roadhouse with exclusive access from freeway or expressway.
Ferry	N/A	-
Climbing route	N/A	-

3.2 Test Areas

To test and evaluate the approach, four different test areas in Switzerland were chosen and will be presented shortly in this section. The test areas were chosen such that the characteristics of the road network differ significantly between each of them (see Table 3.2 for a general overview). The algorithm should not only work in urban settings, but also in rural and even mountainous regions. A criterion for every test area was that also towns and cities are contained in them. Hence, there exists no test area which entirely consists of an urban area, nor entirely of a rural area. Even the mountainous area was chosen such that larger towns are included. The four test areas will be presented in more detail and are named according to a prominent city which is enclosed in the region:

- **Davos:** Named after a well-known town, this is the mountainous test area. The area itself is rather large, but the contained road segments are relatively few.
- **Lucerne:** This area is named after the largest city present in this region of central Switzerland. It contains a relatively large urban area, rural regions and also mountainous areas.
- **Langenthal:** The Langenthal test area is again named after the city, which is located in the center of the area. The whole area is part of the Swiss Central Plateau (*"Mittelland"*). The area is speckled with smaller towns but consists mostly of rural areas.
- **Zurich:** Named after the largest city of Switzerland, this region consists mainly of urban areas, speckled with rural regions.

Table 3.2: Key figures of the four test regions.

Test Area	**Size [km^2]**	**#Segments**	**Total length [m]**	**HS**[1]	**HL**[2][m]	**HE**[3]
Davos	1,200	9757	2,752,991	0	0	0
Lucerne	520	26,232	4,015,310	355	122,580.6	46
Langenthal	600	31,318	5,427,221	87	42'796.9	16
Zurich	700	65,712	8,410,505	853	257'112.2	95

3.2.1 Davos

Figure 3.2 shows the snippet of the TLM3D dataset used for the Davos test area. The region around Davos is characterized by hills and mountains. Hence, most of the roads are narrow and many hiking trails exist. The area is around 40 km by 30 km and covers around 1,200 km^2. It consists of 9757 road segments which account for a total length of 2,752,991 m. Prominent features are the wide road, which surrounds several mountains in the middle of the excerpt and larger towns, such as Davos above the center and Arosa in the upper left corner. The Davos test area is the only area which does not feature any highway segments.

[1] Highway Segments
[2] Total Highway Segment Length
[3] Highway Exits and Entries

3.2.2 Lucerne

The second test area, pictured in Figure 3.3, is named after the prominent city on the eastern end. The area is 26 km by 20 km, leading to a total area of roughly 520 km^2. The road density is significantly higher than in the Davos test area, as the area consists of 26,232 road segments with a total length of 4,015,310 m. Also visible are the foot paths to the south, which surround the *Mount Pilatus*, a mountain just south to the test area. The interesting part about this region is the co-occurrence of major roads and very narrow footpaths. While the footpaths are generally less important in urban and even rural regions, they cannot be ignored here as a significant portion of the network would be lost. In contrast to the Davos area, the highways are very prominent in this area with 355 highway segments featuring a length of over 122 km.

3.2.3 Langenthal

The area around Langenthal can be characterized as a relatively homogeneous region, having a coarse network of major roads with many minor roads in between which grant access to several small towns, farms and hamlets. As can be seen in Figure 3.4, there exist several larger towns with Langenthal in the center. The road density is slightly higher than in the Lucerne test area. On the other hand, the highways are much less prominent. With dimensions of around 26 km by of 23 km, resulting in a total area of 600 km^2, the size of the area slightly larger than the Lucerne area. The extract contains 31,318 road segments, covering 5,427,221 m of length. A total of 87 highway segments with a length of nearly 43 km and 16 highway entries and exits exist in this snippet.

3.2.4 Zurich

The Zurich test area is roughly 35 km by 20 km, which leads to a total area of 700 km^2. While only slightly larger than the Langenthal and Lucerne regions, the road density is much higher, as is clearly evident in Figure 3.5. The snippet contains 65,712 road segments with a total length of 8,410,505 m. The whole region has a predominantly urban character, with the city of Zurich close to the center, right above the *Lake Zurich*. To the southwest and north, more rural regions can be found, however. Another prominent feature is the airport of Zurich-Kloten, and several lakes, which leave wide gaps in the road network. Because it is a very large urban region and a very important road network node in Switzerland, the Zurich test area has a total of 853 highway segments with a total length of nearly 260 km. In addition, 95 exits and entries grant access to the highways.

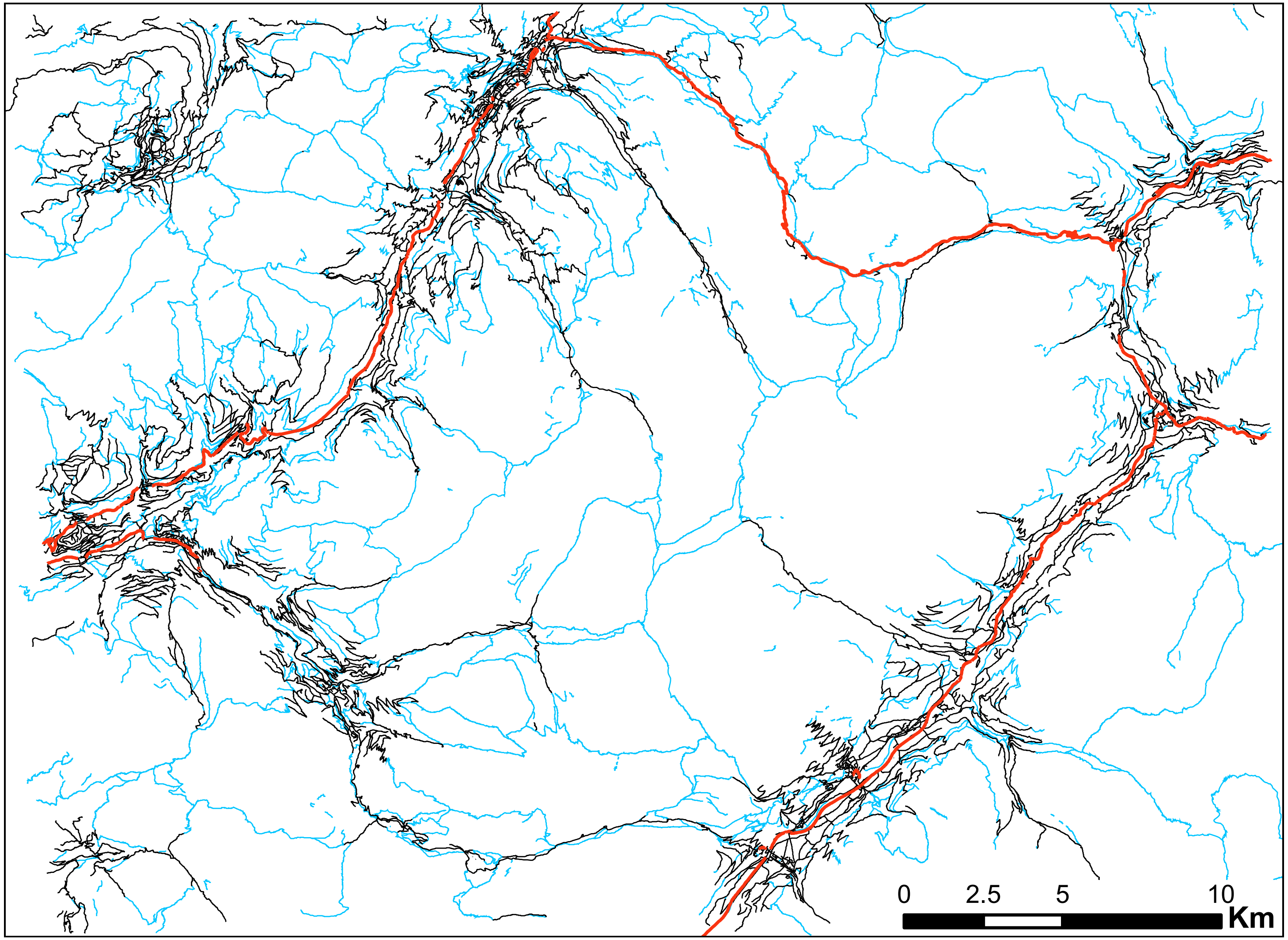


Figure 3.2: Road network of the TLM3D dataset in the Davos test area. Wide roads (6 m to 10 m) are colored in red, roads having a width between 2 m and 6 m are colored in black. Paths and trails up to 2 m are colored in light blue. Road types which are not included in the final results, such as car-carrying trains, have been removed. (© swisstopo)

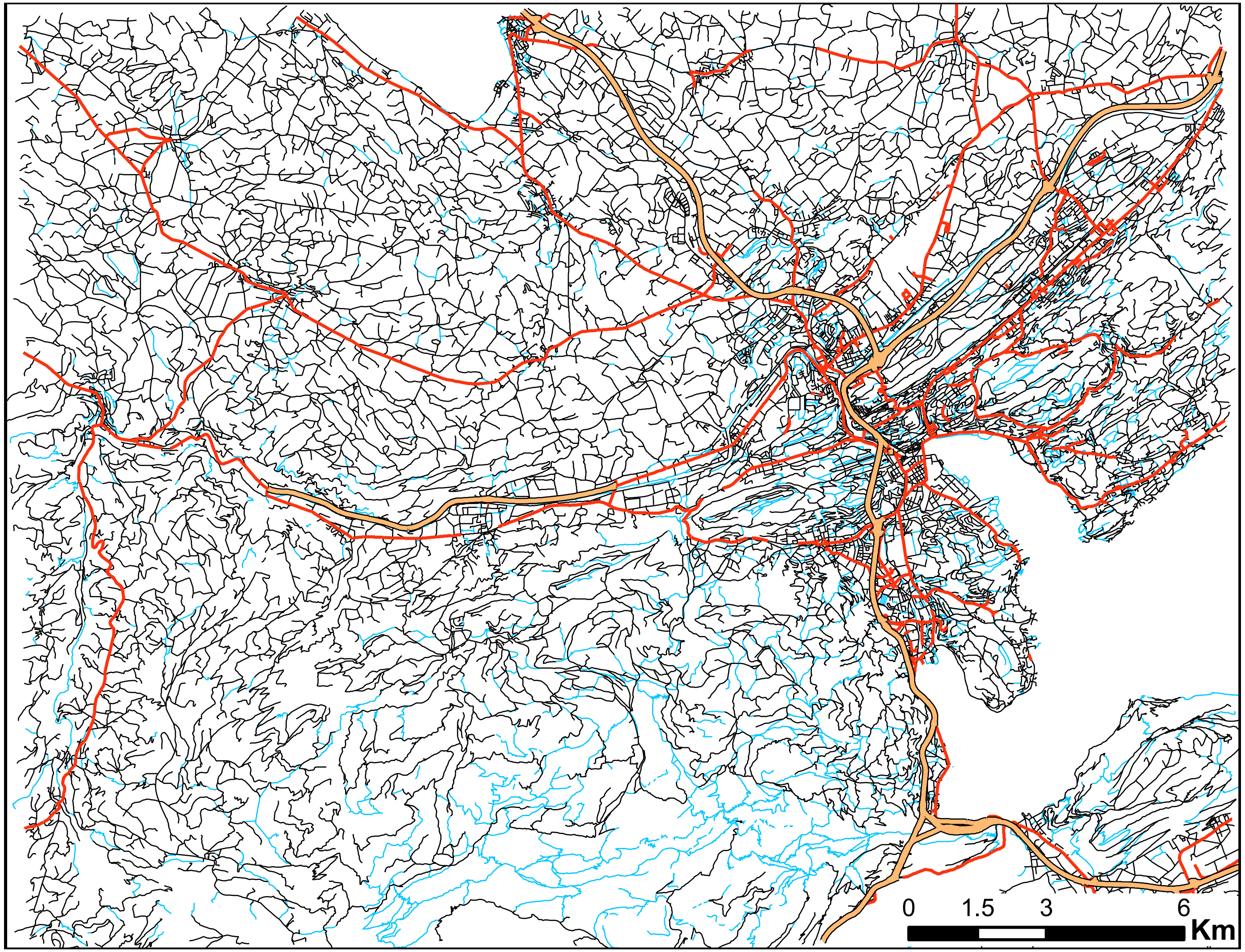


Figure 3.3: Road network of the TLM3D dataset in the Lucerne test area. Wide roads (6 m to 10 m) are colored in red, roads having a width between 2 m and 6 m are colored in black. Paths and trails up to 2 m are colored in light blue. Road types which are not included in the final results, such as car-carrying trains, have been removed. (© swisstopo)

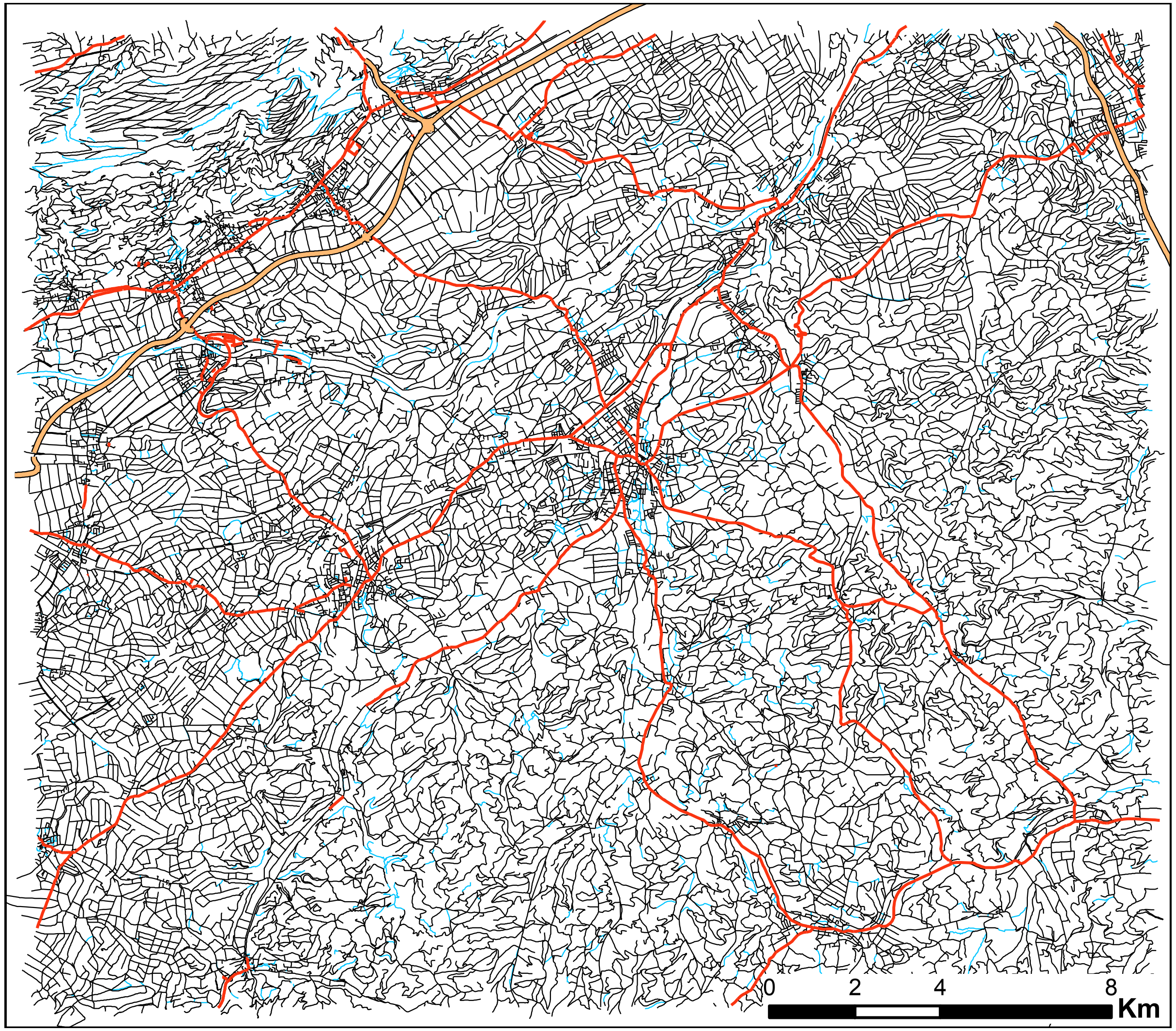


Figure 3.4: Road network of the TLM3D dataset in the Langenthal test area. Wide roads (6 m to 10 m) are colored in red, roads having a width between 2 m and 6 m are colored in black. Paths and trails up to 2 m are colored in light blue. Road types which are not included in the final results, such as car-carrying trains, have been removed. (© swisstopo)

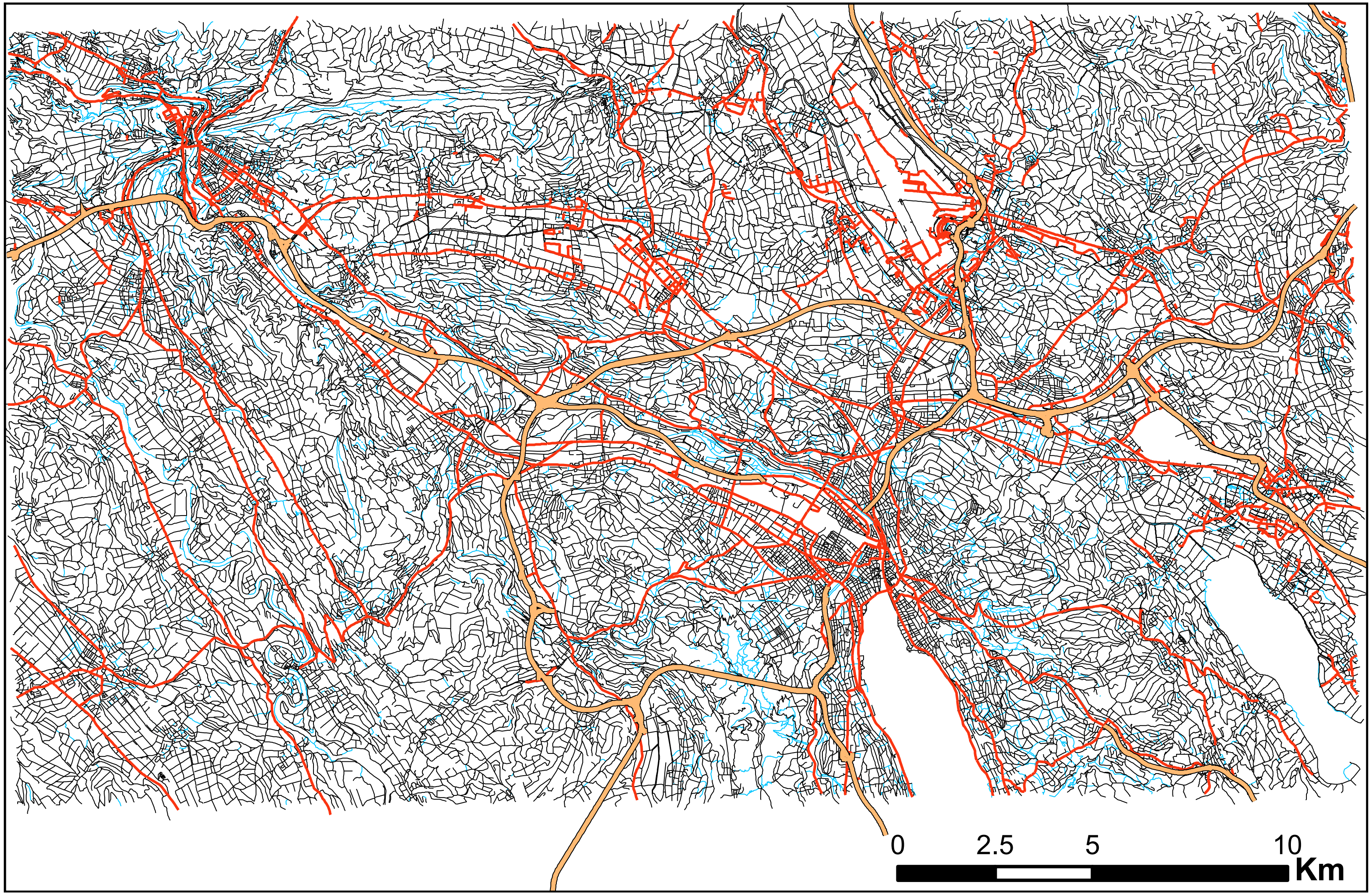


Figure 3.5: Road network of the TLM3D dataset in the Zurich test area. Wide roads (6 m to 10 m) are colored in red, roads having a width between 2 m and 6 m are colored in black. Paths and trails up to 2 m are colored in light blue. Road types which are not included in the final results, such as car-carrying trains, have been removed. (© swisstopo)

3.3 Constraints and Requirements

3.3.1 Constraint-Based Generalization

The idea behind constraint-based generalization is that the requirements to the end product (in this case the pruned road network for a target scale of 1:200,000) act as constraints under which the algorithm may operate (Beard, 1991; Harrie and Weibel, 2007). Constraints can also be understood as *"design specifications to which solutions should adhere"* (Weibel and Dutton, 1998, pg. 2). Constraints do not define *how* a solution needs to be reached, they simply define aspects which may not be violated in the generalized solution - the less constraints a solution violates, the better the algorithm which produced it (Harrie and Weibel, 2007). Constraints may be categorized into different typologies, such as the one discussed in Weibel and Dutton (1998) or Harrie (2003). For the purpose of this work, the one discussed in Harrie and Weibel (2007) is used. Hence, the constraints can be assigned to one of the following categories:

- **Position constraints.** Position constraints limit the movement of objects (in absolute and relative terms).
- **Topology constraints.** Topology constraints ensure that the relationship between objects are maintained.
- **Shape constraints.** These constraints are used to ensure that the shape characteristics of individual objects are preserved.
- **Structural constraints.** The generalized map should preserve the object patterns of the original source map.
- **Functional constraints.** Functional constraints are used to ensure that the generalized map can be used for a certain purpose.
- **Legibility constraints.** These constraints are used to limit or exclude spatial conflicts which limit the legibility of the generalized map.

Because only a road network *selection* is performed in this work, some constraints, such as position constraints, shape constraints and legibility constraints are not of importance, as the algorithm cannot ensure that it does not violate such types of constraints. Topology (**T**), structural (**S**) and functional (**F**) constraints on the other hand are of high importance and the next section lists constraints which fall under one of these categories.

3.3.2 Constraints of the Target Road Network

In addition to some constraints formulated in Spiess et al. (2002), discussions with swisstopo revealed several additional constraints that are important for a road network at the target scale of 1:200,000. They were classified on the one hand as **hard constraints**, which are usually either fulfilled or not fulfilled. They can also be tested relatively easily. For example, the statement *"No footpath may exist in the final result"* is either fulfilled or not fulfilled. On the other hand, there exist **soft constraints**, which cannot be tested as easily and where a wide spectrum of different solutions may be possible. An example may be the statement *"The general structure of the road network should be preserved."*, which does not find a simple binary answer.

In the following, all hard and soft constraints used in this thesis will be discussed. The constraints are also assigned to one of the different constraint categories which were presented before.

Hard constraints:

1. **[F]** All highways and expressways and everything related to them (such as entries and exits) must be included in the generalized map.
2. **[F]** The entries and exits to and from the highways must be directly connected to the road network.
3. **[T]** The road network must be completely connected. That is, from every point in the network, one has to be able to reach every other point in the network.
4. **[T]** No dead-ends may exist in the final result.

Soft constraints:

5. **[S | F]** The overall thinning of the network needs to be appropriate for the target scale of 1:200,000, comparable to VECTOR200.
6. **[S]** The general structure of the road network needs to be maintained.
7. **[S]** Larger towns and cities should still be visible in the road network of the final map.
8. **[F]** Important link roads need to be maintained in the pruned road network.

The inclusion of all highways and expressways is a requirement set by swisstopo and should be fulfilled under all circumstances, as the highways represent the most important roads in such a small scale map. In addition, one should obviously be able to reach the entries and exits of the highways, which is why an additional constraint was added that ensures the accessibility of the highways. In addition, no roads should be disconnected, which means, from every point in the network one has to be able to reach every other point in the network: no isolated roads are allowed. As a last hard constraint, the inclusion of dead-ends is prohibited as they deteriorate the navigability of the road network. Note that also dead-ends, which already exist in the TLM3D dataset are not desirable, as there are simply too many distributed across all of the four test regions. This topic will be discussed at a later stage of this thesis more comprehensively. All of these constraints can be checked very easily by a topological algorithm.

As soft constraints, mainly rather general statements were used. Nevertheless, this does not mean that they are in any way less important - in fact, the first soft constraint, which ensures that the overall thinning of the road network is appropriate for the target scale, could be considered as the most important constraint. The second and third soft constraints, which consider the general structure of the network, are intended to ensure that also the appropriate roads are selected in the end result. Finally, as a last constraint, the inclusion of important link roads needs to be ensured. The soft constraints cannot be checked easily by an algorithm but one has to resort to visual inspection.

The approach and algorithms of this thesis were implemented with these constraints in mind. While some of the constraints seem to be straightforward (such as retaining all highways), they entail other, more difficult to solve constraints (such as ensuring the connectivity of the highway entries and exits). It can also be seen that the structural constraints, which seek to preserve some kind of general property of the original network, are all considered as soft constraints. As mentioned above, hard constraints can be verified relatively easily by visual inspection or by using an algorithm if a quantification is necessary. This is less the case for soft constraints, which need to be evaluated in a more qualitative way. Thus, they were chosen in such a way that they can be evaluated qualitatively by swisstopo experts.

4 The Stroke-based Centrality Approach

4.1 Overview

The approach which serves as the main basis for the proposed methodology was first introduced by Jiang and Claramunt (2004a). It contains elements of graph-theoretic principles (Mackaness and Beard, 1993) as well as stroke-based (Thomson and Richardson, 1999) and semantic-based selection. The main idea behind this approach is to retain the general structure of a road network by extracting only the most central or important roads.

Figure 4.1 shows an overview of this approach. After creating a graph based on the line features of the roads, strokes are generated based on the principle of good continuation. Afterward, the dual graph is produced on which the calculations for the centrality measures are executed. As a last step, a threshold for a specific centrality measure is chosen, which produces the final pruned road network. In Section 4.3, each step is explained in detail.

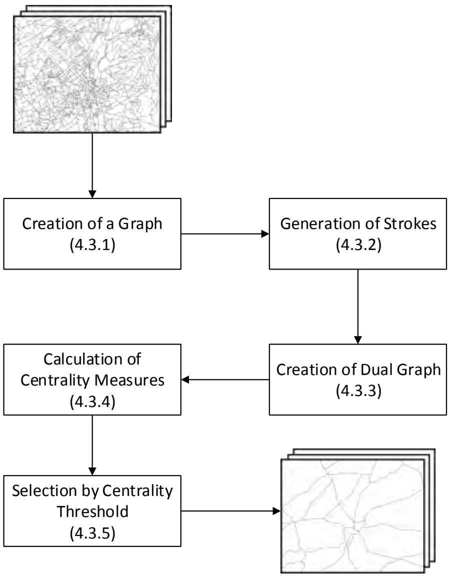


Figure 4.1: Overview of the basic stroke-based centrality approach which is used by Jiang and Claramunt (2004a).

4.2 Software Prototype and Utilized Software

Before starting with the methods section, the utilized software and hardware in this thesis is shortly presented. The implementation of all algorithms is based on Java. This programming language was chosen for two reasons: 1) I have gathered much experience in this language and 2) the GeoTools Java library[1] was used to ease the data handling and to re-use many geometric algorithms. As a programming environment, the Eclipse[2] IDE (Integrated Development Environment) was used. To analyze and visualize the results, ESRI ArcGIS[3] was used. For the statistical analysis, Microsoft Excel 2010 and IBM SPSS Statistics[4], in addition to LaTeX-internal features were used. For details about the software prototype, please refer to Appendix A.

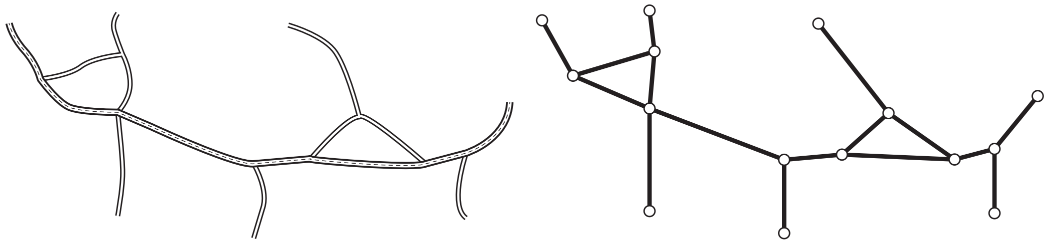
4.3 Methods

4.3.1 Graph

Graph theoretic principles have been used in geography for several years (Mackaness and Beard, 1993). While the general theory behind graphs is widely known, an abstract view of graphs based on the definition in Goodrich et al. (2004) and Jiang and Claramunt (2004b) is shortly presented, because the understanding of the theory and data structure behind it is very important for the understanding of the different algorithms which follow. Furthermore, it serves as a basis for the terminology used in this thesis.

A **graph** G is defined by a set N of **nodes** and a set E of pairs of nodes, called **edges**. Thus, a graph is a way of representing connections between pairs of objects from some finite set V (Goodrich et al., 2004). Edges in a graph are either directed or undirected. An edge which connects two nodes u and v, denoted as $E(u, v)$, is said to be **directed** from u to v if the pair (u, v) is ordered, with u preceding v. An edge $E(u, v)$ is said to be **undirected** if the pair (u, v) is unordered. Thus, in the undirected case, the pair (u, v) is the same as (v, u). A **subgraph** of a graph G is a graph H whose vertices and edges are subsets of the vertices and edges of G, respectively (Goodrich et al., 2004). A graph is **connected** if, for any two vertices, there exists a path between them. If a graph G is **unconnected**, its maximal connected subgraphs are called the **connected components** of G. A **weighted** graph is a graph that has a numeric label $w(e)$ associated with each edge e, called the **weight** of e. A graph is **unweighted** if the edges are unlabeled. An example of a graph is shown in Figure 4.2.

[1] `http://www.geotools.org/`
[2] `http://www.eclipse.org`
[3] `http://www.esri.com/`
[4] `http://www-01.ibm.com/software/ch/de/analytics/spss/products/statistics/`



(a) (b)

Figure 4.2: An example of the graphical representation of a graph (b) which is based on a road network (a). The numbered edges represent the different segments of the network. This representation is also called the primal representation, as linear features stay linear features in the graph (edges) and zero-dimensional features (vertices which connect two segments) stay zero-dimensional in the graph (nodes; based on Jiang et al., 2008).

To represent a street network as a graph, the road segments are first read and their start- and end-coordinates extracted. For each unique coordinate, a new node is created. Afterwards, the segments with matching start- or end-coordinates are attached to the appropriate nodes. This is called the **primal** or **direct representation**, because one-dimensional features (the road segments) are also represented in one dimension in the graph (edges) and zero-dimensional features (the end-points of the segments) are represented in zero dimensions (nodes; Porta et al., 2006b). Note that neither labels (weights) nor directions were assigned to the edges. Thus, the generated graph is undirected and unweighted. However, it is not clear whether the graph is connected or not: while unconnected parts are not allowed in the TLM3D, they can be created when the test regions have been extracted, as we essentially cut out a certain part of the network which could cause roads to get disconnected. However, unconnected components are undesired in a map which is used for navigation purposes, because there exists no possibility to reach them from every point in the road network. To solve this problem, only the largest connected component is kept and smaller ones are deleted. Hence, the resulting graph after this step of the approach is **unweighted, undirected and connected**.

4.3.2 Strokes

At the beginning of this step, an unweighted, undirected and connected graph has been created with the road segments represented as edges. Although it would be possible to build an algorithm on the segment level (i.e. the primal graph of the road network), this does not represent the network really well compared to how humans see a road network. A cartographer does not assess the importance of individual segments, but of interconnected roads as a whole. It would be preferrable if the algorithm would function in a similar fashion. Thus, the algorithm which prunes the network should not base its decision on individual segments, but on whole roads. For this purpose, Jiang and Claramunt (2004a) used road names to concatenate individual segments to build roads. However, road names are not always available in spatial databases (and as a matter of fact, the TLM3D database of swisstopo does only provide them for some roads). Hence, there needs to be a way to concatenate segments to build roads without the need of road names. That is where **strokes** enter the scene. Strokes were introduced by Thomson and Richardson (1999) and Zhang (2005) in an effort to provide an efficient generalization and abstraction of network data (mainly road and river networks). Strokes are nothing else than multiple linear segments, which are concatenated using the perceptual grouping principle of good continuation (see Figure 4.3; Thomson and Brooks, 2000). They may contain any positive number of nodes and segments, and may intersect each other or themselves. The perceptual grouping principle of good continuation states that elements that appear to follow in the same direction tend to be perceived as a group (Cohen and Ward, 1989). Thus, the human visual system spontaneously organizes elements of the visual field, even with no high level or semantic knowledge available (Thomson and Brooks, 2000).

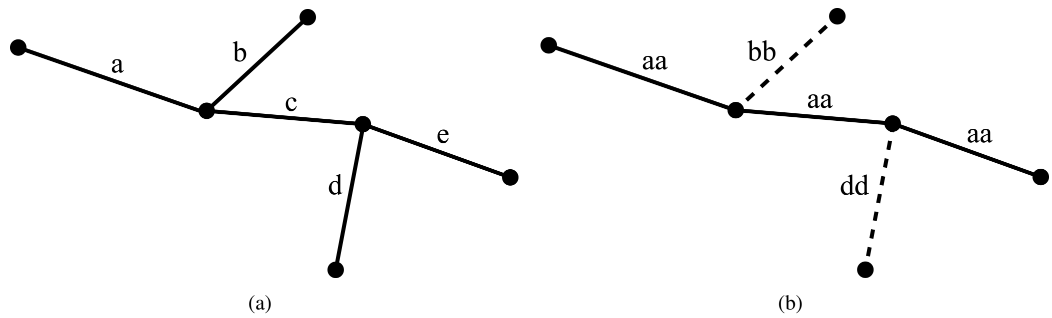


Figure 4.3: Concatenation of road segments into strokes: (a) shows five different road segments and (b) the resulting strokes (based on Zhou and Li, 2012).

There exist several approaches to build strokes (Zhou and Li, 2012). In this section, only the traditional approach, as described by Thomson and Brooks (2000), is discussed. The approach is purely geometric, while other approaches may include different thematic attributes, such as road names. Hence, a simple geometric criterion, the deflection angle from 180° of the angle between two road segments, is used to determine which segments should be concatenated, as depicted in Figure 4.4 (Zhou and Li, 2012). A smaller deflection angle results in a straighter line, with a deflection angle of 0° resulting in a perfectly straight line. During the concatenation of segments, a decision has to be made regarding the angle threshold. The threshold choice directly influences the number of generated strokes: a lower threshold leads to the creation of more strokes than a higher threshold. Several studies regarding this criterion have been made in the past and Chaudhry and Mackaness (2005) suggested to use an angle threshold of 40° to 60° to ensure that all strokes follow the principle of good continuity.

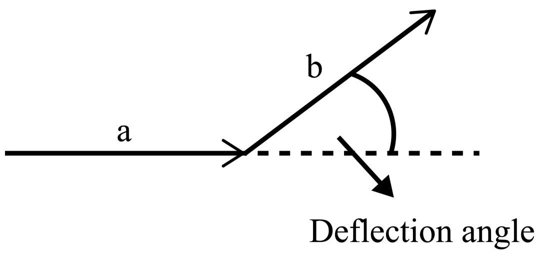


Figure 4.4: Interpretation of the deflection angle between two road segments (based on Zhou and Li, 2012).

For the concatenation of road segments into strokes, Jiang et al. (2008) list three different join strategies: *self-fit*, *self-best-fit*, and *every-best-fit*. *Self-fit* is the simplest of the three strategies: each segment at an intersection arbitrarily selects another segment to concatenate, whose angle is below the threshold. Using the *self-best-fit* strategy, each segment at an intersection selects the best fit (i.e. the segment with which it forms the smallest angle). The last strategy *every-best-fit* first looks at every possible pair combination at an intersection and combines the two segments with the best (lowest) angle. Note that both the *self-fit* and *self-best-fit* strategies depend on the order in which the segments are processed. As an example, we consider the situation at the intersection which connects segment a, b and c in Figure 4.3(a) using the *self-best-fit* strategy. If we start with segment b and assume that the angle to a is below a given threshold, then segment b would concatenate itself to segment a to form a new stroke. Afterwards, there is only one segment (c) remaining, which means that the algorithm would continue with the next intersection: segment a and c, which seem the obvious choice for a concatenation, remain separated. However, if we would have started with segment a or segment c, the result would look exactly as in Figure 4.3(b). This randomness in case of the *self-fit* and *self-best-fit* strategies and the dependency on the processing order of the road segments is not wanted, however. This is why for the algorithm used in this basic approach, the *every-best-fit* strategy was chosen, which also proved to be the best solution of the three according to experiments conducted by Zhou and Li (2012).

Stroke Building. Figure 4.5 shows an abstract diagram of the algorithm that builds the strokes. The algorithm starts at a random node and retrieves all adjacent edges. Afterwards, all possible edge-pair combinations are formed and the angle between them is calculated. After the edge-pair combinations and their angle have been calculated, they are sorted according to their angle in ascending order, such that the best combination with the smallest angle is at the first position of the sorted list. The algorithm iterates through the list until it is empty and selects the first (i.e. best) pair of edges and concatenates them, if their angle is below the selected threshold (in this case, 60° was chosen). If either the list of pairs is empty or the angle of the first pair is above the threshold, the node is marked as visited and the next node of the graph gets selected. When a pair is concatenated, the algorithm checks whether one or both of the pairs are already part of a stroke. If this is the case, either the segment is added to the existing stroke or the two strokes are merged. If neither of the segments is contained in a stroke, a new stroke consisting of two segments is created. Afterwards, the next segment in the list is selected and the algorithm iterates further. The stroke building process is finished, when all nodes of the graph have been visited.

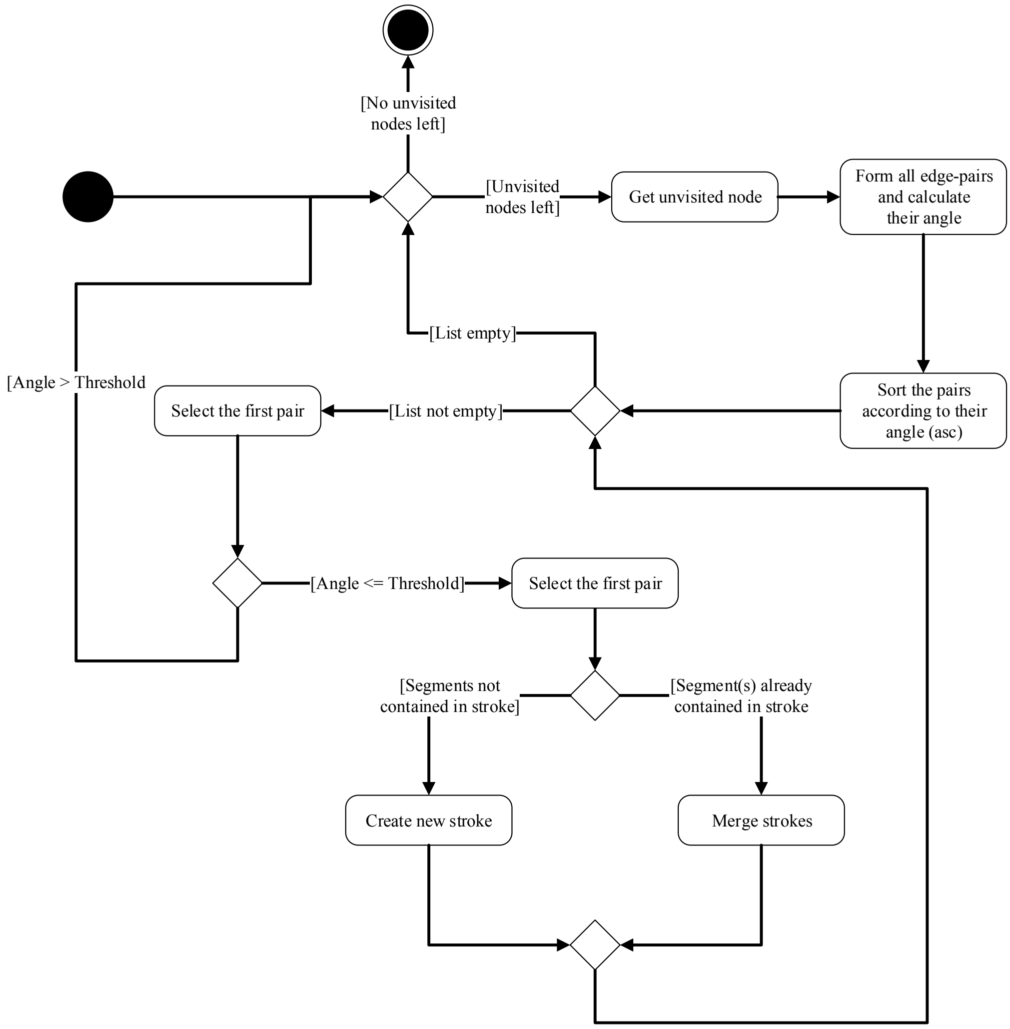


Figure 4.5: Stroke building process depicted as an UML activity diagram. The algorithm essentially consists of two loops: the outer one iterates through all the nodes in the graph and the inner loop iterates over the generated edge-pair combinations. The strategy used to concatenate the segments is the so called *every-best-fit* strategy, as discussed in Jiang et al. (2008) and Zhou and Li (2012).

Angle threshold. In a purely geometric approach, the only parameter which can be varied during the building of strokes is the angle threshold. As mentioned before, the maximum angle should not exceed 60°, as this would violate the perceptual grouping principle of good continuation (Chaudhry and Mackaness, 2005). Figure 4.6 shows the number of generated strokes in the Zurich test region depending on the chosen threshold angle. As can be seen, the number of generated strokes decreases as the angle threshold increases. The resulting curve is in line with Jiang et al. (2008) and is to be expected: the larger the angle threshold, the more segments can be concatenated, thus reducing the number of different strokes.

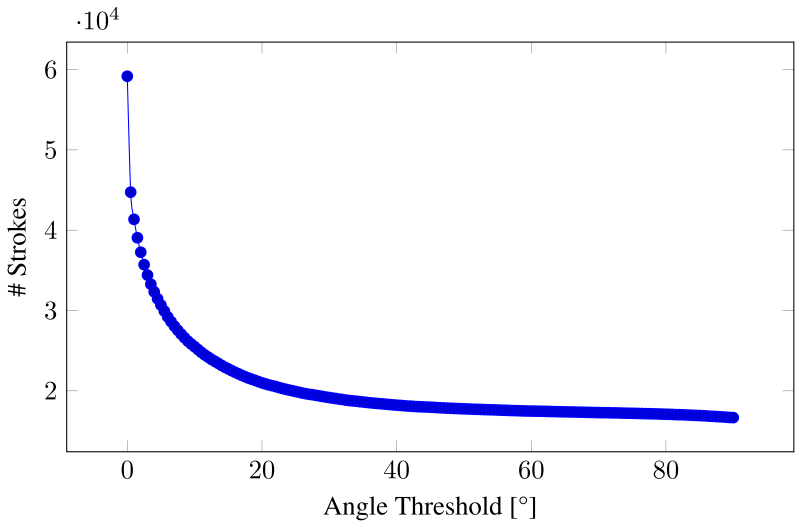


Figure 4.6: The number of strokes decreases as the angle threshold increases: the lower the threshold, the more segments are concatenated, which results in less and longer strokes.

4.3.3 Dual Graph

The theory and the main terminology of graphs have already been introduced in Section 4.3.1. The theory and terminology are the same for so-called dual graphs. The main difference lies in the representation of the road segments and their intersections in the graph. As already discussed in the graph section, one-dimensional elements become edges and zero-dimensional elements become nodes in the direct representation. However, the opposite is the case in the dual graph: in this representation, the one-dimensional segments are represented as nodes and the zero-dimensional intersections as edges (Porta et al., 2006a,b). This representation is also known as the **connectivity graph** (Jiang et al., 2008). An example of such a graph is shown in Figure 4.7, where the dual graph of a fictional road network is presented.

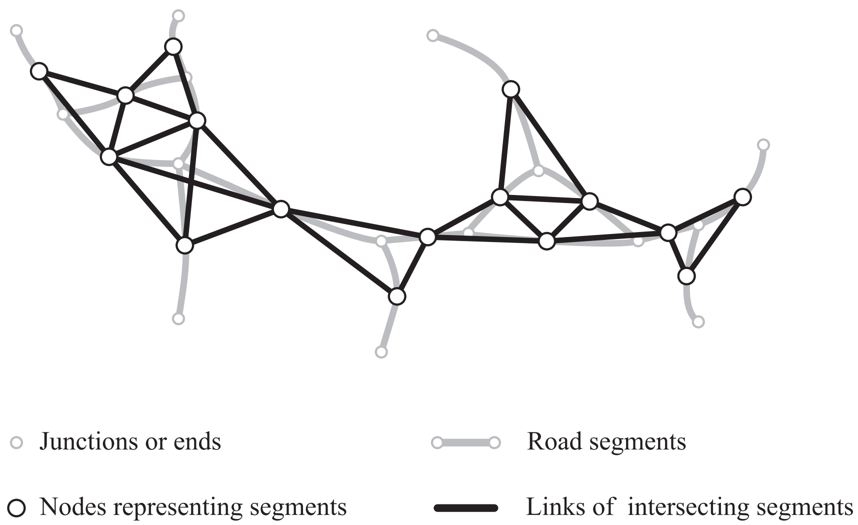


Figure 4.7: The road network (grey) and its dual graph. The nodes of the dual graph represent the segments and are placed in the center of the segments. The edges represent the links between the segments of the road network (based on Jiang et al., 2008).

Instead of representing each segment individually as a node, Jiang and Claramunt (2004a) used the previously generated strokes (in their case, they use the named streets). Such an example is given in Figure 4.8. In this representation, the nodes represent the color-coded strokes. The edges represent the connections between them: an edge shows whether two strokes connect to or intersect each other. This representation reduces the complexity of the graph and uses the strokes as the main actor. This is important, as we want the algorithm to handle roads as a whole and not each segment individually.

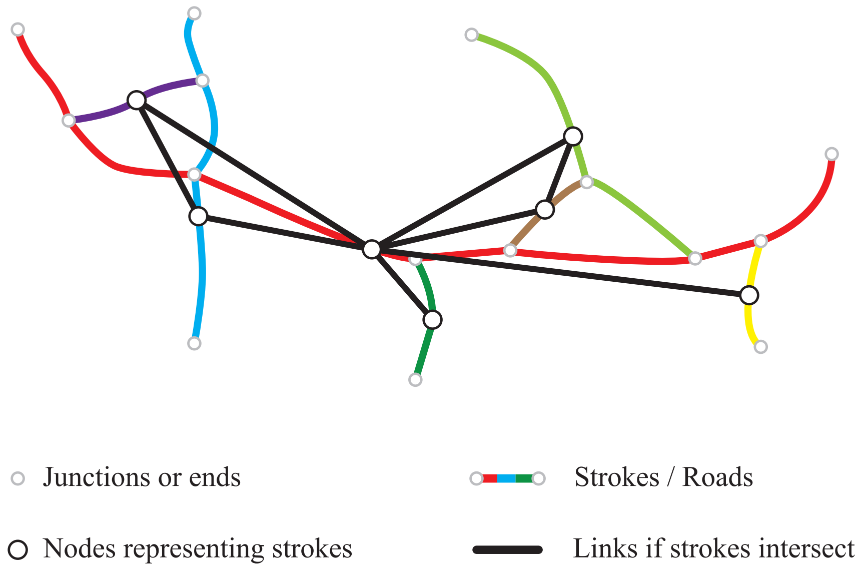


Figure 4.8: The road network with colored strokes, which represent roads. The corresponding dual graph represents the strokes as nodes and the intersections between them as edges (based on Jiang et al., 2008).

The method used to create the dual graph is depicted in Figure 4.9. The nodes are simply created by inserting all previously built strokes into new nodes. The creation of the edges, which represent the intersections between the strokes, is a bit more complex. As explained in Section 4.3.2, strokes consist of the concatenation of individual road segments. To check whether a stroke intersects another stroke, one needs to check whether any segment of the first stroke intersects any other segment of the second stroke. This is simply done by comparing each start- or end-coordinate of each segment of the first stroke to all coordinates of the segments of the second stroke. If a coordinate match occurs, the two strokes intersect and an edge between the two appropriate nodes is inserted.

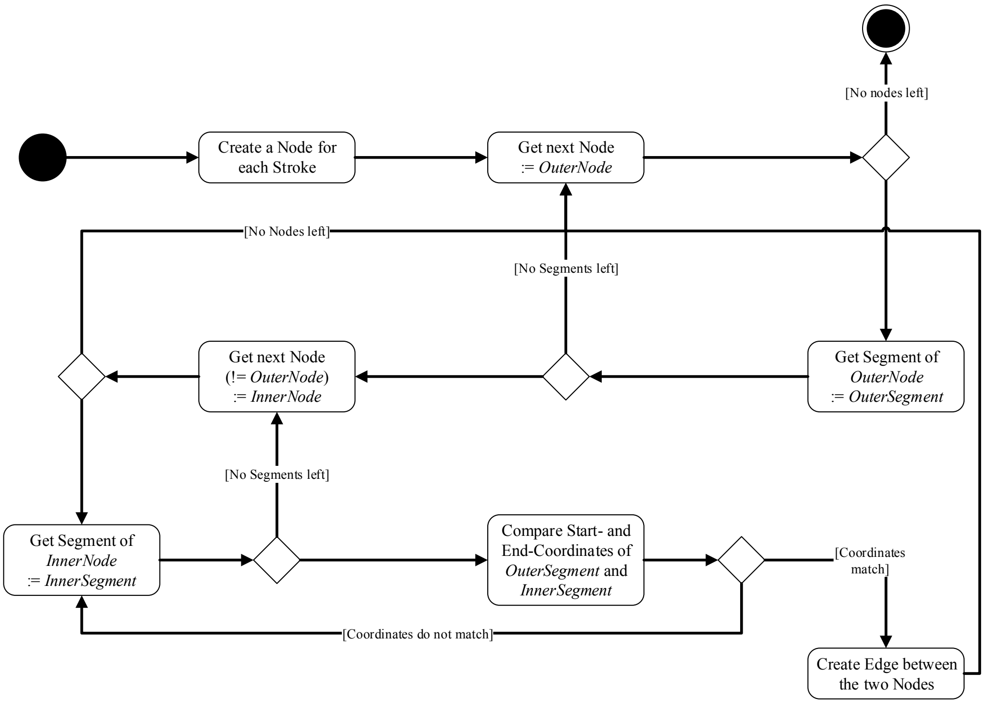


Figure 4.9: UML activity diagram of the creation of a dual graph based on strokes.

4.3.4 Centrality Measures

As discussed in the overview of this chapter, the main idea behind the approach is to determine the most *important* roads in a road network. To determine the importance of a road, Jiang and Claramunt (2004a) used centrality indices, which were introduced several decades ago (Freeman, 1977; Brandes, 2001). The use of centrality indices originated in the field of social network analysis (Bavelas, 1948; Sabidussi, 1966; Freeman, 1978; Brandes, 2001). An example for such a study is Newman (2001), who used such measures to analyze the collaborations of scientists. Centrality measures have also been used in other fields, for example to analyze and optimize mobile networks or to analyze the importance of proteins (Joy et al., 2005; Daly and Haahr, 2007).

To grasp the principle behind the centrality of a point in a graph, let us consider Figure 4.10. It is intuitively clear that the point P_3 of the star formation is the most central point. The problem is, however, to determine ways in which such a central position is structurally defined. Freeman (1978) and others have come up with three distinct structural properties that are uniquely possessed by the center of the star: The position has the maximum possible *degree* (number of neighbors), it falls on the geodesics (i.e. shortest paths) *between* the largest possible number of other points and it is maximally *close* to the others.

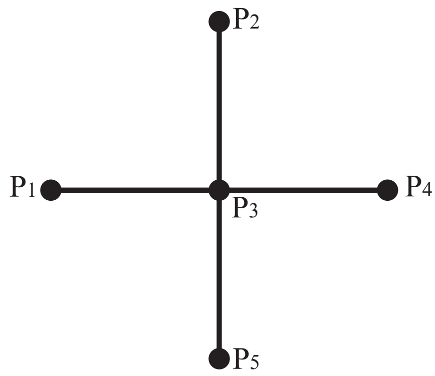


Figure 4.10: A star or wheel with five points to illustrate the concept behind centrality measures (based on Freeman, 1978).

Jiang and Claramunt (2004a,b) were the first who applied these concepts and their related indices in the context of road network selection. While three indices can be applied in this context, this thesis concentrates on the so-called betweenness centrality. The reason for this will be explained after the different centrality measures have been introduced.

Degree centrality. The simplest and most intuitively obvious centrality measure is the degree centrality. In a graph, the degree of a node v_i is simply the count of the number of nodes, v_k ($i \neq k$), that are directly adjacent to it (Freeman, 1978). It is a local index of the number of direct neighbors of a node v_i. The degree of a node v_i is formally defined by

$$C_D(v_i) = \sum_{k=1}^{n} e(v_i, v_k) \tag{4.1}$$

where n is the total number of vertices of a graph G and $e(v_i, v_k)$ is the number of edges between two nodes v_i and v_k. Note that in all graphs that are used for the algorithm, two nodes are either connected or not - there are no multiple edges between two segments, in the case of the primal graph, or two strokes, in the case of the dual graph. Thus, $e(v_i, v_k)$ simply denotes whether an edge between the two nodes exists: it is either 1 or 0.

Closeness centrality. Closeness centrality describes how close a node is to other nodes. Thus, closeness centrality is the shortest distance from a given node to all other nodes (Jiang and Claramunt, 2004a). In the dual graph, the distance is of topological nature, which means that it is simply the number of links from one node to all other nodes. It is defined by the following formula

$$C_C(v_i) = \frac{n-1}{\sum_{k=1}^{n} d(v_i, v_k)} \tag{4.2}$$

where $d(v_i, v_k)$ is the shortest distance between nodes v_i and v_k (Jiang and Claramunt, 2004a). Other known terms for closeness centrality are status, integration or the reciprocal value of the average path length (Jiang and Claramunt, 2004b).

Betweenness centrality. Betweenness centrality measures to what extent a node is located in between the shortest paths that connect pairs of nodes (Jiang and Claramunt, 2004a). It measures how important a node is for the shortest paths between other nodes or whether the node has a bridging role in a graph. According to Brandes (2008), the *betweenness* $C_B(v)$ of a vertex $v \in V$, where V is the set of nodes and E the set of edges in a graph G is defined as

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{4.3}$$

where $\sigma(s,t)$ is the number of shortest (s,t)-paths (also called geodesics) and $\sigma(s,t|v)$ the number of shortest (s,t)-paths passing through node v. If $s = t$, let $\sigma(s,t) = 1$, and if $v \in \{s,t\}$, let $\sigma(s,t|v) = 0$ (Brandes, 2008). By convention, let $\frac{0}{0} = 0$. The measure can be interpreted as the degree to which a node has control over pair-wise connections between other nodes, assuming that the importance of connections is equally divided among all shortest paths of each pair (Brandes, 2008).

The algorithm used to compute the betweenness centrality of each node is based on Brandes (2001) and Brandes (2008). By exploiting the fact that the cubic number of *pair-wise dependencies* $\delta(s,t|v) = \frac{\sigma(s,t|v)}{\sigma(s,t)}$ can be aggregated without computing all of them explicitly, the running time of the algorithm can be reduced drastically, which allows us to use betweenness centrality indices even for large road networks. The reason for this is explained very thoroughly in Brandes (2001), but its essence can be explained very briefly. If we define *one-sided dependencies* as

$$\delta(s|v) = \sum_{t \in V} \delta(s,t|v), \tag{4.4}$$

for all $s, v, w \in V$, we can exploit that

$$\delta(s|v) = \sum_{\substack{(v,w) \in E \ and \\ dist(s,w) = dist(s,v)+1}} \frac{\sigma(s,v)}{\sigma(s,w)} \cdot (1 + \delta(s,w)). \tag{4.5}$$

as shown in (Brandes, 2001). This recursive relation asserts that the dependency of node s on some v can be compiled from the dependencies on nodes one edge farther away (Brandes, 2008).

The algorithm used will be explained in further detail, as it is the backbone of the whole approach. The betweenness is computed by iterating over all nodes $s \in V$, each time generating $\delta(s|v)$ for all $v \in V$ in two phases. In the first phase, a breath-first search (BFS; see Goodrich et al., 2004) is performed, where the distances and shortest path counts from s to all other nodes are determined (Brandes, 2008). In the second phase, all nodes are visited in reverse order of their discovery (those farthest from s first), to accumulate the dependencies according to Equation 4.5.

Table 4.1 lists the the input and output as well as the main data structures used in the algorithm. As an input, we have the dual graph $G = (V,E)$, consisting of a set of nodes, V, which represent the strokes. The edges, E, represent the interconnections between the strokes. For the shortest-path problem, we need a queue, Q, and for the accumulation a stack, S, which are used to store nodes. Additionally, for every node $v \in V$ we need four arrays, containing floating point numbers. All four of them are listed in Table 4.1.

Table 4.1: Input, output and the data structures needed for the shortest-path vertex betweenness algorithm according to Brandes (2001, 2008). The main data structures used are a queue Q, which is needed for the single-source shortest-path problem in the first phase and a stack S, which is filled in the first phase and emptied in the second phase.

input:	graph $G = (V, E)$
data:	queue Q, stack S (both initially empty) and for all $v \in V$:
	$dist[v]$: distance from source
	$Pred[v]$: list of predecessors on shortest paths from source
	$\sigma[v]$: number of shortest paths from source to $v \in V$
	$\delta[v]$: dependency of source on $v \in V$
output:	betweenness $C_B(v)$ for all $v \in V$ (initialized to 0)

Figure 4.11 depicts the whole algorithm in detail based on Brandes (2001, 2008). The algorithm performs for each node $s \in V$ the following three steps:

1. **Initialization:** For each node, the data is initialized (as listed in Table 4.1). Additionally, the distance δ of the source node s is set to 0, and the number of shortest paths σ for s is set to 1, as defined earlier. Afterwards, s is placed in the queue Q.

2. **Single-source shortest-path problem:** In this stage, the shortest paths from the source node s to all other nodes are discovered using a BFS. In the first iteration through the while-loop, the direct neighbors W of s are examined. For each neighbor w, $dist[w]$ is inspected. If it is still equal to infinity, i.e. the initialized value, this means that the node has not yet been discovered. Thus, $dist[w]$ gets updated to the current distance to the source node, which is simply the distance of the previous node plus one, or $dist[v] + 1$. The neighbor gets also enqueued in Q, such that the BFS may progress further after the algorithm is done with the first iteration of the while-loop. In the next step, the algorithm checks whether the distance of the neighbor w to s, $dist[w]$, equals to the distance of the current node + 1. If this is the case, this means that the edge (v, w) lies on a shortest path to s. Thus, the number of shortest paths of w gets updated and v is added to the predecessors of w. This loop gets repeated for every neighbor, until the queue Q is empty. As the dual graph created in Section 4.3.3 is connected, this means that it is guaranteed that every node has been visited.

3. **Accumulation:** At the beginning of this stage, the predecessors of every node w, stored in the list $Pred[w]$, as well as the number of shortest paths to s, stored in $\sigma[w]$, have already been calculated. In the previous stage, each node has been put on the stack S with non-decreasing distance from the source s. The nodes are popped from the stack one by one and their dependencies calculated. That is why this step is also called **back-propagation of dependencies**. For each predecessor v of node w, Equation 4.5 is evaluated and accumulated to $\delta[v]$. After the accumulation, the betweenness value $C_B[w]$ is updated with the accumulated dependencies, given that $w \neq s$.

After the accumulation phase has finished, the algorithm continues with the next source node s and starts again at the initialization phase until all nodes have been processed and the algorithm finishes.

Since $C_B(v)$ is essentially a count, its magnitude depends upon two factors: the structure of edges and the number of points in the graph (Freeman, 1977). However, it is desirable to eliminate the impact of the number of points from the measure. Hence, the measure needs to be normalized such that it is relative to its maximum value in terms of the number of points in the graph (Freeman, 1977). A node which falls on all shortest paths between other points receives the highest possible betweenness value. Such a situation exists in a star configuration, as previously shown in Figure 4.10. According to Freeman (1977), it can be shown that the upper limit of C_B is defined as

$$maxC_B(v) = \frac{n^2 - 3n + 2}{2}. \tag{4.6}$$

Hence, the relative centrality of any point in a graph may be expressed as the ratio

$$C'_B(v) = \frac{2C_B(v)}{n^2 - 3n + 2} \tag{4.7}$$

which implies that the values range from 0 to 1. Using such a normalization, it becomes possible to compare different graphs, regardless of the number of nodes (Freeman, 1977).

Parallelization. If we take another look at the diagram of the algorithm in Figure 4.11, it can be seen that the three main steps are performed for each node. Except for the very last step, the summation of dependencies, the algorithm uses exclusively local variables, which get initialized for every node separately. Thus, the whole algorithm can be parallelized relatively easily (i.e. it is *embarrassingly parallel*; see Wilkinson and Allen, 1999), by splitting up the source nodes, as depicted in Figure 4.12. Instead of iterating through all the nodes, the list of nodes first get split up to equally sized lists. The number of lists can be set either manually or it simply equals to the number of threads available on the computer system. Afterwards, each thread iterates over its list of nodes and takes each one as a source node once and computes the three main steps. The algorithm does not change otherwise, except that each thread stores the betweenness values for each node, $C_B(v)$, by itself. After all threads have finished with their iterations, the betweenness values are simply summed up to receive the final values, after which the algorithm is finished. The impact this parallelization has on the performance of the algorithm is shortly analyzed in Appendix B.

Omission of Degree and Closeness Centrality. As mentioned in the beginning of this section, the focus in this thesis lies on the betweenness centrality, while the other two centrality measures, degree and closeness centrality, will not be used to determine the importance of the strokes in the road network. Earlier studies have already shown that strokes in the suburban regions of networks may have biased ranking values when only degree centrality or closeness centrality measures are used (Yang et al., 2011). According to Jiang and Claramunt (2004a) and Tomko et al. (2008), both measures are unable to provide a reliable measurement of the structural importance of a stroke in a road network, as explained below.

Closeness centrality measures how topologically close a stroke is to other strokes. While it is possible to detect the longest and most important few strokes using closeness centrality, the usefulness of the measure quickly deteriorates as more strokes get selected. The reason for this is quite simple: consider a major stroke, which extends through the whole region. Such a stroke receives a high closeness value, as it is topologically very close to many other strokes. However, every stroke which directly connects to this major stroke, no matter how unimportant or short it is, also receives a very high closeness value, simply because it has direct access to the long stroke. Hence, closeness centrality can be used to determine how well a road is connected to the network, but it is not well suited to identify the structural importance of a road or stroke.

The omission of degree centrality has another reason: the measure only takes the local neighborhood into account. While this may be important in small networks or in the pruning of the road network for larger scales, it is less useful for the target scale of 1:200,000. In addition, the betweenness measure already includes the degree measure in some way: if a stroke is directly connected to many other strokes and lies on the shortest path of those strokes, its value is increased. If it does not lie on the shortest paths of the neighboring strokes, the betweenness value is not increased, in contrast to the degree value, which only acts locally and does not take into account shortest paths in any way.

4.3.5 Centrality-based Selection

The final step of the algorithm lies in the actual selection of the roads. This is simply done by setting a threshold on any of the centrality measures and omitting any stroke which has a lower value than the threshold. Finding an optimal threshold is rather difficult, however, as the different regions behave very differently and consist of a different road network structure. According to swisstopo, they also do not retain a specific number of segments for their VECTOR200 product. After a discussion with swisstopo, the thresholds for the different test regions were chosen such that the general road density of the region was comparable to the VECTOR200 product. However, a direct matching was not possible, as the VECTOR200 product already incorporates all of the generalization steps outlined in Section 2.1. Because the basic principle is the same in the improved version of the approach, the reasoning behind the chosen thresholds will be discussed more thoroughly in Section 5.4.

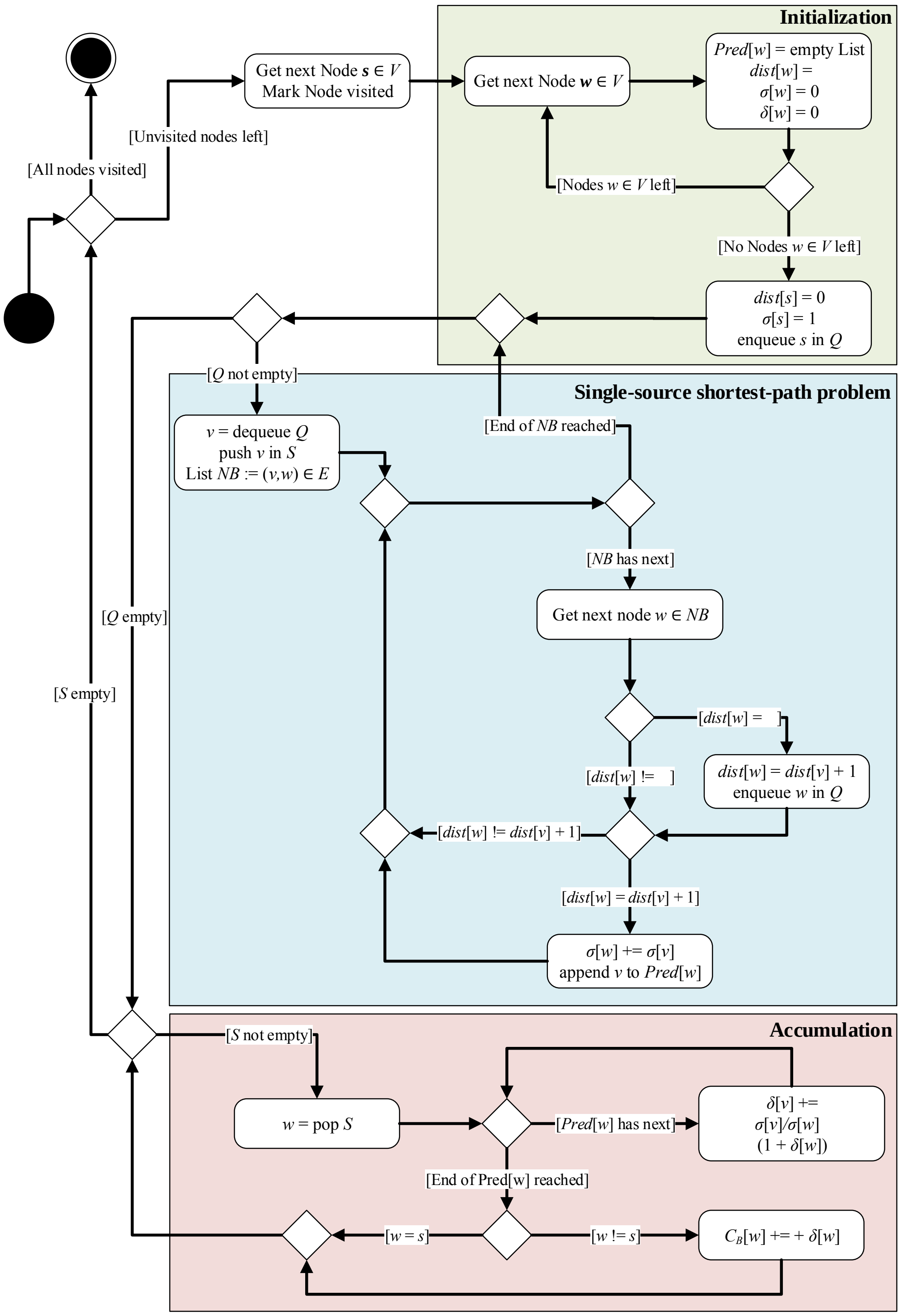


Figure 4.11: UML activity diagram of the shortest-path vertex betweenness algorithm (based on Brandes, 2001, 2008).

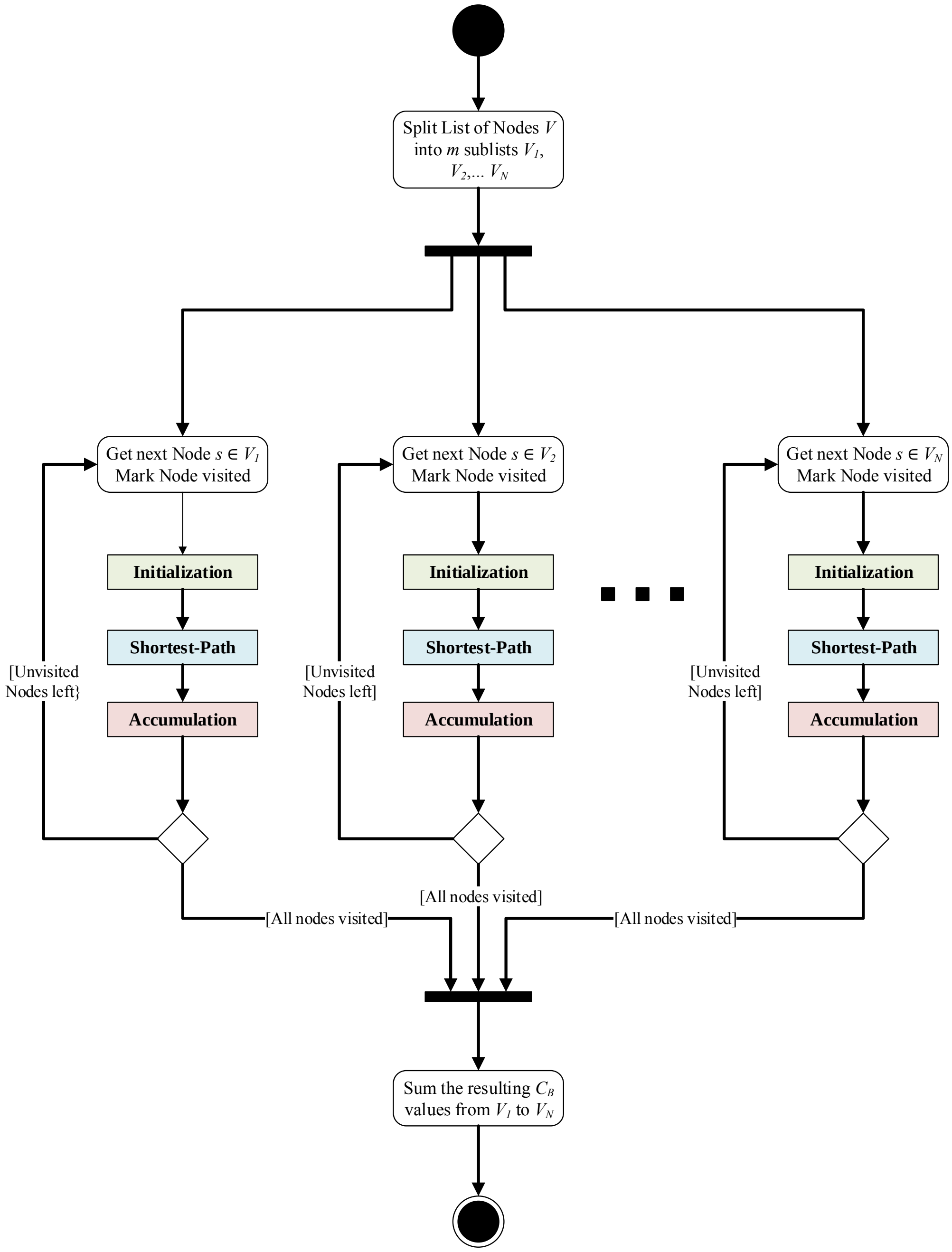


Figure 4.12: UML activity diagram of the parallelized algorithm running on N threads

4.4 Results and Discussion

4.4.1 Results

In Section 3.3.2, several important constraints to the target solution were introduced. While the hard constraints can be evaluated using the key figures in Table 4.2, the soft constraints can only be analyzed qualitatively. To do this credibly, the results should be evaluated by an expert. Because it is already clear from the analysis of the hard constraints that the basic approach is unable to fulfill the requirements satisfactorily, the soft constraints will not be evaluated in detail, but problems which are related to them will be revealed. Figures 4.13 to 4.16 show the final results of the four test regions using the basic approach.

Table 4.2: Key figures of the final result using the basic approach for the four test regions. Dead-ends at the edges of the test regions were excluded from the analysis, as they are an artefact of the extraction of the different regions and not of the selection algorithm.

	Davos	**Lucerne**	**Langenthal**	**Zurich**
SR[1]	2,416 (24.8%)	5,887 (22.4%)	5,529 (17.7%)	11,742 (17.9%)
HS[2]	0	269	0	362
HS missing	0	86	87	491
HL[3]**[m]**	0	98,665.4	0	134,472.8
HL missing[m]	0	13,915.2	42,796.9	122,639.4
HE[4]	0	26	0	22
HE missing	0	20	16	73
HE disconnected	0	2	0	1
DP[5]	1	1	0	1
DE[6]	52	176	102	339

[1] Segments Retained
[2] Highway Segments
[3] Total Highway Segment Length
[4] Highway Exits and Entries
[5] Disconnected Parts
[6] Dead-Ends

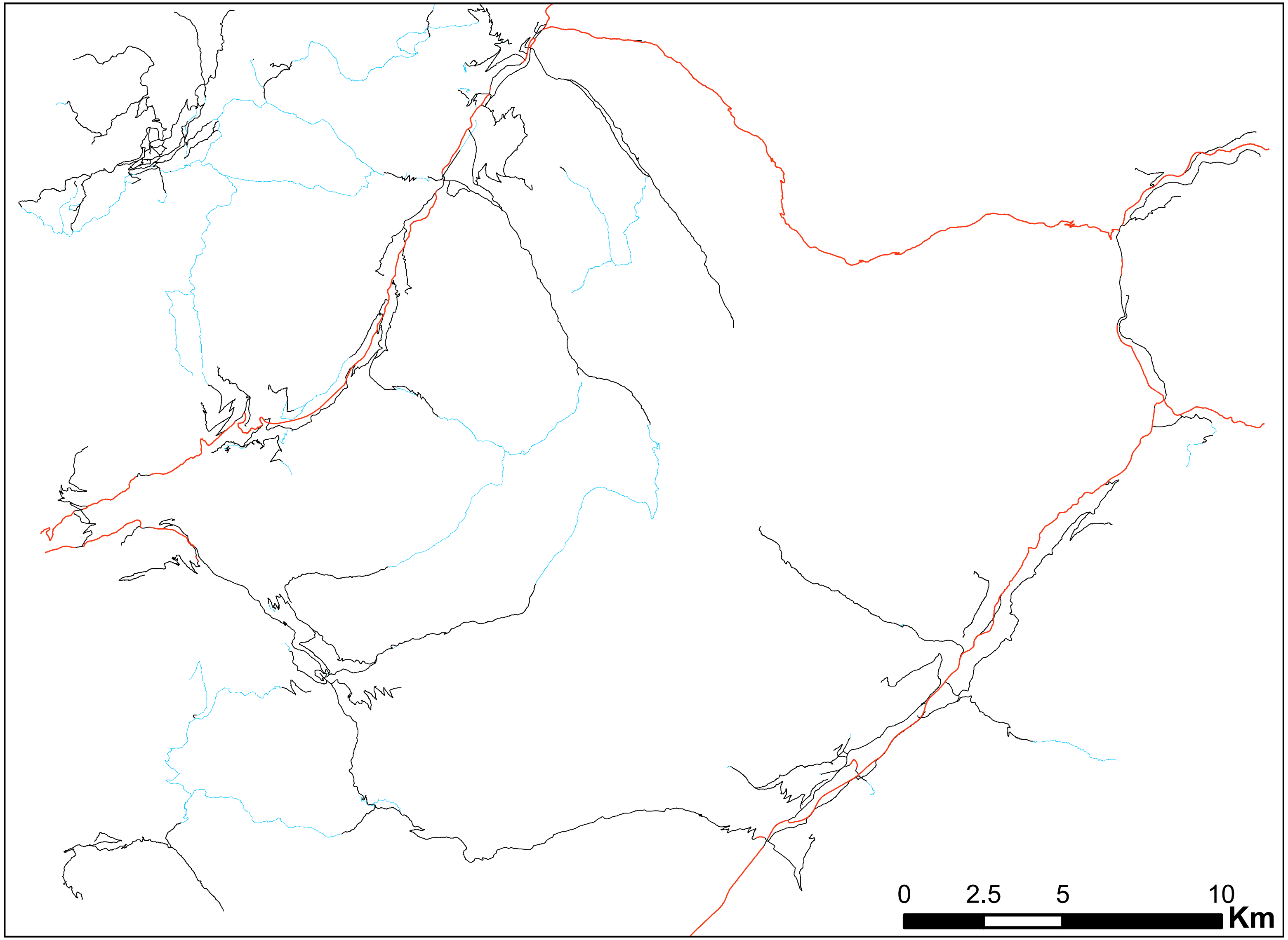


Figure 4.13: Final result of the basic approach for the Davos test area. Major roads (> 4m) are marked as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

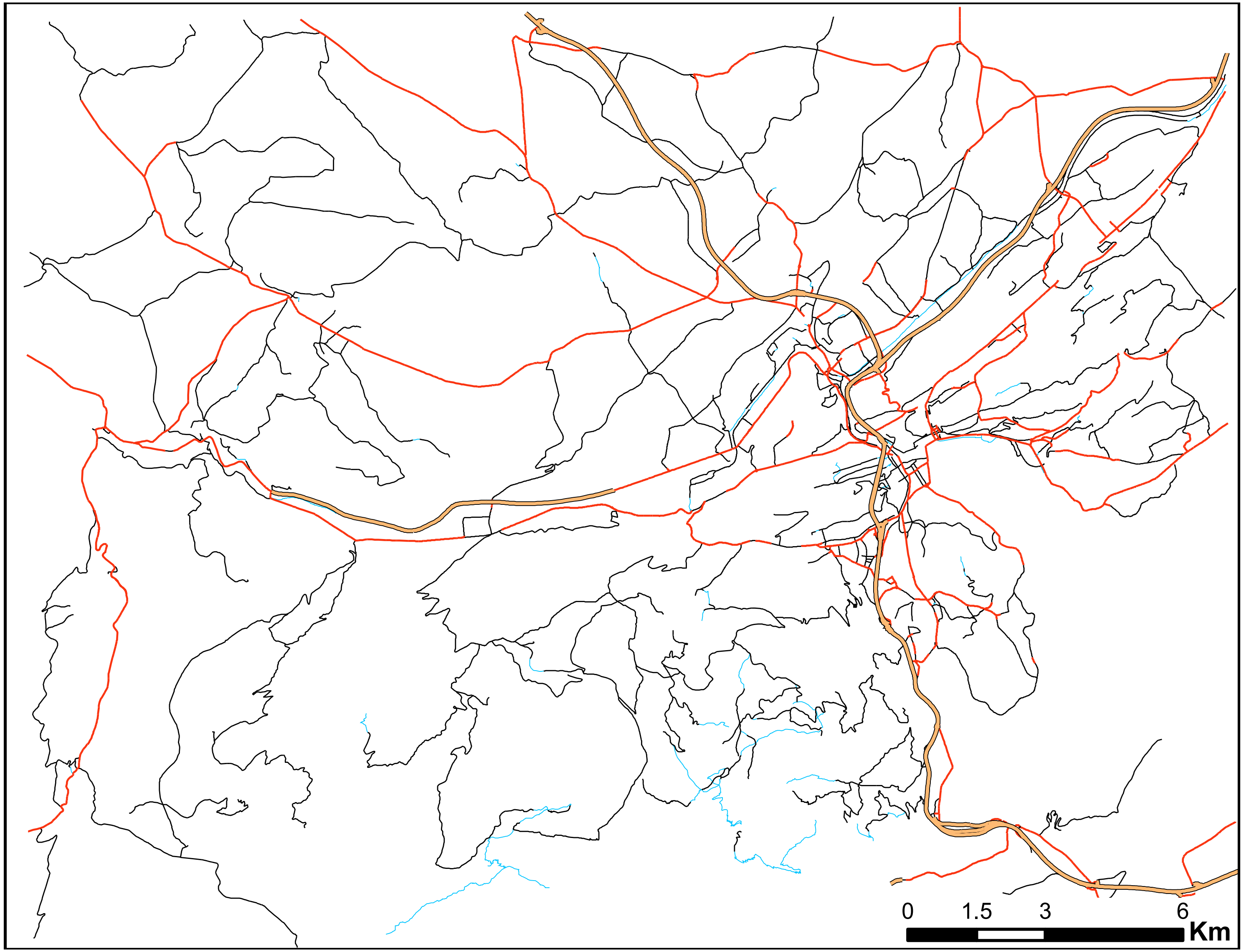


Figure 4.14: Final result of the basic approach for the Lucerne test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

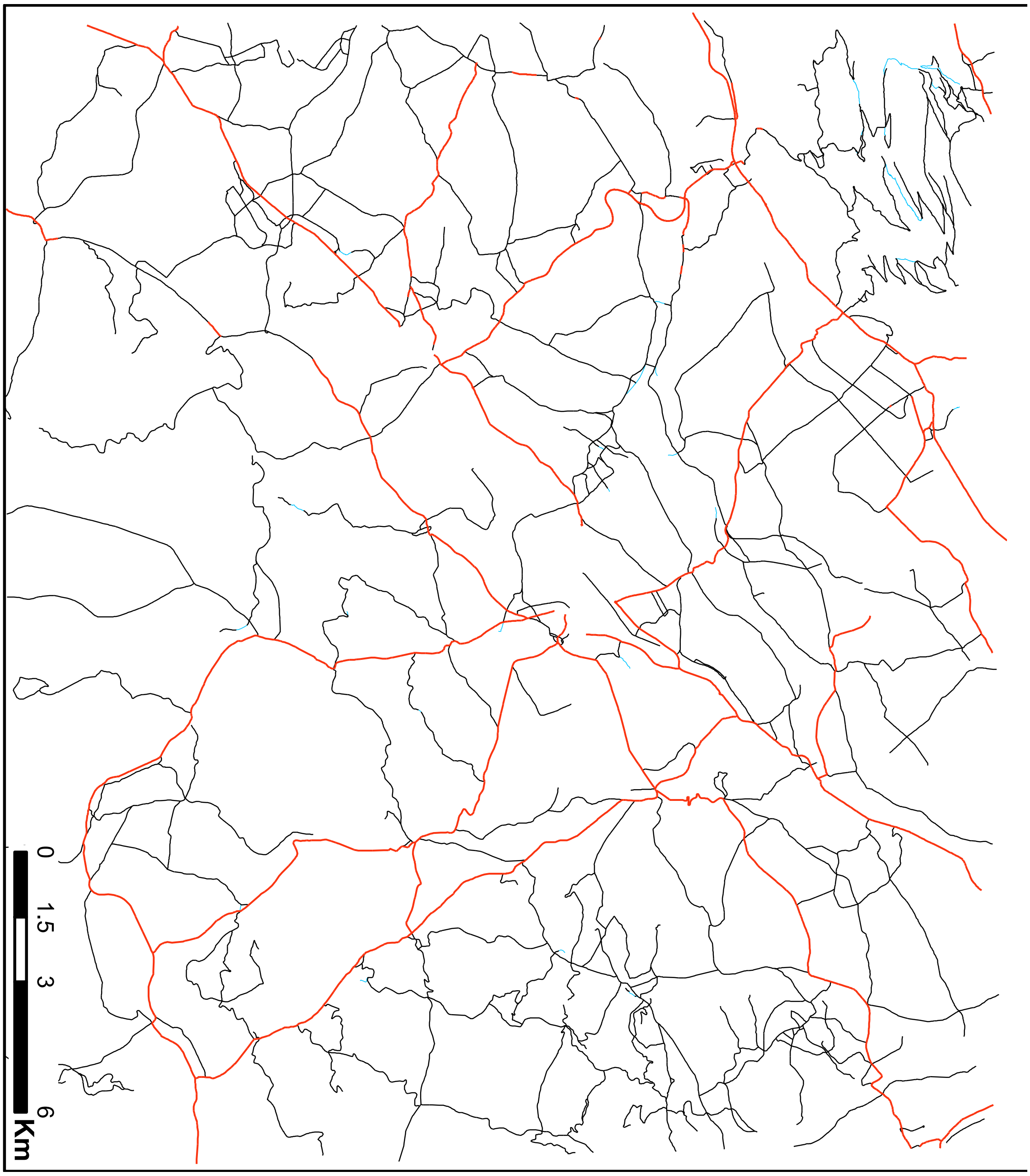


Figure 4.15: Final result of the basic approach for the Langenthal test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

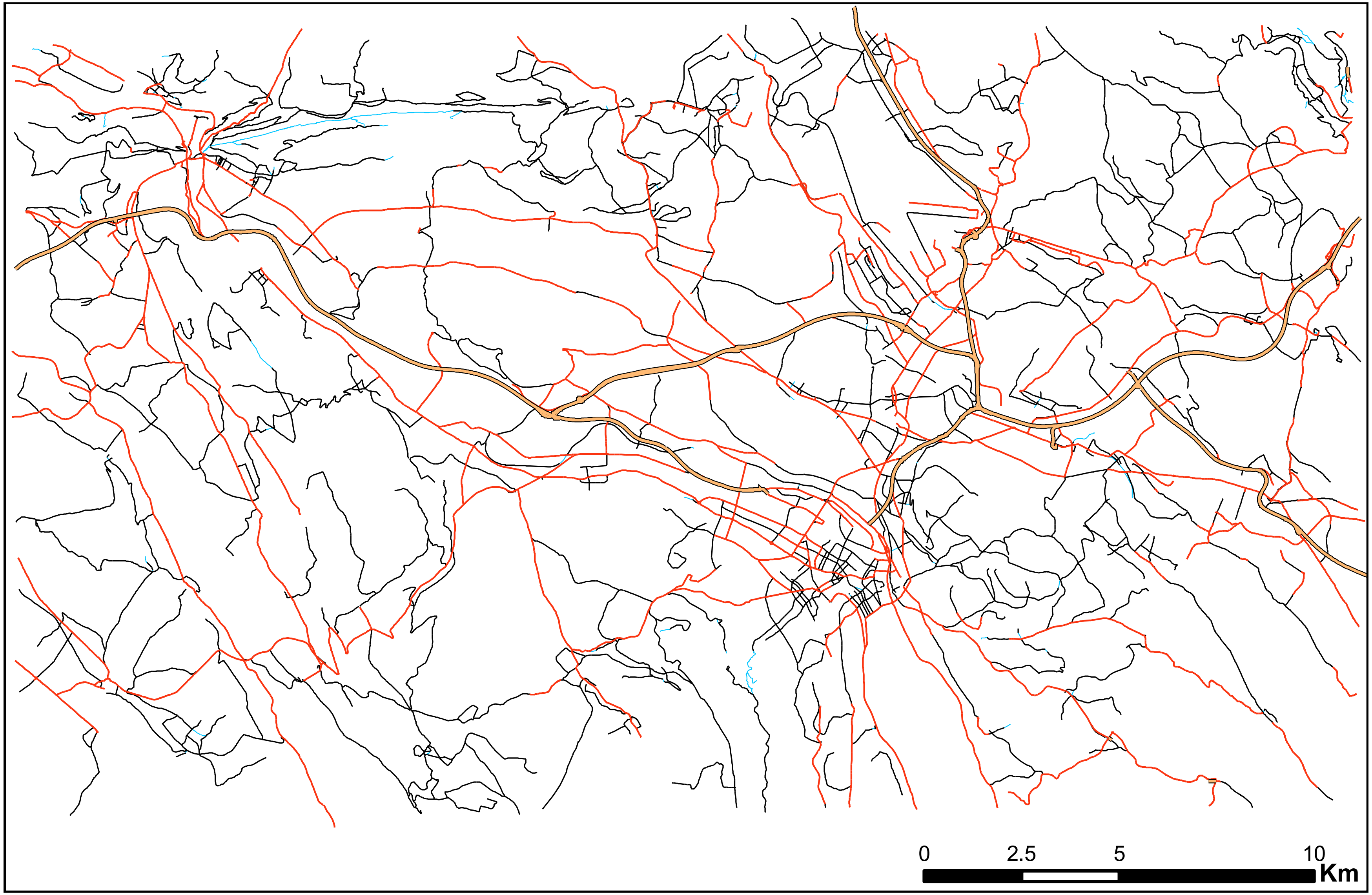


Figure 4.16: Final result of the basic approach for the Zurich test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

4.4.2 Analysis of the Hard Constraints

In this section, the hard constraints on the basis of Table 4.2 and Figures 4.13 to 4.16 will be analyzed. The underlying problems which lead to these (and additional) problems are discussed in the next section.

Constraint 1: All highways and expressways and everything related to them (such as entries and exits) must be included in the generalized map.

With the exception of Davos, all test areas contain highway segments. Table 4.2 reveals that in all of the test areas which contain highway segments, the constraint could clearly not be fulfilled. In Lucerne, close to 43 km of highway are missing, which is over one third of the total length. In Langenthal, not a single highway segment could be retained and also in Zurich over half of the segments are missing. Similarly, many highway entries and exits are missing. Hence, it is clear that the basic algorithm fails to reliably retain highways, as they do not always exhibit a high betweenness centrality. This problem could be fixed relatively easily in an enhanced version by simply forcing each segment related to highways to be retained, however.

Constraint 2: The entries and exits to and from the highways must be directly connected to the road network.

In the test areas that contain highway segments, not many entries and exits are disconnected (2 in Lucerne and 1 in Zurich). The numbers alone show an illusive picture, however, as many access roads to the highways are missing in the first place. The solution mentioned in the first constraint (forcing each highway segment to be retained) would obviously aggravate this problem, as most added entries and exits would be disconnected. Thus, while the constraint itself is only violated slightly, it has to be noted that the solution to it is much more complicated, especially if the missing exits and entries are added to the solution.

Constraint 3: The road network must be completely connected.

Disconnected parts exist in three of the test areas. Hence, they do not pose a big problem in the results. However, it is important to note that a slight change in the centrality threshold could produce other and more disconnected parts. Thus, while the constraint is not violated very strongly in this particular result, a result using a different threshold could contain more or larger disconnected parts. In addition, swisstopo explicitly stated that there must be no disconnected road under all circumstances, as it deteriorates the final map quite heavily. While an easy solution would be to simply delete the disconnected parts or to use a Minimum Spanning Tree (MST; Yang et al., 2011), the problem is closely related to the next constraint, which is not solved in either of the two proposed solutions.

Constraint 4: No dead-ends may exist in the final result.

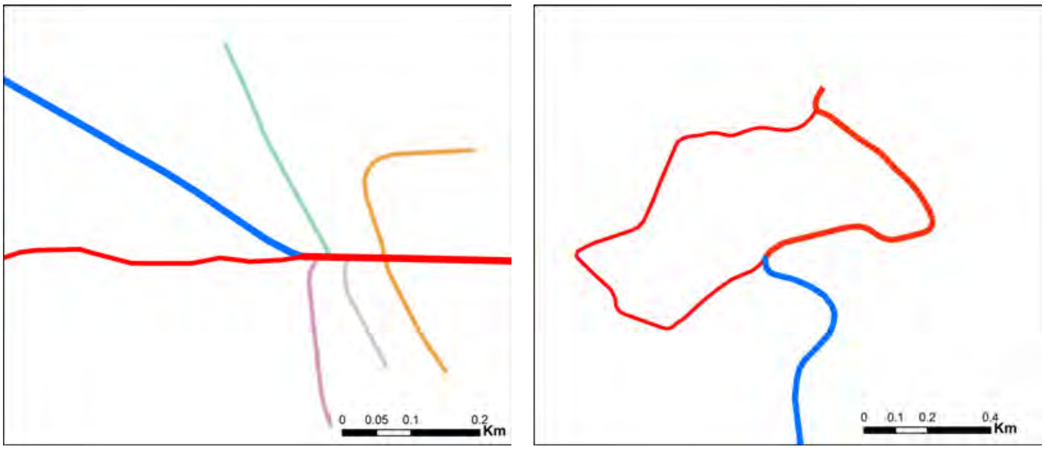
The last hard constraint prohibits dead-ends in the pruned road network. It is evident from Table 4.2 that all test areas violate this constraint heavily. The problem is very severe, as the dead-ends are speckled across all over the different areas (see Figures 4.13 to 4.16). The high number of dead-ends, ranging from 52 to 339 depending on the test area, points to a serious problem of the basic algorithm and stroke-based algorithms in general, as they inherently do not prevent the creation of new dead-ends. Note that no distinction between already existing and newly created dead-ends is made, as both types are generally unwanted in the target scale of 1:200,000.

4.4.3 Identified Problems

In this section, the problems identified in the basic approach will be shortly explained. Some of them are related to the violations of the hard constraints discussed in the previous section, while others are more related to possible violations of the soft constraints, which have not been discussed so far.

Rules for Stroke Generation. To generate the strokes, the only criterion used was the angle between two adjacent segments, as explained in Section 4.3.2. While this is the best option to preserve the principle of good continuation, which is the main concept behind the original stroke algorithm (Thomson and Richardson, 1999), several problems arise when looking at specific situations in the road network. Two such situations are depicted in Figure 4.17, where the best continuation according to the *every-best-fit* strategy using the angle as the only criterion is shown. Figure 4.17(a) shows a problematic situation in a dense region near the center of Zurich. Coming from the east, the red, 6 m wide segment is concatenated to the narrower, 4 m wide segment to form a stroke as they share the smallest angle. However, it is clear that the actual road continues to the northwest: the correct concatenation would be the blue stroke, such that the two 6 m roads are connected together. Figure 4.17(b) shows a similar situation. At the intersection where the blue and red stroke come together, clearly the wrong decision was made. The two segments with the smallest angles were concatenated, which causes the main, winding road to split up into two strokes. Hence, the question arises whether the use of the angle as the sole criterion to concatenate strokes is enough or whether the addition of the road type into the decision process would improve the concatenation accuracy. Another problem, which is not depicted in a Figure, is the threshold of only 60°. While it preserves the principle of good continuation, it also splits up winding roads consisting of hairpin bends and various roads in urban regions, where streets often make turns above 60°.

By enhancing the rules by which strokes are concatenated, the continuity of important link roads (constraint 8; see Section 3.3.2) could potentially be improved.



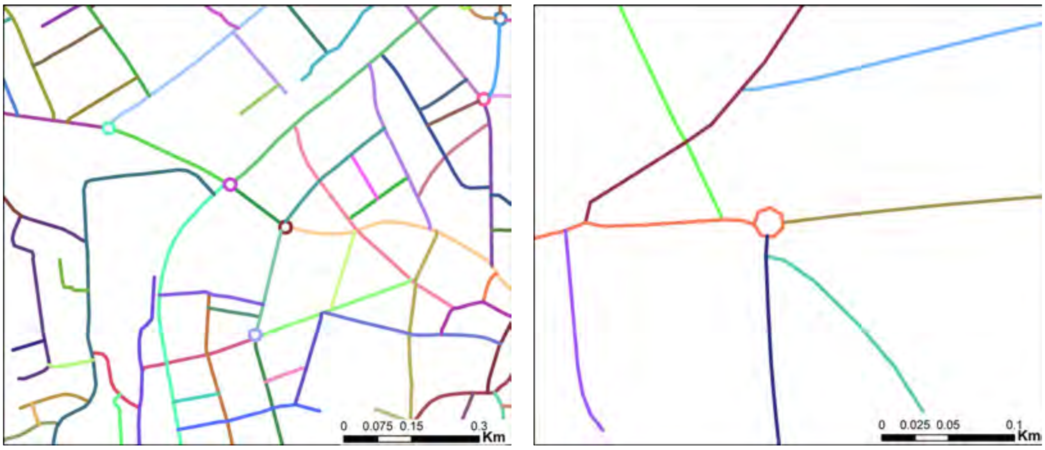
(a) (b)

Figure 4.17: Two snippets from the Zurich test area, highlighting the problems which arise when using a geometric-only stroke approach with a low threshold angle. The segments are colored according to their stroke ID. The wider segments represent 6 m streets and the narrower segments 4 m streets. For illustration purposes, irrelevant strokes are omitted in (b). (© swisstopo)

Roundabouts. Another problem which arises during the stroke building process is the splitting of strokes at roundabouts. Several such situations are depicted in Figure 4.18(a). The different segments are again colored according to the strokes they belong to. It can be seen that the strokes break up at roundabouts and the roundabout form strokes by themselves. While this is an unfortunate result, as we would like to concatenate the opposing strokes together, it is also an inevitable result if only the angle of neighboring segments is taken into account. At a roundabout intersection, the two segments belonging to the roundabout form the smallest angle and are thus concatenated together. This is not always the case, as there exist also situations where a road connects to a roundabout in a very small angle or the data is inaccurate. Such an example is shown in Figure 4.18(b), where the roundabout is concatenated with another stroke. The inclusion of road types would not alleviate this problem however, as the segments in the loop most often belong to the exact same road type as the segments which connect to the roundabout.

The problem is – especially in Switzerland, where roundabouts are quite widespread – not as infrequent as one might think: a quick analysis shows that in the case of the Langenthal test area, over 50 such roundabouts exist, each of which causes 2, sometimes 3 or more strokes to break up. This obviously influences the centrality values of these strokes and ultimately the selection result quite heavily. In the Zurich test area, the situation is even worse, as over 120 roundabouts have been found. As in the first problem, the splitting of strokes at roundabouts also leads to an interruption of important roads and incorrect strokes.

While there exists an attribute in the TLM3D dataset which indicates whether a segment belongs to a roundabout or not, it is not used for two main reasons: 1) this information is still not complete in the TLM3D dataset and sometimes erroneous and 2) the algorithm should also work on other datasets with little adjustments, which would not be the case if it relied on such an attribute.



(a) (b)

Figure 4.18: Splitting of strokes caused by roundabouts: (a) shows several roundabouts where strokes are split up, as the roundabout itself forms a separate stroke and (b) shows a special case where a flat angle of a neighboring segment and a coarse and inaccurate geometry cause the roundabout-stroke to concatenate with the incoming segment, as it forms a smaller angle than the second segment belonging to the roundabout. (© swisstopo)

Differences in Road Density. In the basic approach, the betweenness threshold is set globally. As a direct result, the selection can hardly be fine-tuned: if the pruning in an area features a too dense road network after the initial selection, a higher threshold can be used to counteract this – however, this also affects other areas as well. While this is an expected behavior, the results are more dramatic than in other, segment-based approaches, as the selection is based on entire strokes, which can measure dozens of kilometers in length. Thus, a small threshold increase may lead to the disappearance of a significant portion of the road network.

In areas where the roads are distributed homogeneously, this is not a big problem, as the strokes are pruned uniformly as well. In heterogeneous regions however, where rural regions, small towns and cities coexist, this is not the case at all. Because there exist more roads in cities, the betweenness values of the contained roads are also expected to be higher. Additionally, the problem is intensified by shorter segments and strokes, which results in more nodes in the dual graph relative to rural regions with longer strokes and segments. The manifestation of this effect can be seen in Figure 4.19. The urban area to the east retains a much denser road network than the region outside of the city. While a slightly higher density in cities is desired to some extent, the difference between the urban and rural areas in this example is clearly too high. This problem is also evident in the result of the Lucerne region (see Figure 4.14). Hence, there has to be a means to control the density of such dense regions specifically, without affecting the rural regions. However, the basic centrality approach does not offer any option to locally control such regions in isolation. This leads to violation of constraints 5 to 7 (see Section 3.3.2).

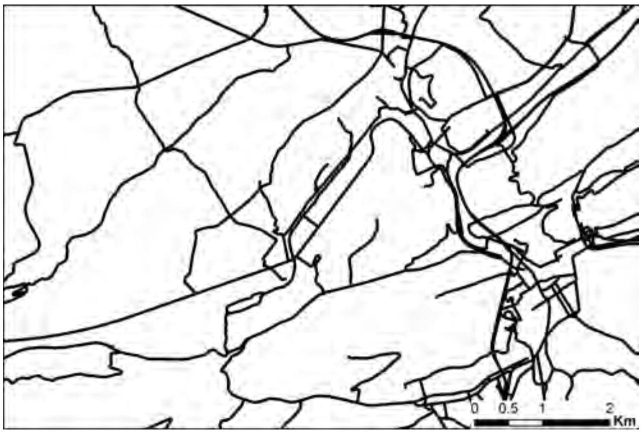


Figure 4.19: Example of the resulting density difference between a city on the right side (Lucerne) and the adjacent rural region. The city retains a very dense road network, while the rural area only shows major roads. (© swisstopo)

Dead-Ends and Disconnected Parts. A problem which becomes evident immediately, is the presence of disconnected parts and, especially, newly created dead-ends in the pruned network (see analysis of constraint 2, 3 and 4 in Section 4.4.2). Completely disconnected parts are not desired at all, as it is a hard requirement from swisstopo that the whole road network has to be connected. Dead-ends are not desired either in a product of this scale and exist, especially in populated areas, only very sparsely and for very specific reasons, for example to provide access to an important infrastructure. Thus, dead-ends are either useless (leading to nowhere important) or very important (leading to a significant site or facility; Touya, 2010). Especially in mountainous areas, they exist to ensure access to lodges or points of interests. As the network is much less dense in mountainous areas, it is possible to also show such access roads. Generally, however, dead-ends are not desired at a scale of 1:200,000 as they deteriorate navigational qualities of the map.

As is evident in the results of Section 4.4.1, dead-ends are very prominent in the pruned network, however. Also wide roads, which often act as the main links in the entire region often end abruptly. The case where dead-ends appear can be grouped into two different groups, which are are also depicted in Figure 4.20. In the first case, the resulting dead-end also exists in the source dataset. This is the case for many unimportant, smaller roads and for various access-roads in the mountains. In the second case, the resulting dead-end did not exist in the source dataset. This means, that the dead-end was created by the algorithm and does not reflect the reality. This behavior of the basic approach stems from the manner the strokes are built: there exist two cases where a new dead-end may be created. In the first case, a stroke is erroneously split up and is selected for the pruned network. If the second stroke is not also selected, this results in a dead-end. In the second case, basically the same happens but for other reasons: even if the stroke is concatenated correctly and ends at a certain intersection, it is possible that the other strokes at that intersection are not selected but removed from the pruned network. This obviously also creates a dead-end. It is important to separate these two cases, however, as it becomes evident that even if the stroke building algorithm generates perfect strokes, this still means that dead-ends may (and most probably will be) created. Thus, it is inherent to the algorithm that it generates new dead-ends, which means that this needs to be handled explicitly in an additional extension.

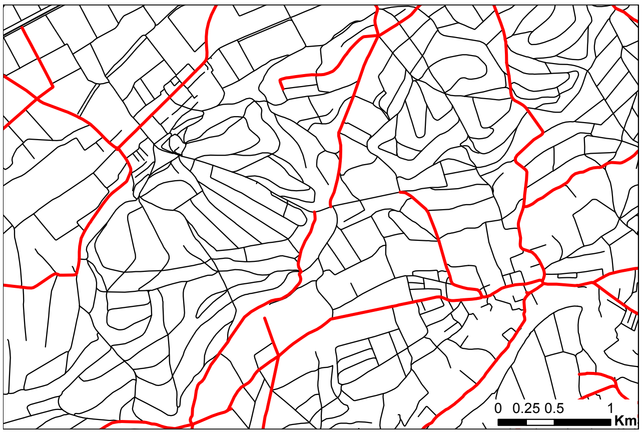


Figure 4.20: Example of newly created dead-ends in the Langenthal test area. The pruned network is shown in red and the source dataset behind in black. It is clear that the dead-ends in this extract should not exist in a final, pruned network. Thus, a method to handle these situations needs to be developed. Note that both dead-ends generated by the algorithm and dead-ends which already existed in the source dataset are present in the extract. (© swisstopo)

5 The Improved Approach

5.1 Overview

The initial results of the basic stroke-based centrality approach, which were presented in Chapter 4, were relatively mixed. While many of the important roads were selected, there exist several problems which deteriorate the effectiveness quite heavily. Some of the aforementioned problems, such as the possibility of a disconnection of the road network as a result of the stroke selection have also been mentioned in earlier work (Jiang and Claramunt, 2004a; Yang et al., 2011). However, methods to counteract these problems have either not been quantitatively evaluated (such as the removal of roundabouts) or have only been partially tackled (e.g. even though disconnection can be avoided, the creation of dead-ends during the stroke selection process has rarely been mentioned).

In this chapter, each of the problems will be analyzed explicitly and possible solutions presented. Figure 5.1 shows the extended approach, which includes completely new or adapted algorithms. Two of the new steps, the removal of roundabouts and the application of an enhanced set of rules for stroke building, seek to improve the strokes and are thus applied during the stroke building process. The other two algorithms, namely the use of an adaptive threshold and the reconnection of dead-ends, are implemented afterward and solve or reduce the problems which are inherent to the stroke approach and centrality based selection.

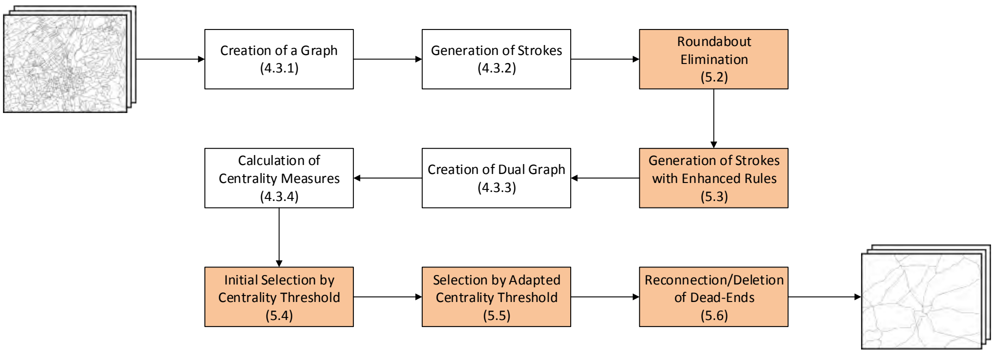


Figure 5.1: Overview of the extended stroke-based centrality approach. The new and adapted steps are highlighted with color.

In the following sections, each step which is highlighted in Figure 5.1 will be described in detail. The other, non-highlighted steps are also present in the improved version, but remain exactly the same as in the basic approach, which is why they are not described a second time.

5.2 Detection and Collapse of Roundabouts

As discussed in Section 4.4.3, roundabouts pose a serious problem. They disrupt strokes, as their continuity is impeded. Hence, an algorithm needs to be employed to either remove the roundabouts or handle the affected strokes separately. Methods which handle this problem have also been proposed in the past. Mackaness and Mackechnie (1999) proposed an approach which is based on cluster analysis and which seeks to detect and simplify junctions. It is a more general approach, however, and it is hardly possible to parametrize the cluster analysis in such a way that only roundabouts are affected. Similarly, Yang et al. (2011) presented an approach which groups strokes across street junctions. Again, the approach is based on cluster analysis and is not trimmed to detect roundabouts. Touya (2010) on the other hand, used a measure of polygon compactness to detect small, round polygons which correspond to roundabouts. The creation of polygons for a large region however, is computationally expensive and seems excessive if only used to detect the roundabouts. As such, a new approach, which is entirely graph- and stroke-based was developed, which is computationally very cheap and used exclusively for the detection and elimination of roundabouts to enhance stroke continuity.

An initial analysis of the roundabouts and the involved strokes shows that there exist three different cases, when the strokes are generated using the purely geometric approach, which are depicted in Figures 4.18(a) and 4.18(b). In the first, most frequent case (Figure 4.18(a)), a stroke forms a roundabout by itself, as explained in Section 4.4.3. In the second case (Figure 4.18(b)), a roundabout is located at the end of a stroke. The third case is similar to the second one, but requires special treatment as explained in a later stage: a stroke belongs to two roundabouts, which are located at both ends of the stroke. Cases two and three should usually not appear in a correct, geometrically accurate dataset. Thus, all the segments which belong to the roundabout are normally completely contained in a single stroke. Theoretically, even a fourth case exists where two segments connect to a loop in such an angle that both get concatenated with two separate segments of the loop. However, this case was only observed for roundabouts in a broader sense (see and Section 5.2.2 and should, as explained, only occur for roundabouts in a strict sense if the data acquisition exhibits very low geometrical accuracy. If the segments of the loop have been created accurately, the angle in between them should always be lower than the ones with the surrounding segments.

Because the roundabouts are completely contained in single strokes, the idea behind the proposed algorithm is to exploit this circumstance and analyze the strokes individually. Before explaining the algorithm in detail, the required steps are outlined as follows:

1. **Roundabout Detection.** Detection of strokes which contain a roundabout. This step also contains the handling of special cases (i.e. two loops belong to one stroke) to increase the resulting detection accuracy.
2. **Roundabout Collapse.** Each detected roundabout is collapsed by generating the centroid of the roundabout, removing the segments which form it and connect the neighboring stroke to the new centroid node.

The detection part of this algorithm was developed together with Stefan A. Benz (see Benz, 2013).

5.2.1 Methods

Roundabout Detection. In the first, and most crucial part, roundabouts need to be detected reliably. Hence, a very simple, yet efficient graph-based algorithm, which relies on strokes, has been developed. Figure 5.4 illustrates this algorithm.

As a first step, a graph G is created for each individual stroke. The strokes have been generated using the simple geometric approach to ensure that loops are completely contained in strokes as explained previously. Next, the dead-ends of the graph are removed. This is not illustrated in detail, but can be done by simply finding a node n in the graph G which only has one adjacent edge e. By repeatedly deleting this node and its adjacent edge, the dead-end is truncated gradually. Nearly all strokes contain no loops at all and exhibit dead-ends at both ends. Hence, by repeatedly removing the nodes and edges of these dead-ends, the graph of the strokes becomes completely empty. These strokes are not considered further in any way as they do not contain a loop. For the ones which do, however, the remaining graph is not empty: if we consider the aforementioned first case (refer to Figure 5.2), where a stroke forms a roundabout by itself, it is clear that there exists no node which only has one adjacent edge, as every node in a loop has at least two adjacent edges. Thus, the graph does not get affected by this procedure and a loop could be extracted from the road network. Loops which belong to the second case are also handled by this method. As only one loop exists at one end of the strokes, the stroke gets truncated gradually from the other end until only the loop remains.

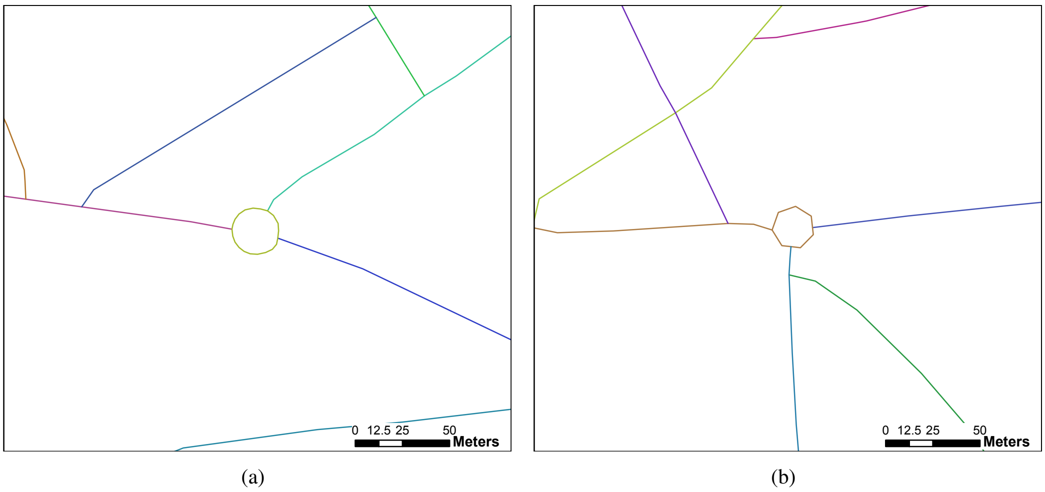


Figure 5.2: (a) Case 1: Resulting strokes using a purely geometric approach at a roundabout with sufficient geometrical accuracy (b) Case 2: Resulting strokes at loops with low geometrical accuracy. (© swisstopo)

The third case however, is not considered by this method. Because loops exist at both ends, there is no node which only has one adjacent edge. This is why the dual loop detection algorithm has also been implemented. The algorithm iterates over the edges of the previously constructed graph of the stroke, if it is not empty after the deletion of dead-ends. If an edge e with length above 100 m is found, the graph G is split up into subgraphs G_1 and G_2 at one of the nodes of e. The dead-ends of G_1 and G_2 are then again removed. The length of 100 m has been selected empirically. It translates to the maximum length of a segment inside of a roundabout. If a longer segment is found, it is considered as a segment which lies outside of a roundabout, which is why graph G is split at exactly this position. If the handled stroke indeed belongs to the third case, the stroke is split up in between the two loops which transforms one case 3 stroke into two case 2 strokes, eliminating the problem of dual loops.

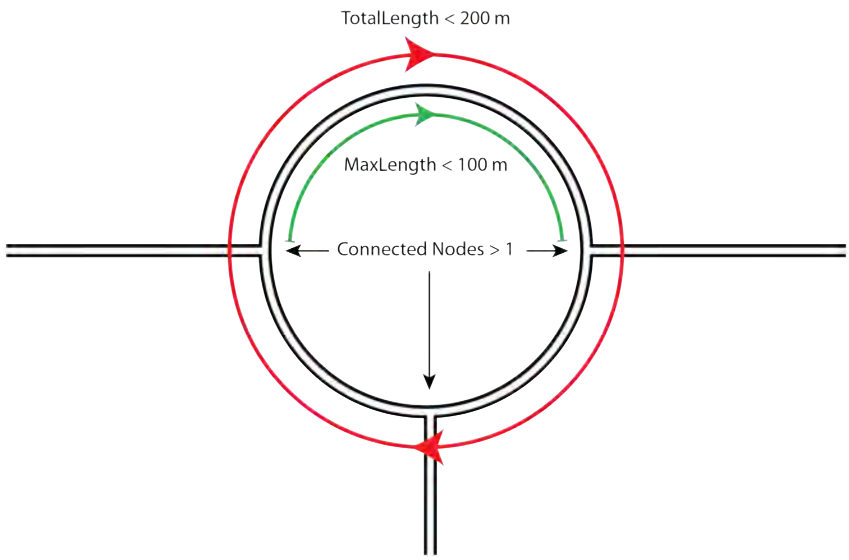


Figure 5.3: The three parameters used to define a loop as a roundabout. The total length of the segments which form the roundabout may not exceed 200 m, the maximum length of a single segment inside a roundabout may not exceed 100 m and there have to be at least two nodes which are connected to three edges in the *original graph* of the road network. This ensures that the loop does not simply consist of a dead-end loop, which loops back to the same node.

In the last step, the detected loops are filtered, as not all of them are roundabouts. A stroke can also form loops by concatenating segments in such a way that it intersects itself. We are not interested in those loops, however. Again, the selected parameters, which are depicted in Figure 5.3, have been selected empirically and they could obviously differ in other countries. However, they proved to be very effective in the all of the four tested areas. Three parameters were chosen to sufficiently characterize a roundabout:

- **TotalLength:** The total length of the segments which form the roundabout may not exceed 200 m.
- **MaxLength:** The maximum length of a single segment inside a roundabout may not exceed 100 m.
- **Connected Nodes:** There have to be at least two nodes which are connected to three edges in the *original graph* of the road network. This ensures that the loop does not simply consist of a dead-end loop, which loops back to the same node.

If a loop fulfills all of these requirements, it is considered to be a roundabout and is added to a list to be used for the next step, which intends to collapse the roundabouts entirely.

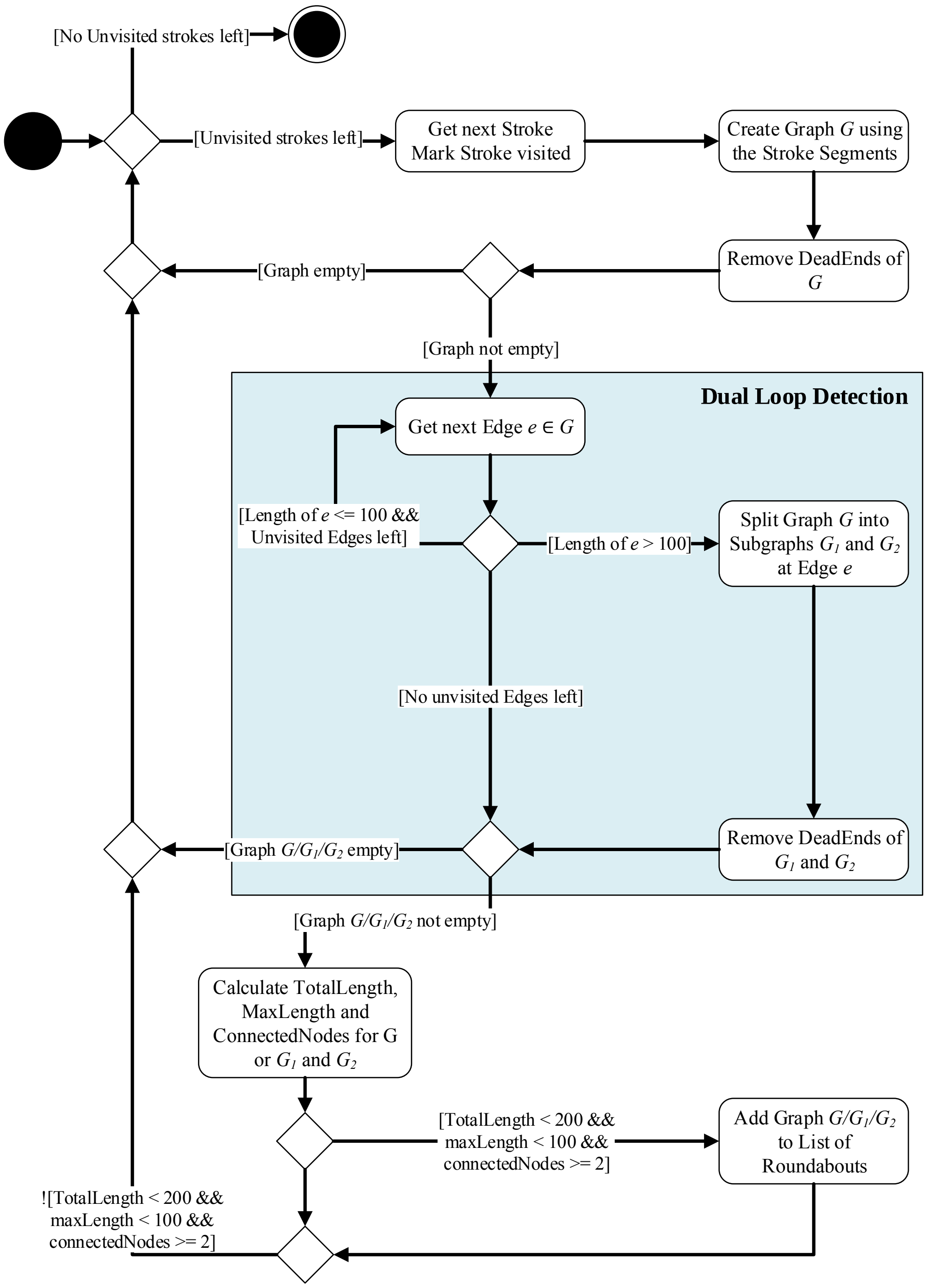


Figure 5.4: UML activity diagram of the roundabout detection algorithm. The highlighted dual loop detection part is optional. However, it enables the algorithm to also detect strokes which contain loops at both ends. The actual algorithm however, reduces itself to a simple loop detection inside of every stroke and a following characterization of these loops.

Collapse of Roundabouts. After all roundabouts have been detected, they need to be collapsed. A UML activity diagram of this step is depicted in Figure 5.5. In this part, the algorithm iterates through all previously detected and gathered roundabouts. For each adjacent edge e of a node in a roundabout, it is checked whether it is part of the roundabout or whether the edge is simply connected to it. If e is not part of the roundabout but connected to it, e is added to the list of incident edges iE. Afterward, a new node c is created at the centroid coordinates of the roundabout. The edges contained in iE are then expanded such that they connect to c. The last step consists of the removal of every segment and the related nodes which form the roundabout. This results in the complete removal of the roundabout and the extension of the adjacent segments to the centroid of the roundabout. Hence, segments, which were previously separated by a roundabout now exhibit a direct relationship as they are connected to a common node.

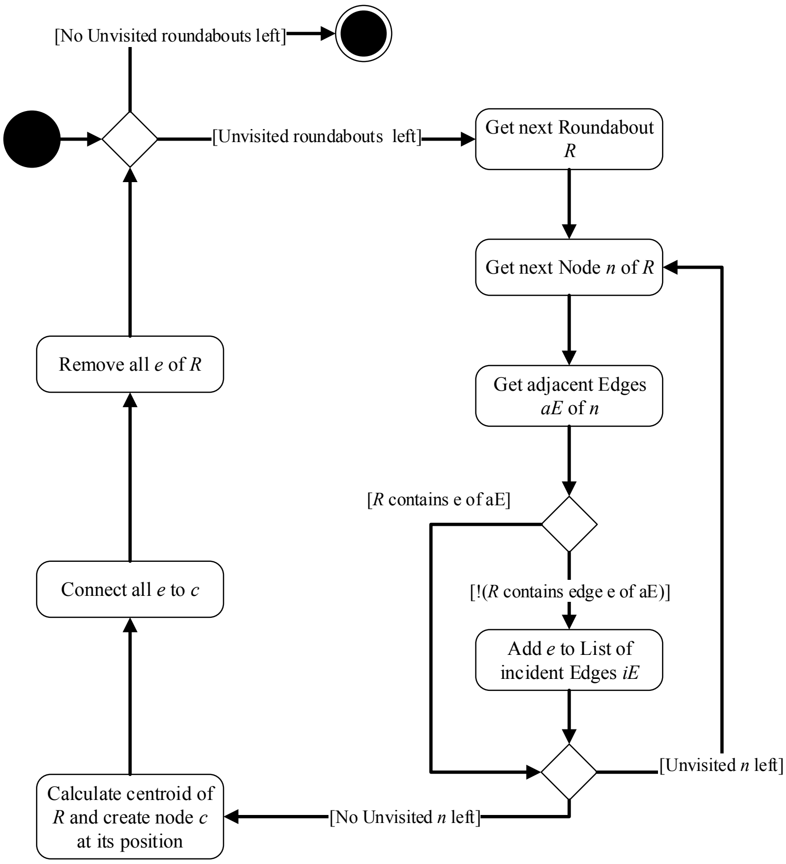


Figure 5.5: UML activity diagram of the roundabout collapse algorithm. The algorithm calculates the centroid node of the roundabout and extends the adjacent segments to its position. Afterward, all segments which were part of the roundabout are completely removed.

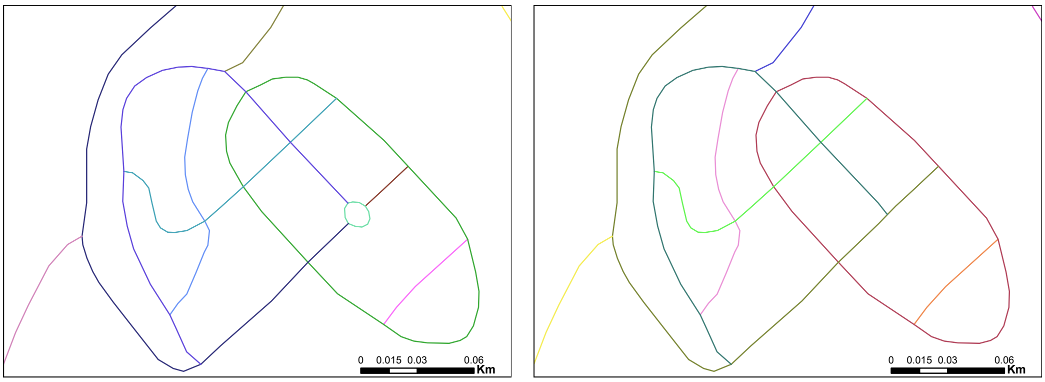
5.2.2 Results and Discussion

The number of roundabouts detected in each of the test areas, as well as the difference in the numbers of strokes generated can be seen in Table 5.1. Visual inspection of the results and the corresponding test areas shows that all roundabouts have been detected and collapsed correctly, with very few exceptions. Also, special attention was given that the algorithm does not collapse other loops which are not considered as roundabouts (i.e. false positives). In the following sections, each test area will be analyzed shortly by comparing the strokes of detailed extracts of the regions, generated before and after the elimination of roundabouts.

Table 5.1: Number of loops detected and collapsed in each test area. In addition, the table shows the numbers of strokes generated before and after the roundabout collapse algorithm. The reduction in strokes can be explained by the removal of the roundabouts itself (around one stroke per roundabout in average) and the increased continuity caused by the collapse.

Test Area	#Loops Collapsed	#Strokes before	#Strokes after
Davos	1	2843	2841
Lucerne	57	8470	8330
Langenthal	29	9890	9816
Zurich	134	19658	19329

Davos. In the Davos test area, only one loop has been detected (which is why an overview of the whole area is omitted in this section). It is the only roundabout that was found in the area. A very low number of roundabouts was to be expected in a very mountainous area. As can be seen in Figures 5.6(a) and 5.6(b), the roundabout has been collapsed successfully.



(a) (b)

Figure 5.6: (a) Strokes generated before the elimination of roundabouts in the Davos area and (b) Strokes generated after the elimination of roundabouts. The roundabout has been collapsed and the strokes concatenated successfully. (© swisstopo)

Lucerne. Figure 5.7 shows an overview of the Lucerne area where all detected roundabouts are marked as red dots. The vast majority of loops are located inside the City of Lucerne. The surrounding towns on the other hand, contain nearly no roundabouts. Visual inspection of the area shows that all roundabouts have been detected and collapsed correctly. Figures 5.8(a) and 5.8(b) show a detailed view with a smaller and a larger roundabout before and after the elimination. The strokes at the roundabouts have been concatenated with success and the continuity could be improved significantly.

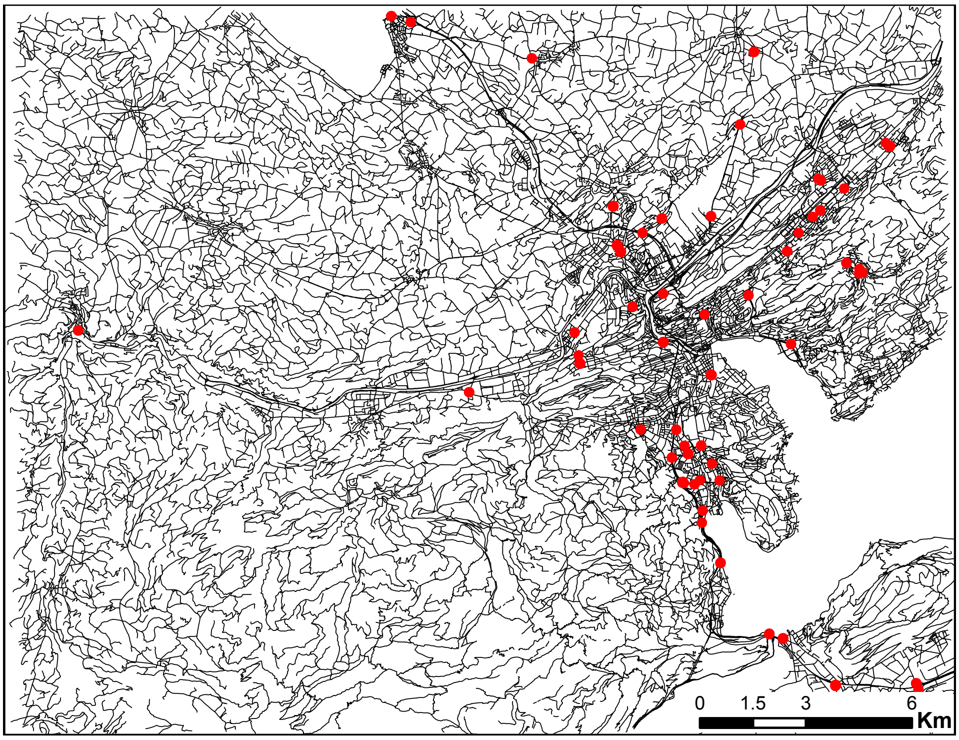
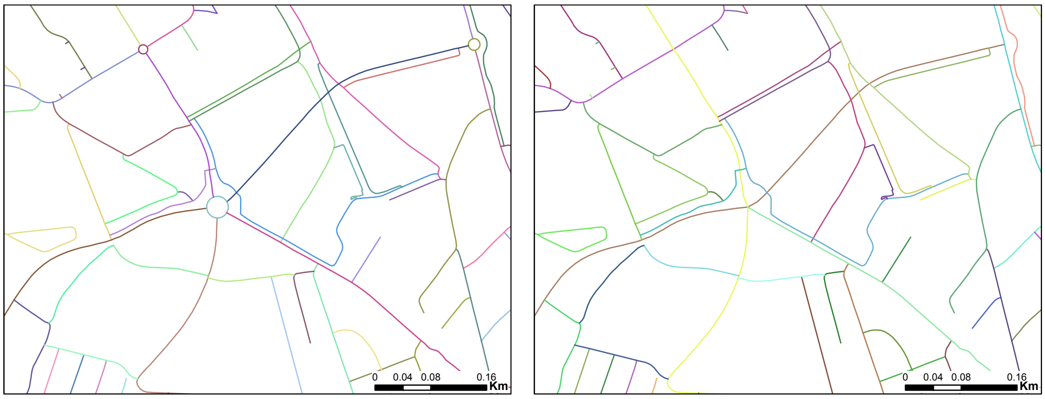


Figure 5.7: Overview of the Lucerne area with all roundabouts marked as red dots. (© swisstopo)



(a)

(b)

Figure 5.8: (a) Sample strokes generated in the Lucerne area before the collapse of roundabouts are split up at both pictured roundabouts. (b) The smaller roundabout, as well as the larger one in the center were collapsed correctly and the continuity of the strokes could be improved. (© swisstopo)

Langenthal. As before, Figure 5.9 shows an overview of the detected roundabouts in the Langenthal area. The results are quite different from the Lucerne area: while a large cluster of roundabouts can be seen in the City of Langenthal, the surrounding towns also contain quite a few roundabouts. Figure 5.10 shows a more detailed view of the city before and after the elimination of roundabouts. In Figure 5.10(a), the strokes are split up at each roundabout, as has been shown in the description of the problem in Section 4.4.3. Figure 5.10(b), where the strokes have been generated again after the elimination, shows a different picture, however. The continuity of the strokes is maintained much better and opposing strokes have been concatenated successfully. The collapse is not perfect, however, as can be seen at the roundabout at the lower left, where the segments which were extended to the centroid of the roundabout form a relatively sharp angle. This results in the wrong segments being concatenated.

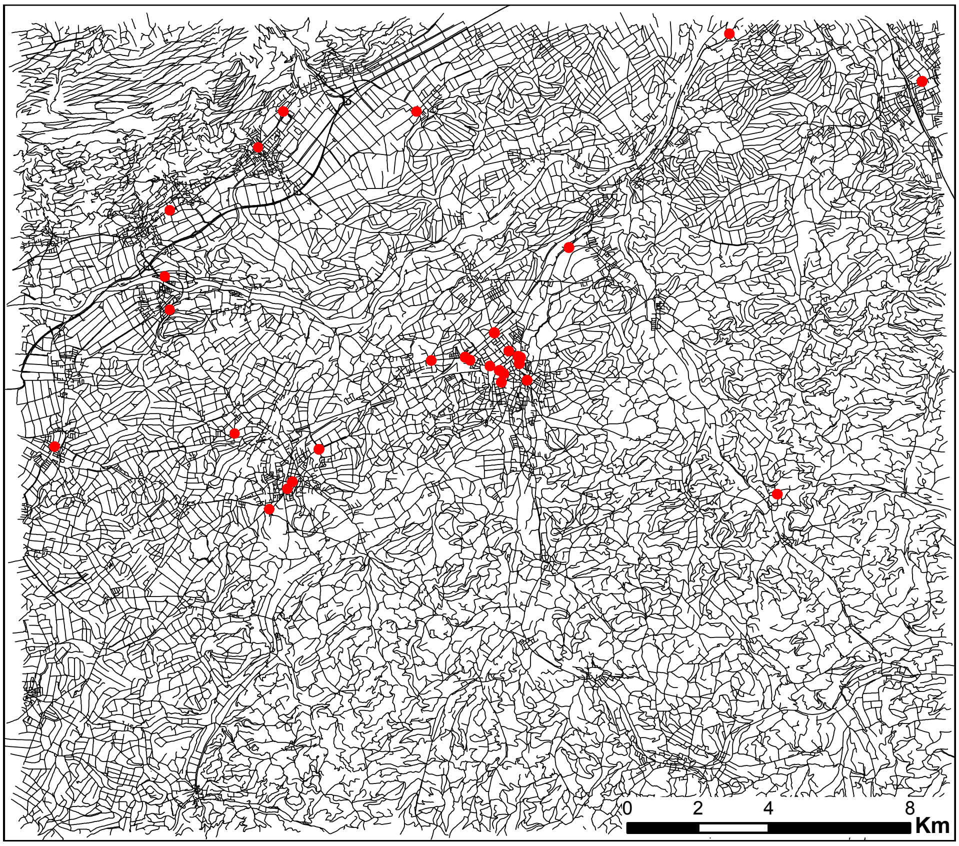
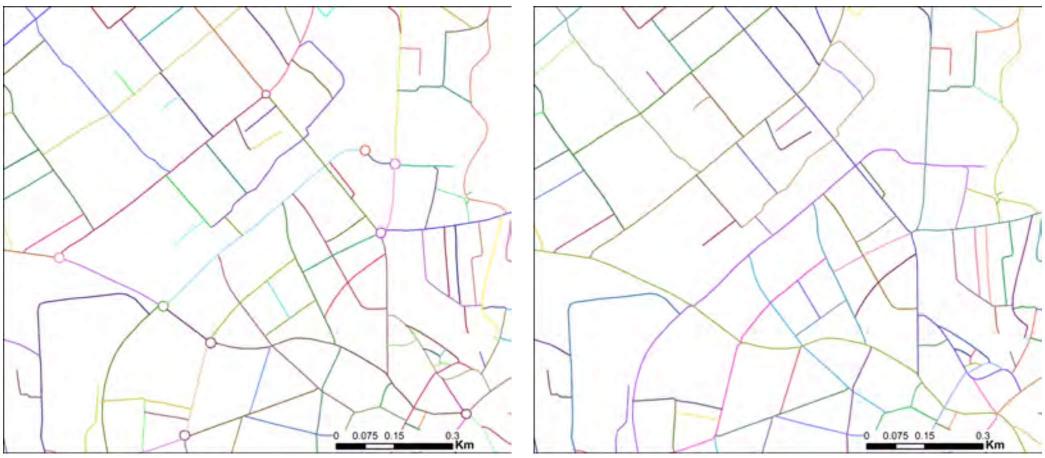


Figure 5.9: Overview of the Langenthal area with all roundabouts marked as red dots. (© swisstopo)



(a) (b)

Figure 5.10: (a) Sample strokes generated in the Langenthal area before the collapse of roundabouts are split up very frequently, at each roundabout. (b) Strokes, which are generated after the elimination of roundabouts are longer and the continuity is kept. (© swisstopo)

Zurich. Again, an overview of the region is shown in Figure 5.11. The distribution of the roundabouts is quite different than in the other, more rural areas. The reason for this is probably the existence of larger towns and cities throughout the area, which make frequent use of roundabouts to increase the traffic flow. The City of Zurich itself, however, shows a remarkable absence of roundabouts. While this may be surprising at first glance, closer visual inspection shows that there are indeed no roundabouts in the city itself. While the algorithm worked again very well and an examination of the results also confirmed that all roundabouts have been detected and collapsed with success, Figure 5.12(a) shows a limitation of this approach. What appear to be roundabouts at first glance, are squares which resemble a roundabout quite closely but are not classified as such. While the geometrical accuracy of the shown structures is relatively low compared to other examples, streets enter such structures in a flat angle very frequently. As such, the square is placed into two separate strokes and the algorithm fails to detect them. Three such cases could be observed in the City of Zurich.

Figures 5.12(b) and 5.12(c) on the other hand, show again two examples where the algorithm worked very well and where the roundabouts have been collapsed and the strokes concatenated with success.

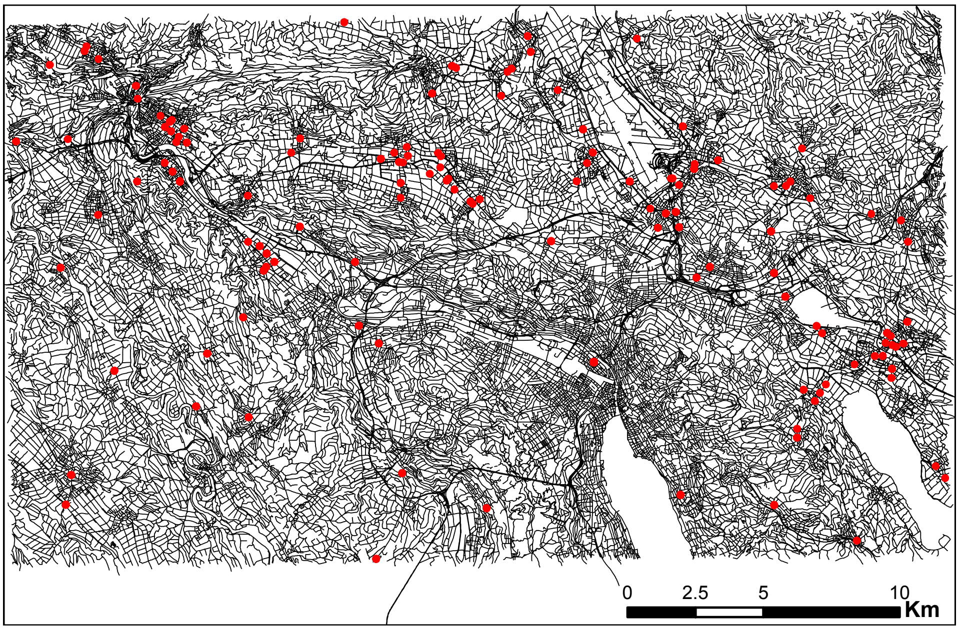
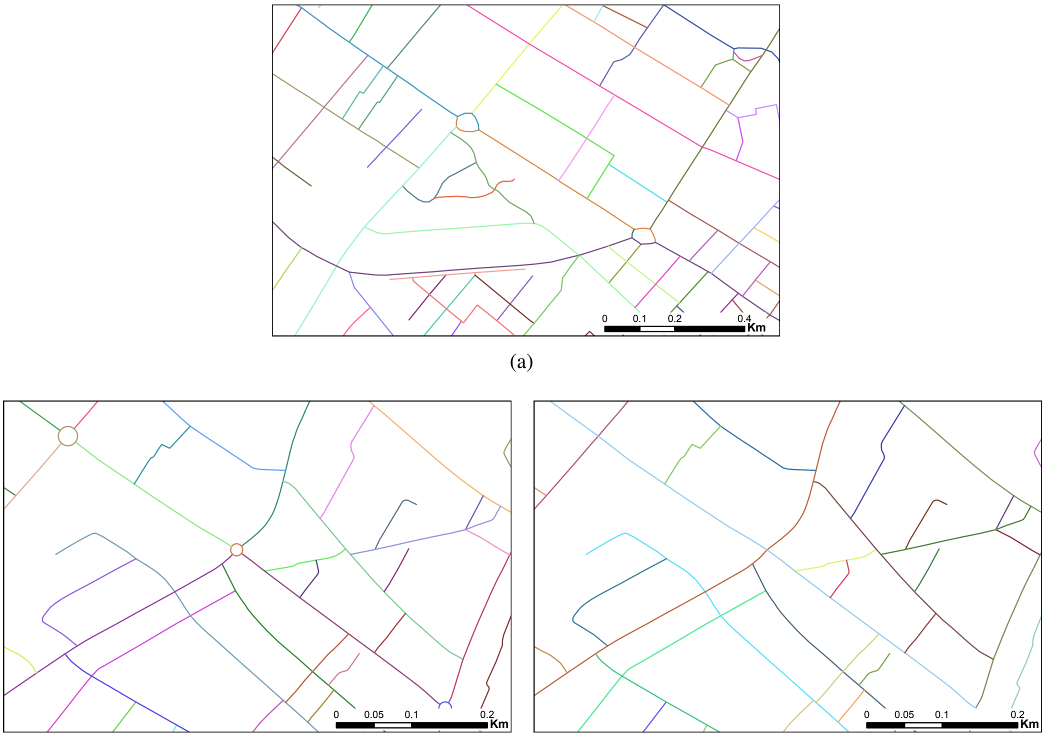


Figure 5.11: Overview of the Zurich area with all roundabouts marked as red dots. (© swisstopo)



(a)

(b) (c)

Figure 5.12: (a) In the Zurich test area, the algorithm fails to detect squares which do not form a circle, as the incoming streets form a very flat angle to the segments contained in the square. As such, it is possible that the square itself is not completely contained in a single stroke: a requirement which is necessary for the algorithm to work. (b) Shows an extract before and (c) after the elimination of roundabouts. They have been collapsed successfully and the strokes have been concatenated correctly. (© swisstopo)

The presented method, which seeks to detect and eliminate roundabouts from a road dataset, proved to be working very well. Roundabouts in a strict sense, which form a relatively accurate circle, have been detected with very high accuracy, i.e. all of them have been detected, as a visual inspection revealed. An important trait of the algorithm is also that it does not detect spurious roundabouts, hence every detection also translated to a roundabout. One problem which is based on the fact that the algorithm relies on roundabouts being completely contained in single strokes are the detection of roundabouts in a wider sense, i.e. squares or more oval structures. Two such cases have been shown in Figure 5.12(a), one more exists in the Zurich dataset and two similar examples also exist in the Lucerne dataset. However, such structures also exhibit different characteristics than roundabouts and would require a different approach than the one presented. In general, however, this is a relatively minor problem, as such situations exist very seldom compared to the total number of roundabouts detected (see Table 5.1).

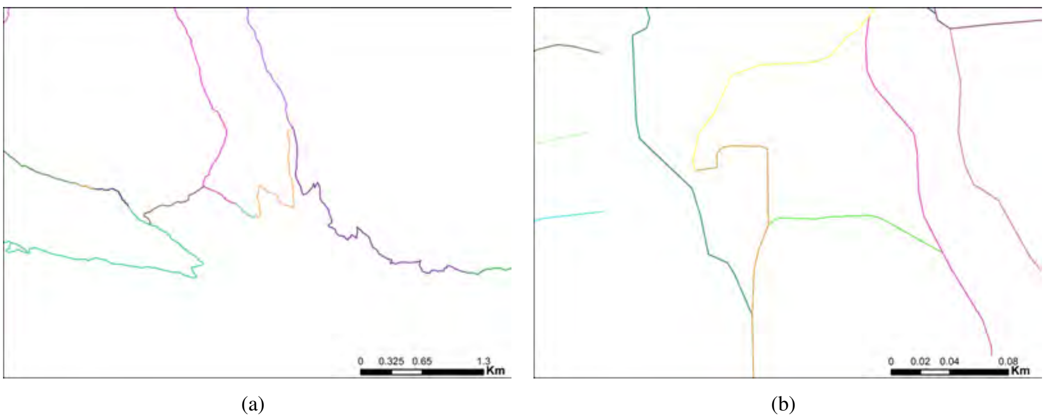
The collapse step of the algorithm works very well in most situations too. The solution sometimes does not look aesthetic, but this is also not the objective of this thesis, as it focuses mainly on selection and at the target scale of 1:200,000, where such details are no longer perceivable. Thus, it is important that it is functionally adequate. While this is mostly the case, spiky angles can sometimes lead to segments not getting concatenated, depending on the chosen angle threshold. This is, however, only a minor problem, as can be seen in the adaptation of the algorithm presented in Section 5.3.

In general, the algorithm worked very well. Besides some exceptions, roundabouts have been successfully eliminated from the dataset and the continuity of strokes could be improved. With slight modifications to the parameters, this algorithm should work on every road dataset which exhibits a sufficient geometric accuracy.

5.3 Enhanced Semantics for Stroke Generation

Two separate problems, also highlighted in Section 4.4.3, have been identified, arising as a result of a purely geometric stroke generation. The first one occurs during the concatenation of the strokes and has also been investigated by other authors, such as Zhou and Li (2012). They analyzed different concatenation strategies and already came to the conclusion that the *every-best-fit* strategy, which has been used in this work, produces the best results. Zhou and Li (2012) also suggested to include thematic attributes, such as the road class, but the authors reported that while this improved the accuracy of the strokes, the difference to a purely geometric approach was not significant. In addition, one has to consider that the TLM3D dataset stores the road class very accurately: the width often changes several times over short distances, for example at an intersection, to allow more lanes next to each other. Hence, a simple rule where segments of equal road classes get concatenated is not enough and would lead to many strokes being split up erroneously.

The second problem lies in the selection of the threshold angle. While many authors suggested a threshold angle between 40° and 60° to maintain the principle of good continuity (Thomson and Richardson, 1999; Jiang, 2009; Yang et al., 2011; Zhou and Li, 2012), an inspection of the data revealed that this might not be enough. There exist situations where roads make a sharp turn, as depicted in Figure 5.13. In Figure 5.13(a), an extract from the Davos dataset is shown where many stroke interruptions are present. The road classes at these positions change and the angle is above 60°, which results in this picture. A different situation is shown in Figure 5.13(b) in the Lucerne dataset, in a more urban area. A relatively wide road makes very sharp turns and changes its width at the same time resulting in an interruption.



(a) (b)

Figure 5.13: (a) Problematic stroke concatenation in the Davos area: strokes are split up very frequently because of very high angles during road type changes (b) Another example in the Lucerne region, where strokes are split up. (© swisstopo)

As such, it is necessary to adapt the algorithm which builds the strokes to take into account road classes, while allowing different classes to join together, as proposed by Thomson (2006). At the same time, the angle threshold is increased to allow strokes to continue even when sharp turns occur.

5.3.1 Methods

Custom Road Classes. As a first step, the very detailed road classes defined in the TLM3D dataset were assigned to different groups. An overview of the new classes is shown in Figure 5.14. This step is not only necessary to reduce the complexity of the algorithm, but also to make an ordering of the classes possible, as the integer coded classes in the dataset do not follow a strict hierarchy, as is also shown in the diagram. With the exception of group 0, which is solely occupied by the connection road type and plays a special role in the dataset, a road hierarchy from group 1 to 5 is now visible, with 1 being the most important and 5 the least important group. This is necessary to make an efficient algorithm possible, as the integer values can be compared directly. The different groups can be described as follows:

- **Connection.** Owed to the structure of the data set (see Section 3.1), connections belong to a separate group, as they are an important part of the TLM to ensure connectivity.
- **Highways.** Everything related to any form of limited-access, high-speed road. This includes entries and exits.
- **Main roads.** Very broad roads up to over 10 m. The lower border is drawn at 6.21 m.
- **Minor roads.** Narrower roads from 2.81 - 6.20 m are depicted as minor roads. Drivable for most vehicles.
- **Narrow roads.** Very narrow paths and roads from 1.81 - 2.80 m are in this group. While not drivable for larger vehicles, they can still play an important role in the development of rural areas or to provide alternate routes to the main roads.
- **Footpaths.** Paths having a width of under 1.81 m and marked paths.

Enhanced Comparator. In the basic stroke approach discussed in Section 4.3.2, a simple comparator was used to sort pairs of segments to calculate the best match. The angle which the segment pairs form was used as the only criterion. In the improved approach, however, a more complex comparator has been developed which makes use of the custom road classes. The comparator will be described in textual form as it would not easily fit in the UML activity diagram of the stroke building algorithm, which is shown in Figure 5.15. The general algorithm of the stroke building process is not changed, however.

In the following, P_1 describes the first pair of segments and P_2 the second pair to be compared. $C_L(P)$ denotes the lower custom road type of a pair, $C_H(P)$ the higher custom road type. Similarly, $E_L(P)$ is the exact lower road type and $E_H(P)$ the higher exact road type, as depicted in Figure 5.14. $\Delta C(P)$ is the custom road type difference of the two segments which are contained in a pair, e.g. if one segments belongs to type 1 (highway) and the other one to type 4 (path), the $\Delta C(P)$ of this pair is 3.

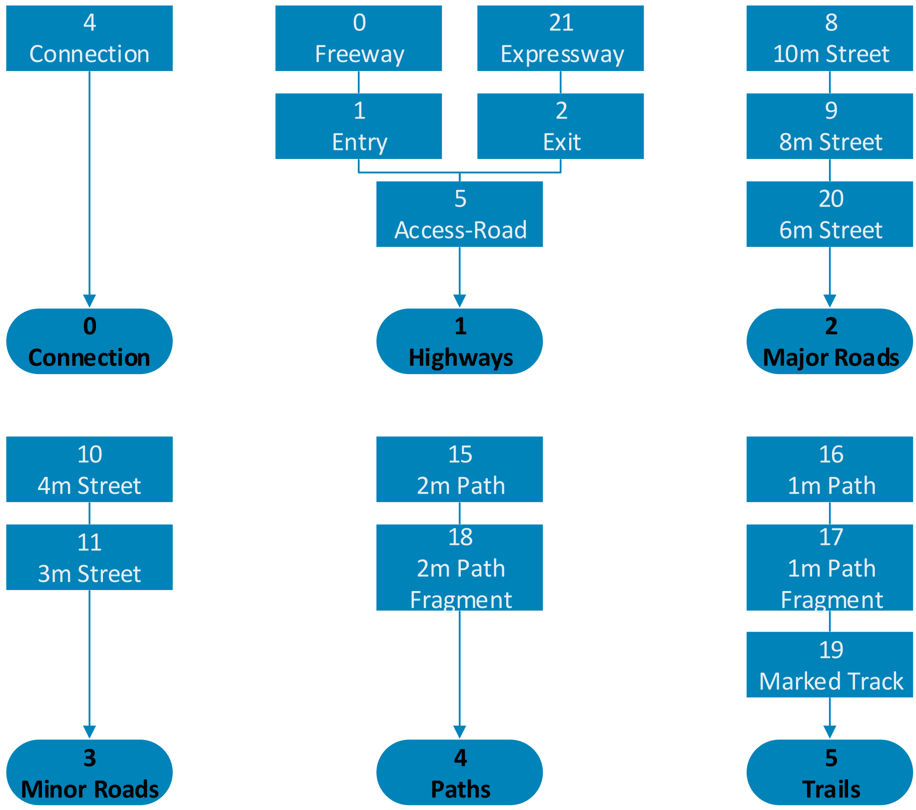


Figure 5.14: Overview of the custom road classes, which group several of the road types together. The white types are the ones defined in the TLM3D dataset and the black ones are the custom grouped types. The numbers show the actual integer coding of the road types in the dataset. With the new groups, a strict hierarchy has been generated to allow the types to be compared effectively in an algorithm. Group 0 forms an exception, as a connection segment can potentially be placed in between every other type.

The comparator is using a hierarchical approach to determine whether a pair is more relevant than another. The different steps which are checked in sequence are listed below:

- **Highways.** First, the algorithm checks whether one of the pairs contains a segment which belongs to the highway custom object type: if $C(P_1) = 1$ and $C(P_2) \neq 1$, P_1 is preferred, otherwise P_2. If neither of the pairs contains a highway segment, the comparator continues.
- **Connections.** Next, it checks whether one of the pairs contains a segment of the connection object type: if $C(P_1) = 0$ and $C(P_2) \neq 0$, P_2 is preferred, otherwise P_1. Hence, pairs which *do not* contain a connection segment are preferred.
- **Road Type Difference.** If both of the pairs fulfill the requirement $C_L(P) \neq 2$ and $C_H(P) \neq 3$, i.e. if the lower road type of both pairs is not 2 (major roads) and the higher road type is not 3 (minor roads), then $\Delta C(P)$ of the pairs is compared. If $\Delta C(P_1) = 0$ (i.e. both segments of P_1 belong to the same class) and $\Delta C(P_2) \neq 0$, then P_1 is preferred, if the opposite is true P_2. If both pairs exhibit a type difference of 0, then the comparator continues with the next step. The reasoning behind this is to allow only segments of the same road class (i.e. classes 2 to 5)

to concatenate. However, classes 2 and 3 (major and minor roads) should also be allowed to be connected, which is why this step is skipped if they occur in both pairs.

- **Custom Road Type.** At this point, either both pairs contain classes 2 and 3 or both pairs exhibit a type difference of 0 and the comparator could not yet decide which pair is more important. Hence, $C_L(P)$ itself is compared: if $C_L(P_1) < C_L(P_2)$, then P_1 is preferred, otherwise P_2. Thus, lower (more important) classes are preferred.
- **Exact Road Type.** As one of the last resorts, the comparator looks at the *exact* type which was defined in the TLM3D dataset (see Figure 5.14. If $E_L(P_1) = E_H(P_1)$ and $E_L(P_2) \neq E_H(P_2)$, i.e. the exact road types of the segments of only one pair are equal, then this pair is preferred.
- **Angle.** Now, as the very last step, the angle of the pairs comes into play. If $\measuredangle P_1 < \measuredangle P_2$, then P_1 is preferred, otherwise P_2.

In fact, two different comparators were used: a general purpose comparator (as presented above) and one which has been trimmed slightly for mountainous regions. The trimmed version only received slight modifications. Essentially, it allows $\Delta C(P)$ to be 1 for *every* road class to allow paths to be connected to minor roads or trails to paths. This has been done to adapt for the different road network structure in mountainous regions, where roads are generally narrower and where paths are often quite important and used as minor roads. In addition, segments were concatenated without checking any rules if no other option existed (i.e. if only one candidate pair exists in the first place).

Enhanced Comparator Thresholds. At this point, the comparator has sorted the pairs according to their importance. The order determines which pair of segments are concatenated first. As depicted in Figure 5.15, the algorithm then selects the first pair and checks whether it fulfills the thresholds or not. While the threshold in the basic algorithm solely consisted of the angle of the pair, it is again slighlty more complex in the enhanced version. The algorithm checks several requirements sequentially and either concatenates the segments contained in the pair or selects the next pair in the ordered list. The segments are concatenated if *any* of the following requirements are met:

1. The pair contains a connection segment.
2. The segments are of the same TLM3D type and the angle is below 90°.
3. The pair contains a highway segment, the object type difference is 0 or the pair contains a minor and a major road segment *and* the angle is below 60°.

If none of these requirements are met, the segments will not be concatenated. The reasoning for the different requirements can be explained as follows: (1) If the pair contains a connection segment, the concatenation is allowed. Because this road type is quite special and often features very sharp angles (which are not present in reality), it is simply allowed without any further requirements. It has to be noted however, that this is actually the very last option as pairs which contain connection segments have been placed to the last position in the comparator. Hence, if there exist no other viable options, the concatenation is allowed to preserve continuity across such road types. (2) If the segments are of the exact same type, the maximum angle is set to 90° in order to make sharp turns, for example in urban areas, possible. The angle requirement is made more stringent for (3), however. It is unlikely that a road makes a very sharp turn and changes its type (but remains in the same custom type group) at the same time. If this is the case, then it is likely that the two segments do not belong to the same road. At the same time this also means that if the object type difference of a pair exceeds 0, i.e. the segments are not part of the same custom type group, then the concatenation will not be allowed, no matter how small the angle is. An exception is made for the major and minor road type, which are allowed to connect together. Hence, if the pair contains a segment which belongs to the minor road type and a segment which belongs to the major road type, the connection is allowed if the angle is below 60°.

If summarized, the algorithm concatenates only segments which are of the same custom type and exhibit an angle of below 60°, with the exception of minor and major roads. If the segments belong to exactly the same road type however, the angle requirement is relaxed by increasing the threshold to 90°.

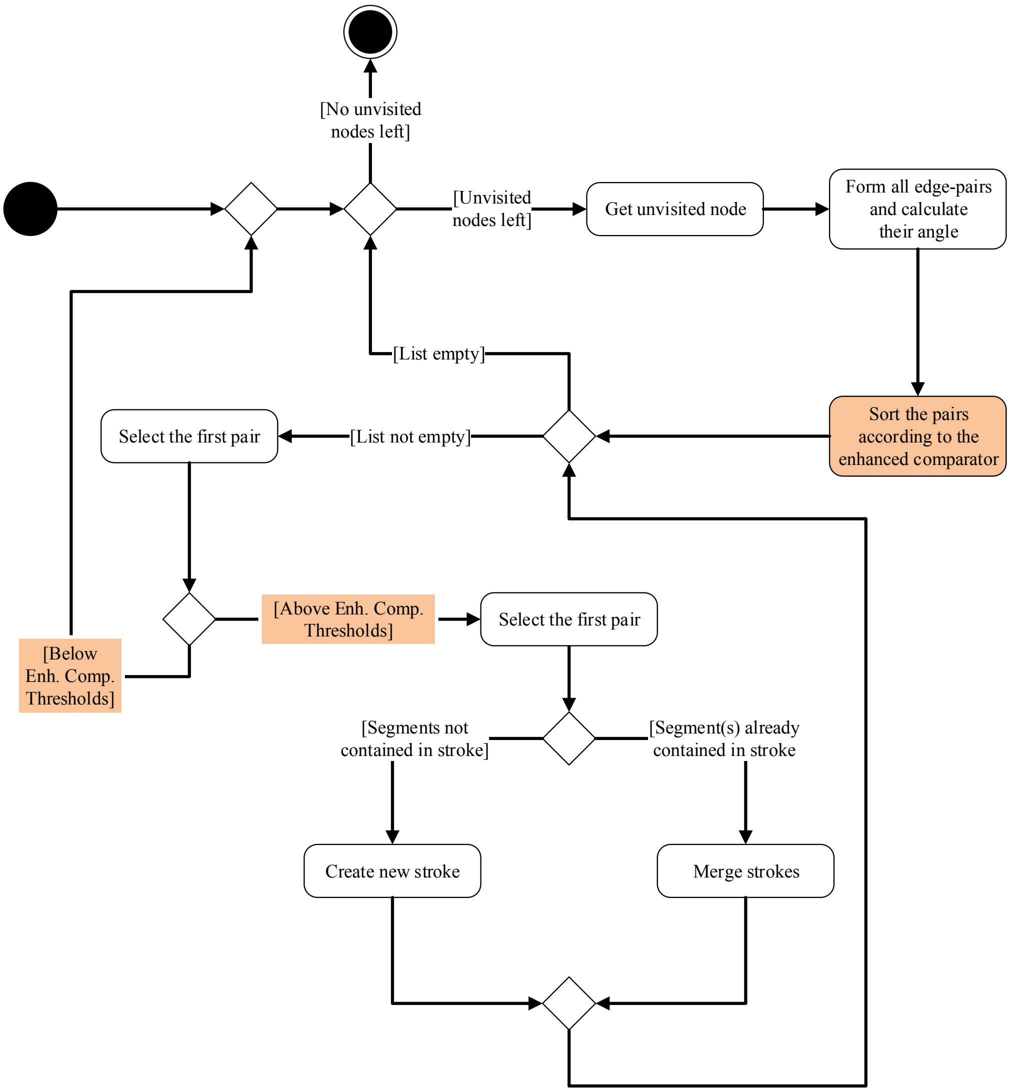


Figure 5.15: UML activity diagram of the improved stroke building algorithm. The general algorithm remains the same as in the basic approach (see Section 4.3.2). However, the comparator and the thresholds used have been changed, which are both highlighted in the diagram.

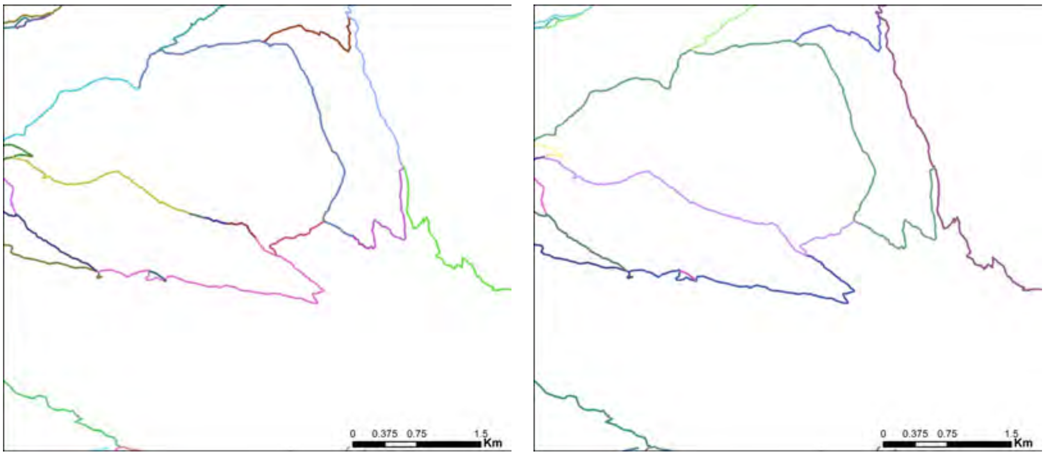
5.3.2 Results and Discussion

Table 5.2 shows the numbers of strokes created with the purely geometric approach and with the enhanced version. Except for the Davos dataset, which shows slightly less strokes with the enhanced version, the total number of strokes generally increased. This might look strange at first glance, as the angle threshold has been increased in certain situations. However, one has to remember that with the inclusion of the road type and the resulting limitation that only segments which belong to the same road group are concatenated together, there exist more constraints. This results in many splits between strokes, even though they are in compliance with the principle of good continuation. Such splits can be considered as desirable as we do not want a trail to be part of a minor road, because this results in complications during the selection phase of the algorithm (as strokes are always selected in their entirety). The different result for Davos can also be explained by the fact that it uses a different comparator which allows different groups of road types to concatenate together (in the general purpose comparator, this is only allowed between major and minor roads). Thus, the reasoning behind the comparator is again highlighted: maintain continuity as long as possible where it makes sense (as long as the segments are of the same group) but do not hesitate to ignore the continuity principle if the road class difference between the segments is too high.

Table 5.2: Number of strokes generated with the basic algorithm and with the enhanced version using different thresholds and an enhanced comparator.

Test Area	**#Strokes Basic**	**#Strokes Enhanced**	**#Strokes enhanced (without loops)**
Davos	2843	2793	2791
Lucerne	8470	9662	9524
Langenthal	9890	11013	10936
Zurich	19658	21961	21648

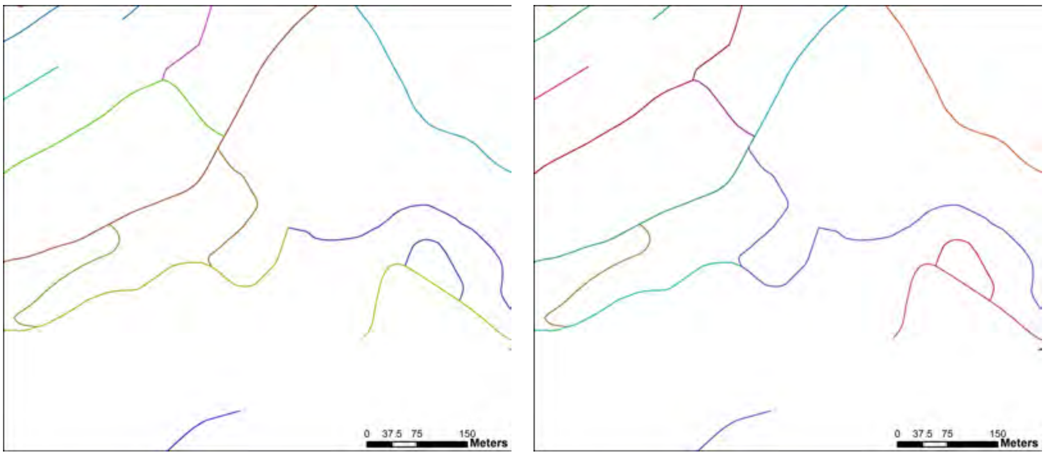
Figure 5.16 shows a snippet of the Davos area, depicting the strokes generated using the basic generator on the left and the ones created using the improved version on the right. As can be clearly seen, the continuity of the strokes could be preserved in this case, as the improved version uses a higher angle threshold in addition to the road classes. The basic version, on the other hand, simply used an angle threshold of 60° which resulted in many unnecessary disruptions.



(a) (b)

Figure 5.16: (a) The purely geometric stroke generator creates strokes in the Davos area which are split up very frequently, as the angle between segments exceeds 60°. (b) The improved version is able to preserve the continuity as the stroke generator uses the road class in addition to a higher angle threshold. (© swisstopo)

Figure 5.17 shows a case in the Lucerne test area where a relatively long stroke is split up at a very sharp angle. It is clear that the stroke should continue, however, as the road class at that position of the two segments is exactly the same. In the improved version, the stroke is correctly concatenated. In addition, it can be seen that a different concatenation is chosen at the intersection in the center of the figure. In the improved version, a concatenation with a higher angle is chosen – however, the road class of the two segments is exactly the same (a 2 m wide path). In the basic version, a 2 m wide path and a 1 m trail are concatenated, which is clearly the worse decision.



(a) (b)

Figure 5.17: (a) The basic stroke generator splits up a stroke in the Lucerne area at a very sharp angle, even though the two segments clearly belong to the same road. (b) The improved version is able to connect the two affected segments correctly, as the angle threshold is increased. In addition, a different concatenation is chosen at the intersection close to the center of the map, as the road class is taken into account. (© swisstopo)

As a last example, Figure 5.18 depicts the opposite effect the enhanced stroke comparator has on the resulting strokes, which is also reflected in the higher number of strokes generated (see Table 5.2). In certain circumstances, the improved version splits up more strokes than the basic stroke generator. This is due to the fact that more restrictions exist: while the enhanced version uses a higher angle threshold than the basic version in certain cases, it also introduces additional thresholds (e.g. the road class must belong to same group), which do not exist in the basic version. Hence, it is clear that more strokes are generated where these constraints come into effect. Such a result can be seen in the long, orange stroke on the left, which is split up twice on the right. While this might be undesirable at first glance, one has to remember that using a stroke-based selection, always complete strokes are selected. A closer examination of the aforementioned stroke reveals that it changes its road class several times. To be more precise, the purple segment on the right figure is a 4 m wide street. The following green segment already narrows to a 2 m wide path and continues to narrow further to a 1 m wide trail. All these segments are combined to one stroke in the basic approach, which is clearly unwanted, as this would mean that if the algorithm decides that the 4 m wide road is important (based on the resulting centrality), it needs to select the small paths and trails as well. On the other hand, if one wants to decide that no trails should remain in the output dataset (as was decided for the Zurich dataset), the algorithm would need to discard the whole stroke, including the 4 m road.

Hence, inserting discontinuities at important points can improve the end result and increases the scope in which the subsequent selection algorithm can operate.

(a)

(b)

Figure 5.18: (a) The purely geometric stroke generator creates very long stroke in the Lucerne area in this case (i.e. the orange stroke in this example). (b) The aforementioned stroke is split up in the improved version, which is a desirable effect, as it splits a small path from an important major road. (© swisstopo)

5.4 Initial Selection

As described in the overview of this chapter (5.1), the creation of the dual graph as well as the calculation of centrality measures follows after the strokes have been generated using the enhanced approach. As these two steps are exactly the same as in the basic approach (Section 4.3.3), they will not be described again in this chapter. Aside from the fact that the selection of the roads based on their centrality value does not produce the final result in the improved approach, as is the case in the basic approach, this step also slightly differs in another point: dependent on the test area, paths (as previously denoted in Figure 5.14) are either excluded from the selection or included. Because of the small scale and relatively high road density, they are excluded in the test areas of Langenthal and Zurich, as they are nearly completely missing in the VECTOR200 dataset of swisstopo as well. A large portion of the Lucerne area however, lies in the Alpine foothills, which contain many important smaller paths and the Davos test area is part of a very mountainous region, which is why paths are included for those two areas.

For the initial selection, the Radical Law (RL) of Töpfer and Pillewizer (1966) was considered, which is depicted in Equation 5.1. n_f denotes the number of objects which can be shown at the derived scale, n_a the number of objects shown in the source dataset, M_a the scale denominator of the source map (in this case 5,000 to 25,000) and M_f the scale denominator of the derived map (200,000).

$$n_f = n_a\sqrt{M_a/M_f} \tag{5.1}$$

If we exclude n_a from the equation and simply look at the multiplying factor ($\sqrt{M_a/M_f}$), the radical law suggests that 15.8% (for 1:5,000) to 35.4% (for 1:25,000) of segments should be retained. The difference between the lower and upper threshold is very high, however. Doubling respectively halving the number of segments in a map could mean the difference between completely unusable and usable. As a consequence, the threshold (T) was determined empirically in the range of the Radical Law. To verify the chosen thresholds, several possible results were generated and evaluated by cartographers of swisstopo. The thresholds were set in such a way that the overall density of the road network resembled the one in the VECTOR200 product of swisstopo. They were then varied such that approximately 10% more and 10% less segments were selected, resulting in three possible thresholds for the individual test areas. The results of this evaluation will be discussed in further detail in Chapter 6, however. For the following part of the improved approach, the algorithms are described and evaluated using the threshold which was considered to be the most accurate.

Figures 5.19 to 5.22 show plots of the number of selected segments as a function of the betweenness threshold of the individual areas. The course of the graphs show an expected behavior. As Jiang (2007, 2009) mentions in his work, the hierarchy of streets follows a power law or heavy tail distribution. Hence, there are few streets which show a high betweenness value and many streets which have a small betweenness value (Jiang, 2009).

In the pictured graphs, the chosen betweenness thresholds (T) are shown in green and the lower and upper thresholds based on the Radical Law (RL_L and RL_H respectively) are indicated in red, to give an idea of the difference in the number of selected road segments. The chosen thresholds (based on the empirical testing and evaluation of swisstopo cartographers) and approximate numbers for the threshold resulting from the Radical Law are shown in Table 5.3. The thresholds for the Radical Law are only an approximation, as the selection is based on complete strokes – hence, the calculated number of segments cannot be matched exactly.

As described in Chapter 3 the intended scale for the TLM3D dataset lies between 1:5,000 and 1:25,000, which is reflected in the upper and lower thresholds calculated by the Radical Law. The number of segments to be pruned from the road network is heavily dependent on the structure and characteristics of the target area. In addition, the resulting road network is intended to be used in conjunction with other thematic layers, which is another factor why the actual threshold may vary widely between different test areas. This is also reflected in the results obtained from empirical testing: with 31.2% retained segments in Davos, the least dense region, the least amount of segments are pruned in relative terms. It also lies relatively close to the higher threshold obtained from the Radical Law (35.4%). In Zurich on the other hand, which shows the highest street density as the other extreme, only 17.1% of the segments are left after the initial selection. This marks the other end of the spectrum and is very close to the lower value obtained by the Radical Law (15.8%). Hence, the percentage of retained segments gets smaller the denser the area gets (at least for the four test areas). The difference between Lucerne (21.3%) and Langenthal (18.3%), despite having a similar overall density, can be explained in the very different characteristics of the test areas (see Section 3.2). A large portion of the Lucerne test area belongs to the Alpine foothills which exhibit a rather low street density, resembling the Davos area. Thus, more segments can be retained in this part of the region, resulting in a higher percentage of retained segments. In general, however, both areas position themselves a bit more in the middle of the spectrum.

There is a possibility that the amount of segments to be pruned could be calculated automatically from the dataset by analyzing the distribution and composition of street segments. However, this thesis focuses solely on four different test areas. To extract such rules for the automatic selection of thresholds, more regions would have been needed to reach a statistically solid solution. It is certainly an option which would be viable to research further, as even the (rather small) dataset suggests that such a rule could indeed be extracted. By finding a rule to automatically select an appropriate threshold based on the underlying data alone, the approach could be further automated.

Table 5.3: Resulting segments and thresholds for the lower (15.8%) and upper (35.4%) boundaries of the Radical Law (RL) and the ones obtained by empirical testing (E).

Test Area	**RL Segments**	**RL Threshold**	**E Segments (% retained)**	**E Threshold**
Davos	1542 3449	0.0440 0.0045	3040 (**31.2**)	0.007
Lucerne	4147 9274	0.0070 0.0006	5596 (**21.3**)	0.003
Langenthal	4951 11072	0.0070 0.0005	5737 (**18.3**)	0.0035
Zurich	10389 23232	0.0055 0.0005	11244 (**17.1**)	0.004

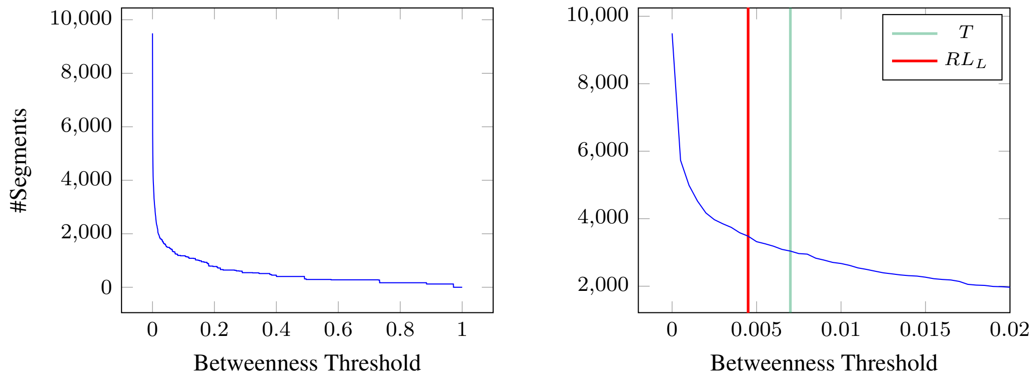


Figure 5.19: Number of segments selected as a function of the betweenness threshold for the Davos area. The plot on the right shows a zoomed in view with the chosen threshold (T) and the lower ceiling resulting from the Radical Law (RL_L) indicated. The upper threshold (RL_H) of 0.044 is not visible in this graph).

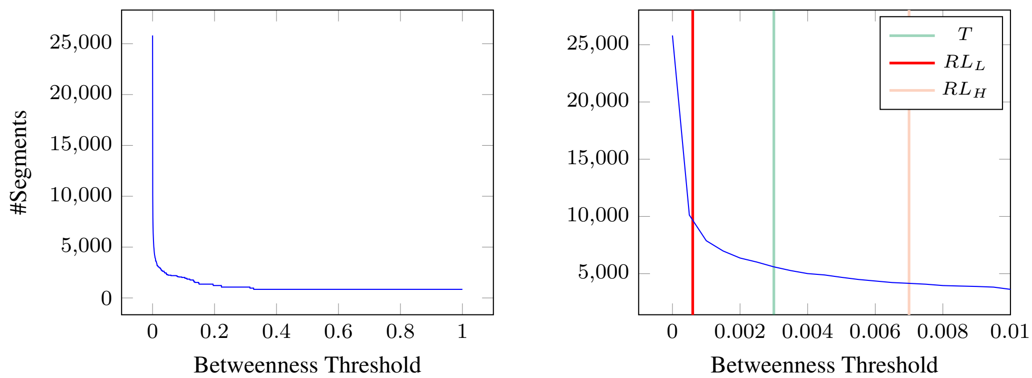


Figure 5.20: Number of segments selected as a function of the betweenness threshold for the Lucerne area. The plot on the right shows a zoomed in view with the actual threshold (T) and the lower (RL_L) and upper (RL_H) ceiling resulting from the Radical Law (RL_L) indicated.

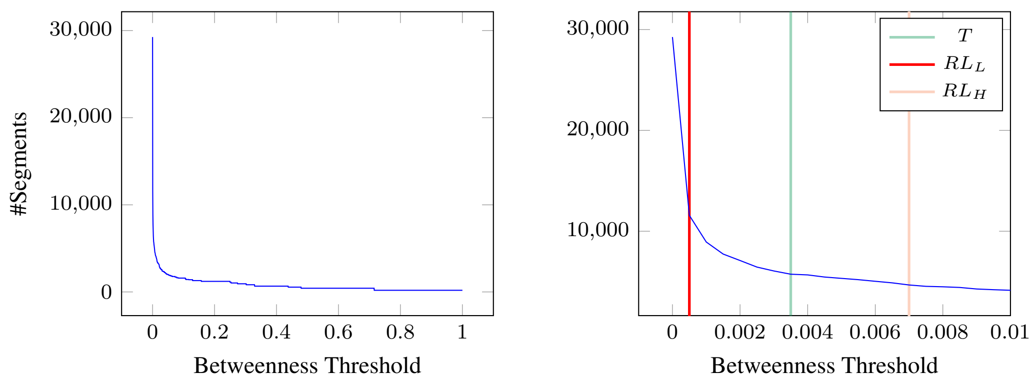


Figure 5.21: Number of segments selected as a function of the betweenness threshold for the Langenthal area. The plot on the right shows a zoomed in view with the actual threshold (T) and the lower (RL_L) and upper (RL_H) ceiling resulting from the Radical Law (RL_L) indicated.

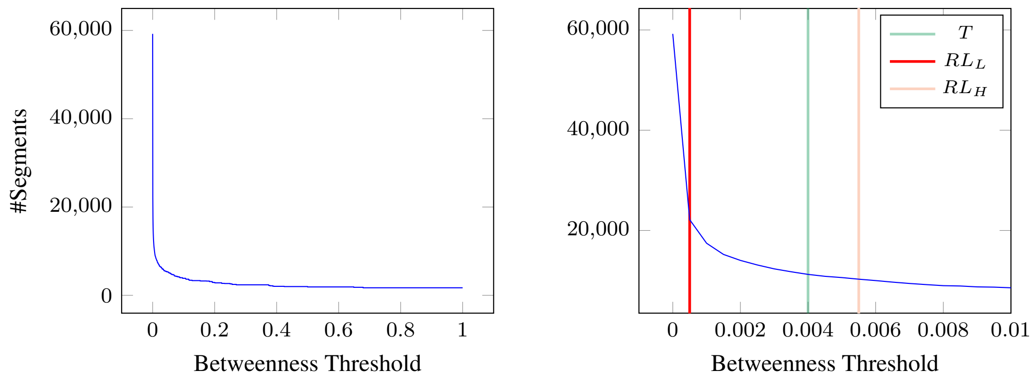
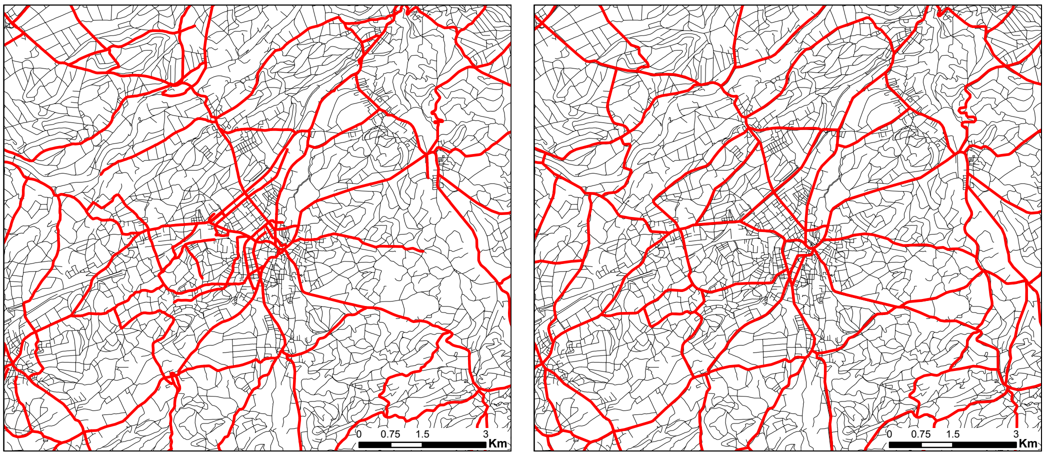


Figure 5.22: Number of segments selected as a function of the betweenness threshold for the Zurich area. The plot on the right shows a zoomed in view with the actual threshold (T) and the lower (RL_L) and upper (RL_H) ceiling resulting from the Radical Law (RL_L) indicated.

In contrast to the basic algorithm, which ends after the selection by centrality thresholds (see Figure 4.1), the improved version uses this selection only as an intermediate solution which will be further optimized. The first two improvements (the elimination of roundabouts and the incorporation of an enhanced stroke generator) focused mainly on improving the generation of strokes. The two remaining enhancements, which will be presented in the next two sections, improve the approach by introducing additional steps after an initial selection has occurred.

5.5 Adaptive Threshold based on Road Density

In Section 4.4.3, the density problem of the basic approach has been described. Urban areas remain too dense and rural areas tend to be thinned out more. While a density difference in urban and rural areas is desired to some extent in order to retain the structure of the original network, urban areas remained still too dense after the initial pruning. Figure 5.23 shows an example where the City of Langenthal can be seen in the center of the map. Figure 5.23(a) shows the result after the initial selection, with the roundabout elimination and enhanced semantics improvements already in place. While the general density outside of the city is very good, the city itself is too dense when compared to the VECTOR200 data of the same area, pictured in Figure 5.23(b). An incrementation of the betweenness threshold may reduce the road density of the city, but it does so also for the other parts. However, when comparing the rural regions of the two pictures, it is evident that a further decrement of the density would result in a too loose road network around the city. Because the density of urban regions and the density of rural regions are tied to the same global threshold, the problem is drastically increased in its complexity.



(a) (b)

Figure 5.23: (a) The area around the city of Langenthal after an initial selection was performed using the improvements presented in Sections 5.2 and 5.3. (b) The same area using the VECTOR200 data of swisstopo as a comparison. It can be seen that the density in the city itself remains too high compared to the rural region surrounding the city. (© swisstopo)

To tackle this problem, one has to find a way to control the density of rural and urban areas separately. An initial idea was to use the settlement areas of the TLM to identify regions where the road network should be pruned more extensively. However, closer inspection revealed that the settlement dataset contains too many such regions, as even very small settlements are included. Even though the strokes in settlement areas have generally higher centrality values, this is not *always* the case, especially not in very small, rural towns. Hence, the algorithm would reduce the density in areas which are not even targeted. This is the reason why an approach using cluster analysis was developed using the selected result as a basis instead of the source dataset. Thus, the algorithm aims to identify regions which remain too dense after an initial selection and adapt the density in these regions. Figure 5.24 shows an overview of the algorithm. It looks for dense regions using a DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise; see Ester et al., 1996), identifies the strokes which are potentially affected and adapts the betweenness threshold for these strokes.

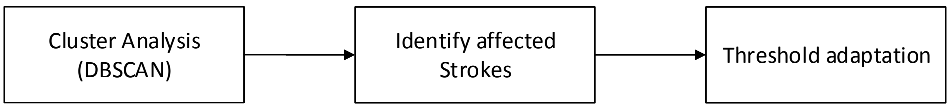


Figure 5.24: Overview of the algorithm used to adapt the road density in dense regions.

5.5.1 Methods

Cluster Analysis. In order to adapt the thresholds for certain regions, they first have to be identified. As explained above, the settlement layer proved to be ineffective to do that, which is why a clustering algorithm has been implemented to adapt for the unique situations which arise after an initial selection in the different test regions has occurred. Figure 5.25 depicts the method which was used to identify clusters. As a basis for the algorithm, the centroid points of the segments were used. To detect clusters, the DBSCAN algorithm of Ester et al. (1996) was used, which will not be discussed in detail. Important to note is however that the algorithm can be parametrized using two parameters: ϵ and *MinPts*. To decide whether a point p is inside a cluster (i.e. a dense region), DBSCAN checks how many points (*MinPts*) exist in a circular neighborhood of a certain radius (ϵ) around p, called ϵ-neighborhood. If the number of points in the ϵ-neighborhood of p exceeds *MinPts*, p is considered to be in a dense cluster. While Ester et al. (1996) proposed to use a fixed *MinPts* parameter of 4 for 2-dimensional datasets, this proved to be ineffective in this case. A reason for this is probably the special nature of the dataset, with the resulting centroid points of the segments arranged in a linear fashion. Hence, extensive empirical testing was necessary to reveal possible parameters which give good results. For each region, many pair-wise combinations were tested using a *MinPts* parameter of 2 up to 300 and an ϵ parameter of 100 m up to 3000 m. Several characteristics of the resulting clusters have been analyzed, such as average density per cluster, total number of segments per cluster or total segment length. The parametrization of this step was especially difficult as the number of clusters, their size and density were all important factors. The clusters between each region and even inside of them varied widely in size. An algorithmic solution to find optimal parameters was also unsuccessful, which is why possible solutions needed to be inspected visually to converge to a good solution. The parameters used for the different areas are listed in Table 5.4. This is by no means the only possible solution, as different combinations of the parameters lead to very similar results. For example, reducing ϵ to 750m and in turn also reducing *MinPts* results in nearly identical clusters which only vary slightly in size. In addition, slightly different clusters at different locations were also possible.

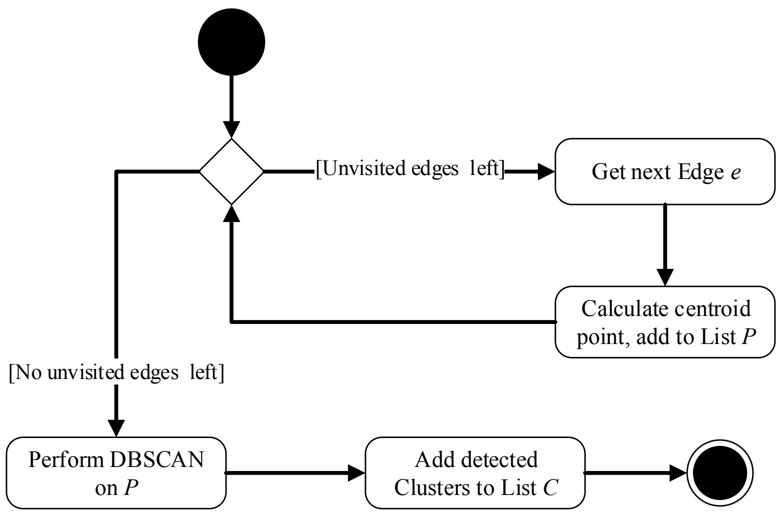


Figure 5.25: UML activity diagram of the clustering algorithm. DBSCAN is not explained in detail, as there exists literature which thoroughly explains the algorithm (e.g. Ester et al., 1996).

Density Selection. In Figure 5.27, the second part of the algorithm is visualized. As a first step, the convex hull h of each cluster c is calculated to generate the areas which are considered to be of too

Table 5.4: Parameters used for the DBSCAN algorithm and number of detected clusters for each test area.

Test Area	MinPts	ϵ [m]	#Detected Clusters
Davos	60	1000	8
Lucerne	120	1000	2
Langenthal	120	1000	4
Zurich	120	1000	9

high density. Figure 5.26 shows a snippet of the Langenthal area showing the generated convex hulls with its related points which were considered as *density-connected* or *density-reachable* (see Ester et al., 1996). Now, two aspects have to be considered: (1) The algorithm has to decide for which strokes the threshold needs to be increased and (2) how much the threshold needs to be increased. A first attempt could be to simply increase the thresholds of all strokes which are part of the dense area, i.e. all strokes which are represented by the dense points. However, this would produce unwanted results: strokes which are only partially inside of the dense area would be affected as well, which would cause problems with strokes spanning over large portions of the area. This means that the algorithm could potentially increase the threshold of strokes which are essential for the structure of the road network, which is why only strokes are considered which are mostly inside dense areas. This is defined as the percentage of the stroke length which lies inside such areas and is denoted as *pInside* in Figure 5.27. For the strokes which are contained inside dense areas (i.e. more than *pInside* of the stroke is contained in dense areas), the betweenness threshold T is increased by the multiplier specified in M.

Table 5.5 shows the parameters used for each region. *pInside* was set to 0.6, i.e. a stroke is considered to lie inside of a dense area if at least 60 % of its length is contained in a dense area. The multiplier M was varied for the different test areas and was chosen by empirical testing. For the mountainous Davos area, a high multiplier was chosen, whereas for the urban Zurich region, the multiplier was set to a relatively low value. The multiplier for Lucerne and Langenthal was in between the other two regions.

Table 5.5: Parameters used for the density algorithm for each area

Test Area	pInside	M	T	Adapted T
Davos	0.6	7	0.007	0.049
Lucerne	0.6	4	0.003	0.012
Langenthal	0.6	4	0.0035	0.014
Zurich	0.6	2	0.004	0.008

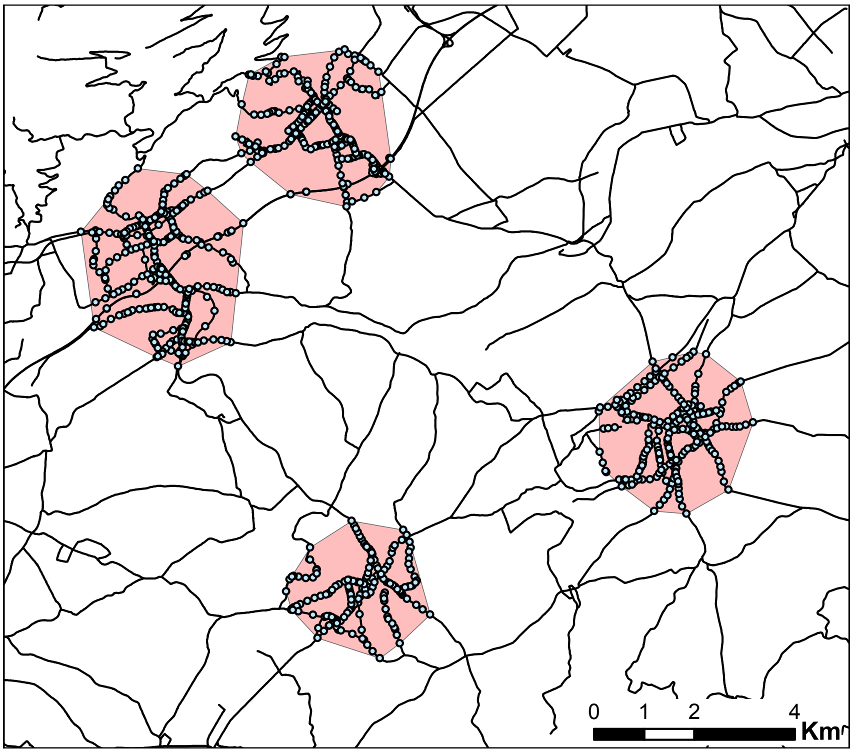


Figure 5.26: Result of the DBSCAN algorithm in an extract of the Langenthal area, showing 4 different clusters. The centroid points of the individual street segments which were considered as *density-connected* or *density-reachable* are shown with the respective convex hulls, enclosing the dense regions. (© swisstopo)

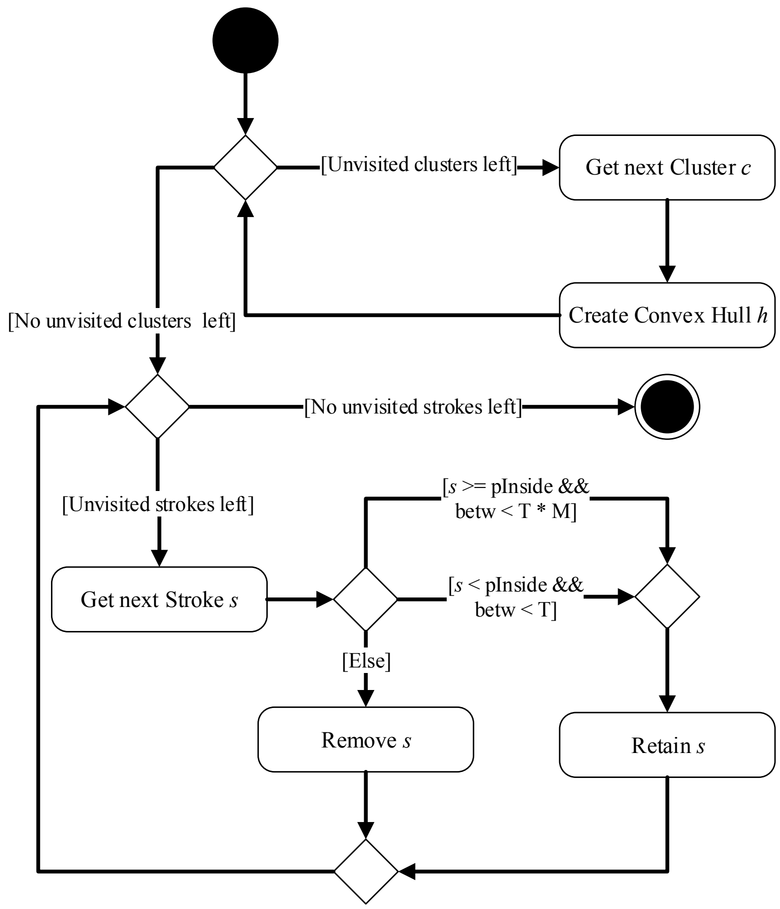


Figure 5.27: UML activity diagram of the adaptive density algorithm, which iterates over all identified clusters and extracts all strokes which are contained in their convex hulls. The betweenness threshold is increased for the affected stroke by the multiplier M.

5.5.2 Results and Discussion

This section presents some selected results of the algorithm. Table 5.6 shows the number of segments and strokes which were removed as a result of the density adaptation. From a relative standpoint, the most segments and strokes were removed in Davos. This is also reflected in the chosen parameters previously listed in Table 5.5, with Davos having a much higher threshold multiplier than the other regions. Additionally, the *MinPts* parameter used in the cluster detection algorithm was also halved compared to the one used in the other regions to adapt for the generally lower density of the mountainous area. As mentioned in Section 5.4, where the results of the initial selection were discussed, the initially high percentage of retained segments of the Davos area could only be seen as an initial approximation. Because the number of segments will again be changed in the final improvement, the changes and the consequences for the comparison with the Radical Law will be presented in Chapter 6.

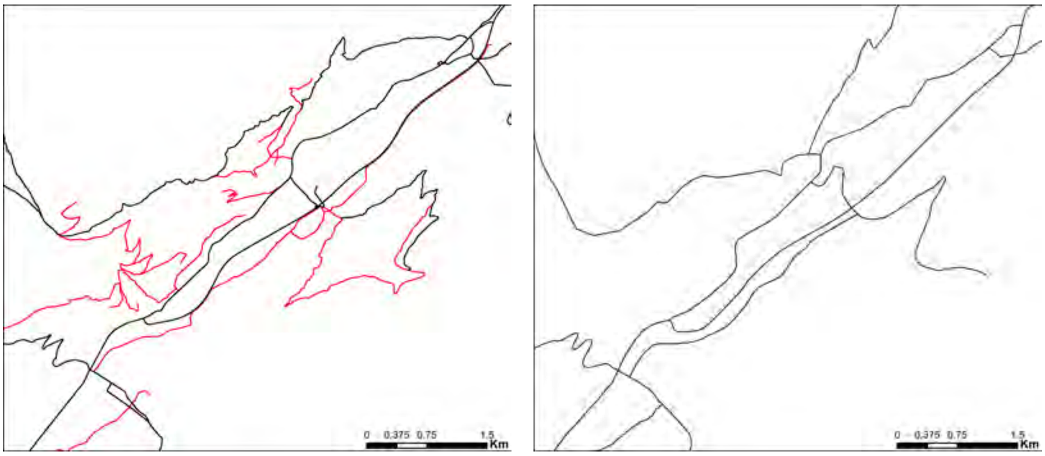
If we compare the retained segments of the Lucerne and Langenthal test areas, it becomes clear that more segments and strokes were removed in the Lucerne region, despite using exactly the same parameters as in the Langenthal region. The reason for this lies in the structures of the different areas. The Lucerne area contains the large City of Lucerne, which is marked as a dense region. While only 2 clusters were detected in Lucerne compared to 4 in Langenthal, the clusters in Lucerne are much larger, resulting in more strokes being affected by the adapted threshold.

Despite the lower threshold multiplier used for the Zurich area, the number of removed segments is comparable to the Lucerne region, in relative terms. This can also be explained with the structure of the area: the road network in the whole area is very dense, which resulted in many large clusters. Even though the strokes which were contained in dense regions were selected using a lower threshold multiplier, many more were affected, which resulted in the deletion of a similar amount of segments.

Table 5.6: Number of segments and strokes before and after the density adaptation.

Test Area	#Segments before	#Segments after	#Strokes before	#Strokes after
Davos	3040	2385	235	161
Lucerne	5596	4743	291	232
Langenthal	5737	5467	258	237
Zurich	11244	10329	536	465

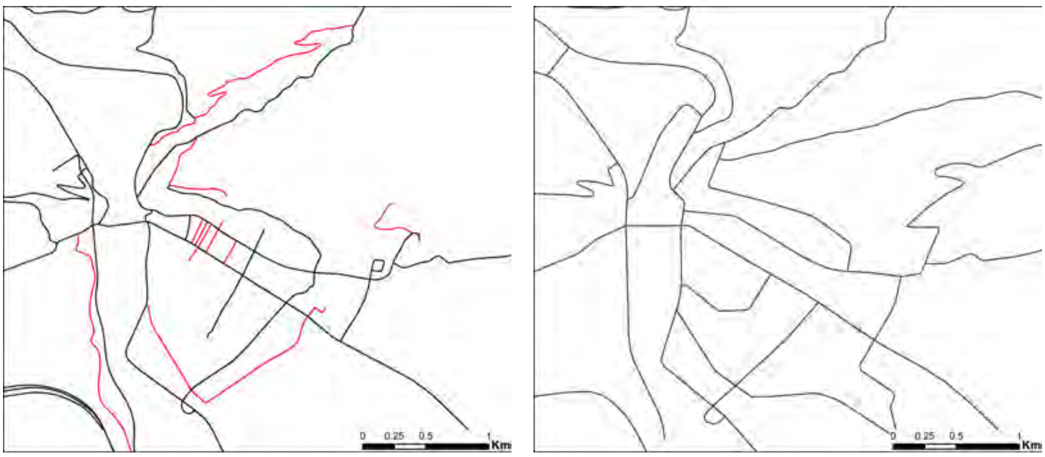
Figure 5.28(a) shows a snippet of the Davos area. The segments which were removed as a result of the adaptive threshold are marked with red color. While it looks as if too many segments were removed, a comparison with Figure 5.28(b), which shows the same snippet of the VECTOR200 dataset, suggests a different conclusion. In the original selection (the red and black segments combined), the density of the road network was much higher than in the VECTOR200 dataset. With the removal of the red segments (i.e. the segments which had a betweenness value which was below the raised threshold in the marked area), the overall density of the snippet could be approximated to the version of swisstopo. It also shows again a problem of the stroke-based approach, as some more dead-ends are created as a result of removal the of additional strokes. The problem is not necessarily caused by the density algorithm, but is inherent to all stroke-based algorithms: different thresholds result in different dead-ends. Because the next improvement (Section 5.6) alleviates this issue, it does not influence the final result heavily, however.



(a) (b)

Figure 5.28: (a) A snippet of the Davos test area showing the removed segments based on the adaptive threshold in red. (b) The same area using the VECTOR200 data of swisstopo as a comparison. While not exactly identical, the overall density of the region can be approximated using the adaptive threshold. Potential dead-ends which are created as a result of this are handled in the next improvement. (© swisstopo)

Figure 5.29(a) shows a snippet of the Zurich test area. Again, the segments which were removed as a result of the adaptive threshold are marked in red. Because of the lower betweenness value, several streets were removed in this area. The same area of the VECTOR200 dataset is shown in Figure 5.29(b). A comparison between the two figures shows that the removal caused by the adapted density results in a better overall road density.



(a) (b)

Figure 5.29: (a) A snippet of the Zurich test area showing the removed segments based on the adaptive threshold in red. (b) The same area using the VECTOR200 data of swisstopo as a comparison. While not exactly identical, the overall density of the region can be approximated using the adaptive threshold. Potential dead-ends which get created as a result of this are handled in the next improvement. (© swisstopo)

As Figures 5.28 and 5.29 show, the road density in the network could successfully be reduced in areas where a high density was detected after the initial selection. The overall structure of the network resembles the one in the VECTOR200 product of swisstopo more closely using an adaptive threshold. However, the results also reveal several shortcomings of this approach. First, it still uses a stroke-based selection, inheriting all the related problems, such as the creation of new dead-ends. This is only a minor problem, if some sort of reconnection algorithm is implemented (as it is the case in the improved approach). In addition, dead-ends are created in every stroke-based approach, hence one can assume that a stroke-based algorithm which incorporates this improvement either ignores dead-ends completely or has some method in place to eliminate them. Another, more important problem, is the fact that it is rather difficult to handle areas in isolation using a stroke-based approach. As mentioned in the method description, the algorithm has to make sure not to influence strokes which are structurally important to the whole region, only because a small part of them passes through a dense region. This is why only strokes are affected which are at least 60 % inside dense regions. Theoretically, it is also possible that the density of a region cannot be reduced at all if there exist only long strokes which do not fulfill the 60 %-requirement. Hence, the density adaptation algorithm is only a best-effort solution and does not guarantee an appropriate density reduction. As a last problem, the rather difficult parametrization needs to be mentioned as well. First of all, the DBSCAN algorithm needs to be parametrized correctly to detect dense regions. As a result of the rather unpredictable behavior of the stroke-based centrality approach, it is difficult to automatically extract appropriate parameters. In addition, the multiplier for the adapted threshold also has a large influence, which also needs to be empirically and visually evaluated.

For the purpose of this work however, the algorithm worked very well, especially in conjunction with the reconnection algorithm, which is presented in the next section.

5.6 Reconnection of Dead-Ends and Disconnected Parts

A major problem that exists in the basic approach is the creation of new dead-ends and disconnected parts. While solutions to the removal of dead-ends exist (for example, the usage of MSTs; Yang et al., 2011), they do not solve the problem of dead-ends. Even for larger scales, a strict requirement for swisstopo is that no new dead-ends are created except for special circumstances (e.g. to provide access to points of interest Benz, 2013). For smaller scales, this is true as well with the exceptions that cartographers are allowed to insert dead-ends to hint at certain access-roads or special relationships between roads. In general however, there exist very few dead-ends in the VECTOR200 map of swisstopo. The main reason for this circumstance is that dead-ends deteriorate the ability of a map to offer guidance for a user in finding appropriate routes. In addition, existing dead-ends are often not of high importance (e.g. access-roads in settlements).

Because MSTs do not eliminate dead-ends but could even produce new ones, they were not used in this approach. Another important aspect is that the detection of dead-ends and disconnected parts alone is not difficult, hence such situations could also be remedied by simply removing the affected road segments. However, as is evident in the results of the basic approach (Section 4.4), there exist quite many dead-ends and the removal of these would cause the network to deteriorate heavily. This is why a completely different approach was implemented to solve the problem of dead-ends and disconnected parts alike, both of which are directly caused by the stroke-based approach.

The presented method tries to reconnect dead-ends to the main (pre-selected) network using a custom set of rules. The process can be parametrized differently for different regions to adapt accordingly.

The algorithm can be subdivided into the following steps:

1. **Identification of dead-ends.** As a first step, the algorithm needs to find all dead-ends, which is an easy undertaking.
2. **Generation of reconnecting paths.** Next, the algorithm creates all possible reconnecting paths from the dead-end to the main network.
3. **Filtering of paths.** As a final step, the best reconnecting path needs to be filtered, which is done by using a variety of rules and constraints.

5.6.1 Methods

Identification of dead-ends. As already explained in Section 5.2, where dead-ends in strokes were removed to detect roundabouts, the sole detection (and elimination) of dead-ends is relatively straight-forward. If the number of adjacent edges of a node in a graph is exactly 1, it is a dead-end. For the purpose of this improvement, no distinction between disconnected parts and dead-ends is made, i.e. a disconnected part is simply made up of at least two dead-ends and handled as such. Hence the complete graph of the initially selected road network is checked for dead-end nodes. Because dead-ends were created at the edges of the test areas when they were initially selected for this thesis, a mask was created which excludes all dead-ends located inside a buffer at the edges of the test areas, in order to remove these edge effects (see Figure 5.30). After the identification of dead-end nodes, the corresponding dead-end paths are extracted, as depicted in Figure 5.31. This is done by starting from the end of the dead-end and moving back as long as the number of adjacent edges is equal to or below 2. At this point, a decision is made to either try to reconnect the dead-end to the main network or to delete it (i.e. truncating its path). This decision is purely based on the length of the dead-end path. If it is below a length threshold, it is marked as too short. This threshold was set to 800 m for all test areas, translating to 4 mm on a 1:200,000 scale map. The threshold simply determines the minimum length of a dead-end path to be processed in the reconnection algorithm. If the dead-end path contains a segment which belongs to a highway, it is excluded from this threshold, to ensure that everything related to highways does not abruptly end. A shorter dead-end generally means that it is less important and that its removal does not heavily deteriorate the pruned network. At the same time, it is harder for the reconnection algorithm, which will be discussed shortly, to find an appropriate reconnecting path. The minimum length is not a necessary criterion, but it helps to eliminate unimportant dead-ends. Empirical testing has shown that a threshold of roughly 800 m or below eliminates the most unimportant dead-ends, while still retaining most, if not all of the relevant ones.

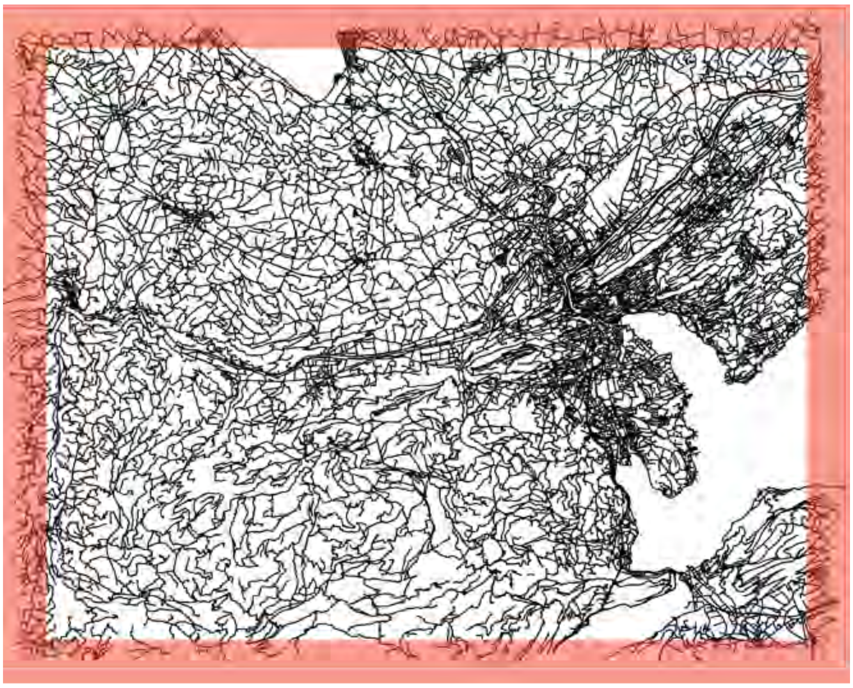


Figure 5.30: Buffer used in the Lucerne test area where dead-ends are ignored to eliminate edge-effects. (© swisstopo)

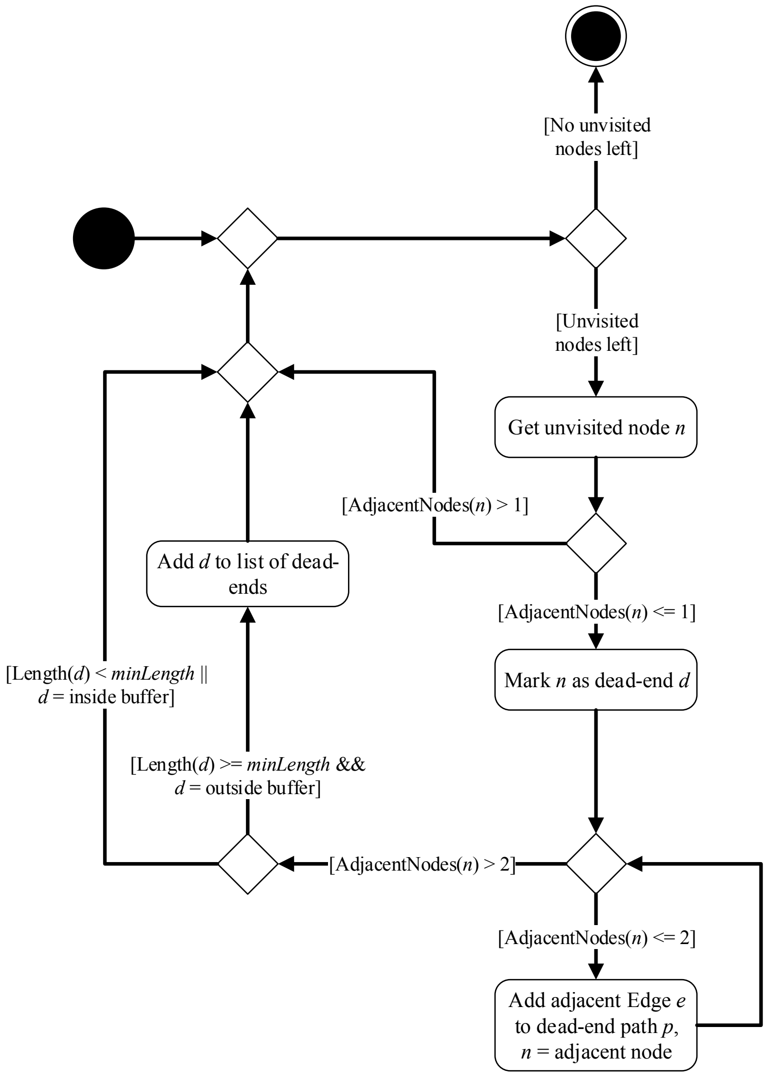


Figure 5.31: UML activity diagram of the dead-end identification process and the creation of the corresponding paths.

Generation of reconnecting paths. After all of the relevant dead-ends and their corresponding paths have been identified and stored in a list, the algorithm continues with the generation of reconnecting paths. The main idea behind the algorithm is to generate all possible paths which connect the dead-end node with a node which is also present in the initially selected graph, generated in Section 5.4. Figure 5.32 depicts the algorithm used.

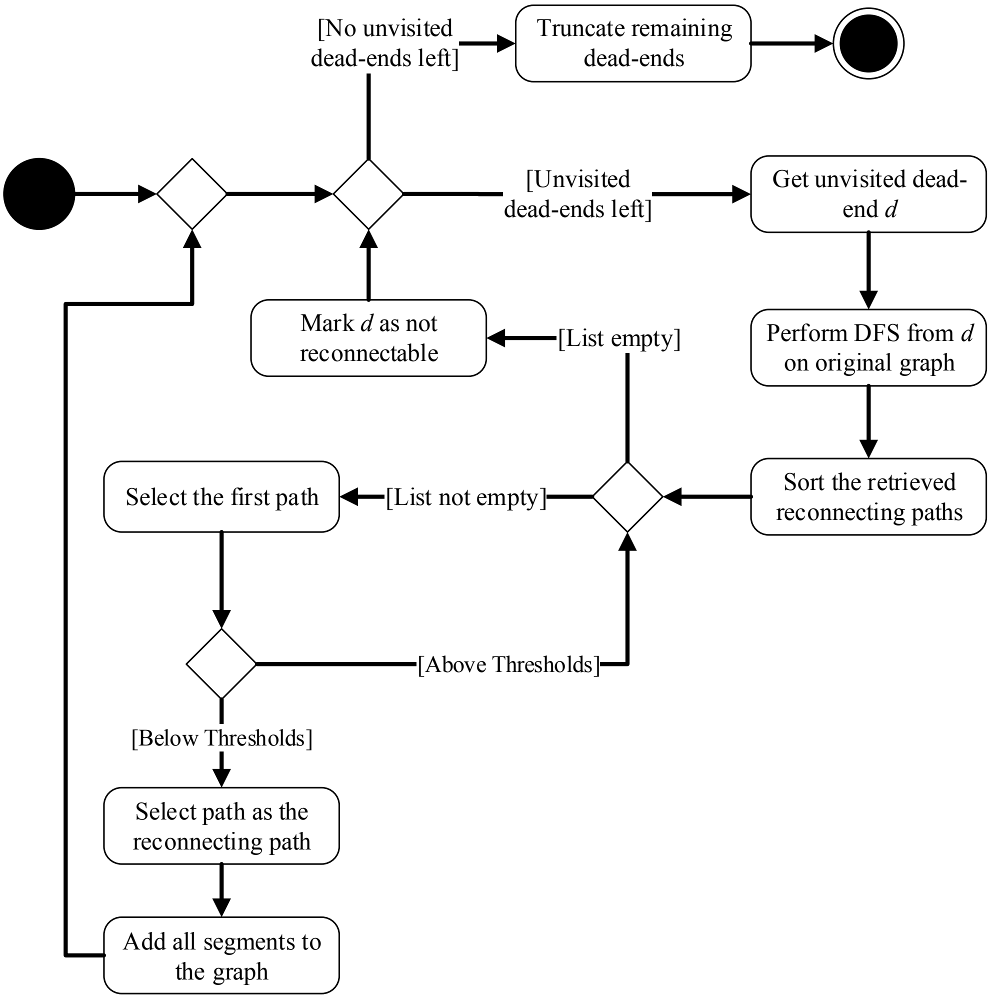


Figure 5.32: UML activity diagram of the reconnection process. The algorithm first creates all possible paths and filters the paths afterward. The best fitting path is selected at the end and added to the selection result.

The algorithm iterates through the previously detected dead-ends and first performs a *Depth-First-Search* (DFS; see Goodrich et al. (2004) for details) in the original, non-pruned graph. The DFS starts at the end-node of the current dead-end of the iteration and backtracks as soon as a node is found which is also present in the initially selected graph, forming a path from the end-node of the dead-end to the node of the initially pruned graph in the process. The path is then stored as a candidate path which reconnects the dead-end to the main network. The DFS continues until all possible paths have been generated. Because the number of paths exponentially increases with the number of nodes, the depth of the DFS is limited to 20 (i.e. a reconnecting path may contain a maximum of 20 nodes). In the tested areas, this was enough to produce all possible, relevant paths. Paths which consist of more than 20 nodes are highly unlikely to be of use as they simply consist of many turns (note that the DFS is not limited in any way, so a path could possibly move for 3 nodes in one direction, turn around, move in the opposite direction, and so forth). In addition, the DFS is configured in such a way that no loops are allowed (no node may be used more than once for each path). Also, no paths are allowed which end at the dead-end path (as they fulfill the requirement of connecting to a node of the initially selected graph, but obviously do not truly reconnect the dead-end or disconnected part to the main network).

Reconnection Comparator. Once all paths have been generated, the paths are sorted. This approach is similar to the one used in the stroke building process (see Section 5.3), as a number of possibilities (in this case, the paths) are generated, sorted according to some rules and checked against different thresholds. In this case, the sorting is based on two parameters: (1) the class of the path and (2) the length of the path. As with the stroke algorithm, a comparator is used to sort the paths.

In the following, P_1 and P_2 are two reconnecting paths to be compared and P_D is the dead-end path. The highest road class which is present in a path is denoted as $C_H(P)$ (see Section 5.3 and especially Figure 5.14 for details). The comparator checks the following parameters in sequence:

1. If $C_H(P)$ of exactly one of the two paths to be compared is lower or equal than $C_H(P_D)$, then this path is preferred. In other words, paths which contain a segment with a road class which is higher (i.e. narrower) than the highest class in P_D, are penalized. The comparator should prefer paths which are at least as wide as the narrowest segment in the dead-end path.
2. If neither of the paths contains a segment with a road class which is higher than the highest one in P_D, then the comparator distinguishes between two special cases:
   a) If $C_H(P_D)$ is 1 (i.e. the dead-end path's maximum class is *Highway*), then the comparator prefers paths where $C_H(P)$ is also equal to 1 or, as a second option, it prefers the path which has a lower $C_H(P)$.
   b) If $C_H(P_D)$ is 2 (i.e. the dead-end path's maximum class is *Major Road*), then the comparator prefers paths where $C_H(P)$ is lower or equal to 3. Hence, it makes no distinction between major roads or minor roads, it simply prefers paths which do not exceed the class of minor roads.
3. As a last step (if no decision could be made with the previous checks), the comparator compares the length of the two paths and prefers the shorter one.

Hence, the comparator uses the highest road class of the dead-end path as a reference and tries to find a reconnecting path where no segment has a higher class, with some special cases for highways and major roads. If no decision can be made, the length is used as the main criterion. This guarantees that the comparator does not choose a reconnecting path, which should reconnect, for example, a major road, which contains narrow paths or even trails over a path which has the same or a lower road class but is slightly longer.

Reconnection thresholds. As mentioned in the beginning, the algorithm should choose paths which have a similar class, are of short length and lead in the same general direction as the dead-end path. The first two criteria are included in the comparator, i.e. the algorithm already prefers short paths with a similar highest road type. The angle criterion, however, is only checked after the paths have been sorted.

Figure 5.33 shows the main principle behind this criterion. It shows a dead-end path in red and a corresponding reconnecting path in green. The colored polygon denotes the area where a reconnecting path is allowed to reconnect to the main network according to the angle criterion (the distinction between the two colors of the polygon is explained later). If the end-node of a path falls outside of this region, it is considered to lead in an unwanted direction. This could also be described as another way of determining the good continuity criterion, very similar to the method used to build strokes, but in a broader fashion (only the end-nodes of paths are compared). The algorithm compares the angle (α), which is formed between two lines: the (straight) line that is formed by connecting the start-node of the dead-end path with the end-node of the dead-end path and the line that is formed by connecting the start-node of the dead-end path with the end-node of the reconnecting path. The maximum angle was set to 60° for all test areas.

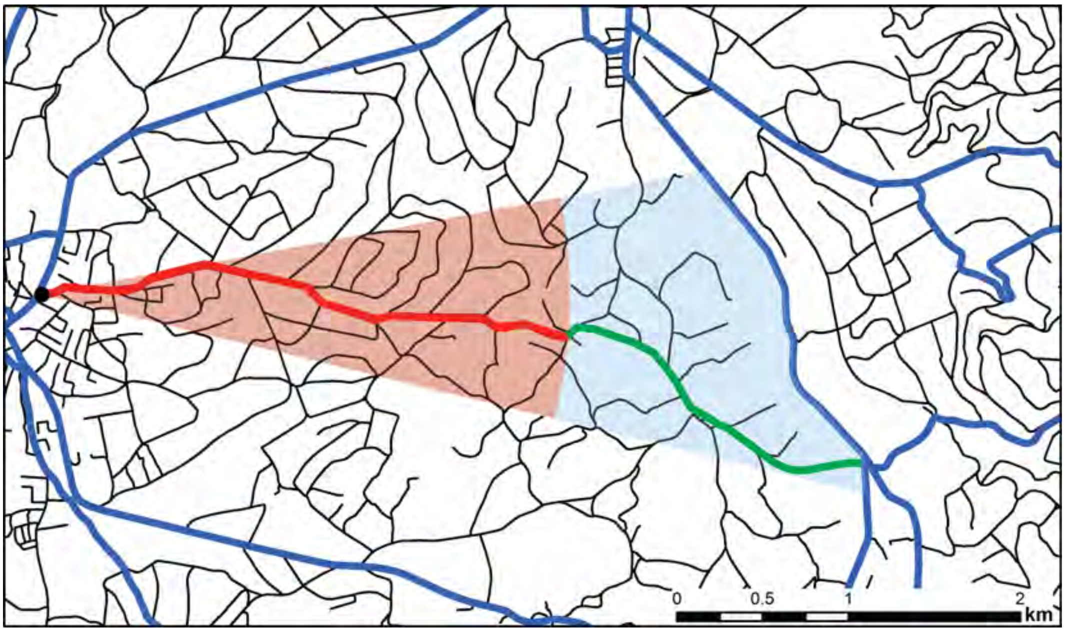


Figure 5.33: Conceptual illustration which shows the principle behind the direction criterion to find a suitable reconnecting path. The end-node of the reconnecting path (green), which reconnects the dead-end path (red) to the main network (blue), needs to be inside of the light blue region. The red area denotes the region where the angle criterion is also fulfilled. However, the end-node of the reconnecting path may not be closer to the start of the dead-end than the end-node of the dead-end path. (© swisstopo)

Thus, the algorithm iterates through the sorted list of potential reconnecting paths and chooses the first one which fulfills *all* of the following criteria:

1. The angle α has to be below the defined threshold of 60° in order to guarantee a good overall continuity of the dead-end path and reconnecting path.
2. If the dead-end path belongs to a highway (which includes highway exits and entries), the reconnecting path is not allowed to reconnect to a highway to ensure that an entry does not connect again to an exit. Hence, a dead-end which belongs to a highway exit, must connect to a major or minor road, for example.
3. The end-node of the reconnecting path must be farther away from the start-node of the dead-end path than the end-node of the dead-end path. This criterion, including a 100 m buffer, is in place to ensure that the reconnecting path does not lead in the opposite direction. As hinted at earlier, this is the distinction between the two colors of the polygon in Figure 5.33. This criterion, together with the angle criterion, ensures that a path must end in the blue area.

If a path is found which fulfills all of these criteria, it is marked as the reconnecting path for the corresponding dead-end. After every dead-end has been processed, the reconnecting paths are added to the selection. Every dead-end path for which no reconnection could be found (either because it is a real dead-end which also exists in the TLM3D dataset or because no path could fulfill all criteria), is truncated afterward. Thus, no dead-ends (except for the ones at the edges of the test area) exist in the final result.

One could ask why the stroke-method was not also chosen for this step, as it also seeks to preserve continuity. While this would certainly be an option, the question can be answered if one recalls the reason why the reconnection algorithm is even necessary in the first place: it is because strokes were split up at such dead-ends (e.g. the angle criterion was not fulfilled). Hence, it would be unwise to use the stroke-method again where it failed before. This is why not only the angle criterion, but also the other rules used in the comparator, are less strict.

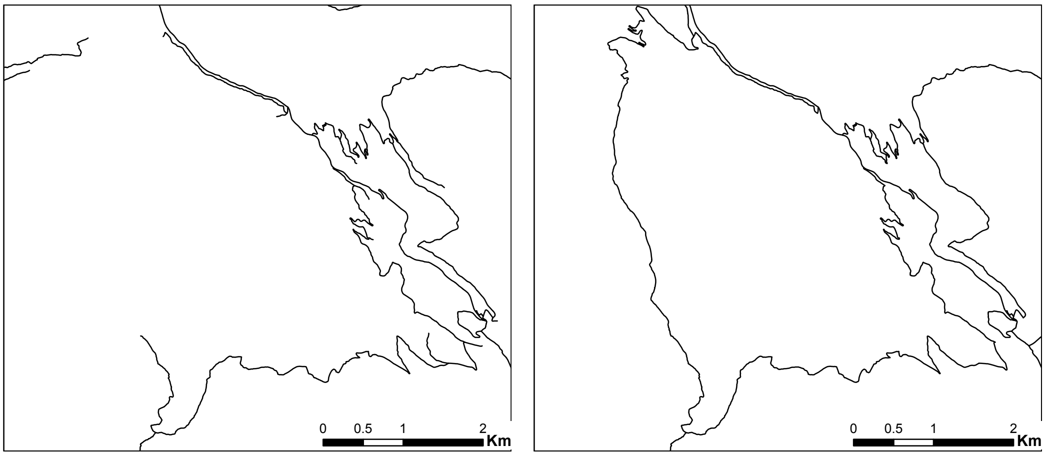
5.6.2 Results and Discussion

Table 5.7 shows the number of detected and reconnected dead-ends of each test area. Note that the dead-ends which are located inside the edge buffer (see Figure 5.30) have already been excluded. Between 38 % and 57 % of the detected dead-end paths could be reconnected. The rest of the paths were truncated. A look at the change in the number of total segments in the result shows that depending on the test area, an increase (Davos and Zurich) or a decrease (Lucerne and Langenthal) has taken place.

Table 5.7: Number of detected and successfully reconnected dead-ends.

Test Area	**#Dead-ends detected**	**#Dead-ends reconnected**	**#Segments before**	**#Segments after**
Davos	65	37 (57 %)	2385	2496
Lucerne	105	40 (38 %)	4743	4718
Langenthal	73	39 (53 %)	5737	5499
Zurich	167	88 (53 %)	10329	10352

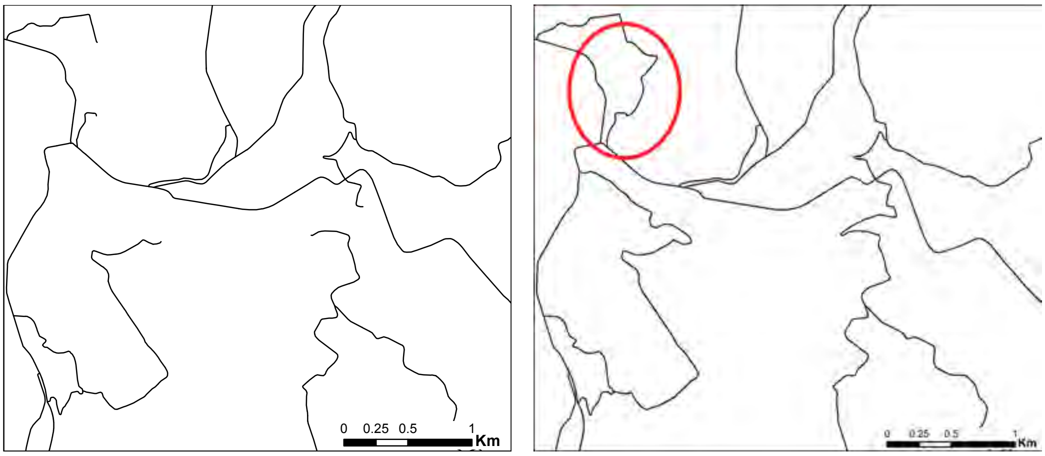
Figure 5.34 shows a snippet of the region of Davos which shows the result of the reconnection algorithm. On the left, an area in the Davos test area is shown which exhibits many dead-ends. On the right, the corrected version of the same area is shown: some dead-ends could be reconnected to the main network and the overall structure could be significantly improved. Dead-ends which could not be reconnected (either because the possible paths could not fulfill the threshold criteria or because the dead-end already existed in the original dataset) were simply truncated.



(a) (b)

Figure 5.34: (a) A snippet of the Davos test area showing many dead-ends caused by the stroke selection algorithm. (b) The same area after the reconnection algorithm has processed the region. Dead-ends were either reconnected if a suitable path could be found or truncated. (© swisstopo)

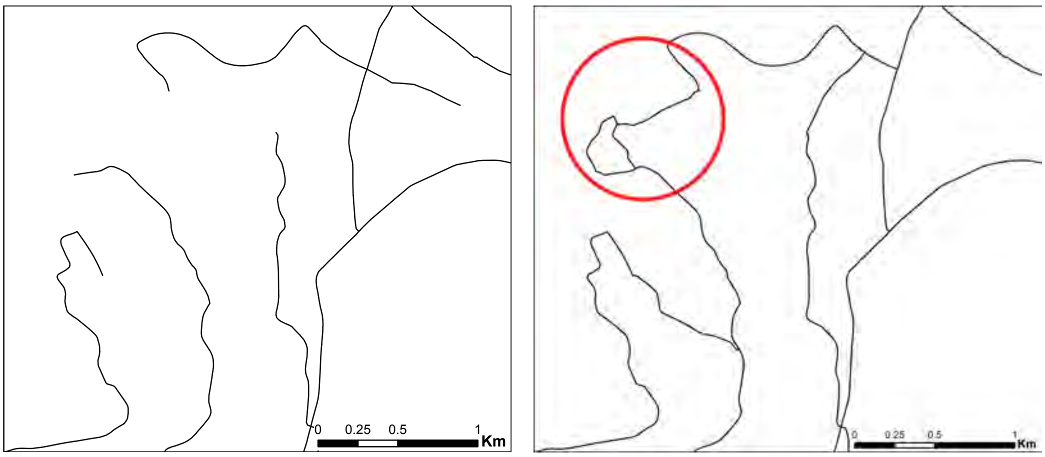
Figure 5.35 again depicts a snippet of the Lucerne dataset with and without the reconnection algorithm. The red ellipse in Figure 5.35(b) shows an example where a reconnection did not make sense. If we consider the path in the north, the reconnection to the lower path would have been excluded, as it violates the criteria that the end of a reconnecting path must not be closer to the start of the dead-end path than the end of the dead-end path. If we look at the situation starting from the lower path, the reconnection is allowed, as it fulfills all necessary criteria (granted that no segment of the reconnecting path has a lower road class than the dead-end path). Hence, the algorithm connected one dead-end with another dead-end, which was only allowed from one direction. This was not considered in the algorithm and could certainly be improved further. Nevertheless, the algorithm successfully found good paths to reconnect two other dead-ends.



(a) (b)

Figure 5.35: (a) A snippet of the Lucerne test area showing many dead-ends caused by the stroke selection algorithm. (b) The same area after the reconnection algorithm has processed the region. Dead-ends were either reconnected if a suitable path could be found or truncated. The red ellipse points to a path where a reconnection only makes sense from one direction, however. (© swisstopo)

Figure 5.36 depicts a snippet of the Zurich area, again without the reconnection algorithm on the left and with it on the right. A total of five dead-ends can be seen before the reconnection, one of which was deleted in the second picture, as no suitable reconnection was found. However, the red ellipse again points to a problem: the two dead-ends connect to each other using two distinct paths. First, the upper one uses a reconnecting path which ends in the middle of another dead-end path. The lower dead-end is not reconnected with this step, however, and thus uses a different path. The reconnection of the other two dead-ends and the truncation of the one on the right can be seen as a very good result, on the other hand.



(a) (b)

Figure 5.36: (a) A snippet of the Zurich test area showing many dead-ends caused by the stroke selection algorithm. (b) The same area after the reconnection algorithm has processed the region. Dead-ends were either reconnected if a suitable path could be found or truncated. (© swisstopo)

The reconnection algorithm could successfully reconnect many dead-ends in all of the test areas. While the success rate varied from 38 % to 57 %, the overall structure of the road network could be enhanced significantly, as all dead-ends have either been removed or reconnected. Note that dead-ends which already existed in the source data set are also included in the above numbers. The constraints used to filter the paths in general were very useful in finding a well suited path which reconnects a dead-end to the main network. There are several areas where the algorithm could be improved in a potential future version, however:

- In the current state, the algorithm analyzes the initial selection (see Section 5.4) once in the beginning and continues to find a best path for each dead-end in isolation. To improve the algorithm and to avoid situations as the one depicted in Figure 5.36(b), the algorithm should re-evaluate the situation after every single dead-end reconnection. While the algorithm checks whether a reconnecting path ends at another dead-end, it does not check whether it ends in the middle of a dead-end path.
- Currently, the algorithm searches for a reconnecting path starting at an end-node of a dead-end. Hence, it assumes that the dead-end path is "correct", that is, it assumes that the generated strokes are already optimal. Another assumption could be that the stroke generator made a mistake (in the sense that while it did choose the best path from a local viewpoint, it did not do so looking at the situation in a broader view).
  For example, let us consider a situation where the stroke generator needs to decide between two segments of exactly the same class, diverting to the angle criterion. The segment forming the smaller angle abruptly ends unexpectedly, however, and creates a dead-end in the process. The other option would have lead much further and would reconnect to a main road. In its current form, the algorithm would be unable to find a reconnection, as none exists and would need to truncate the whole dead-end path. Another option would be to truncate the dead-end step by step (i.e. node for node) and check for new reconnecting paths after each step. This could lead to a higher success rate regarding the percentage of successfully reconnected paths.
- An entirely different, but potentially better option could be to again resort to strokes. As explained before, the concept of strokes, as discussed in Section 4.3.2, was purposefully not used during the development of the reconnection algorithm, as it was assumed that most dead-ends were created as a result of the stroke algorithm in the first place, hence resorting again to strokes would have been unwise. An option which involves strokes could be to either use much less strict constraints (i.e. a higher angle threshold and less tight requirements regarding the road classes) for the purpose of finding a suitable reconnecting path or to use the same constraints but ignore the first segments after the dead-end.

6 Results and Discussion

This chapter reports on the final results of the improved approach. Figures 6.1 to 6.4 show the final result of the four test areas and Table 6.1 lists several key figures, similar to the results of the basic approach in Chapter 4. In the following section, the results will be discussed regarding the constraints listed in Section 3.3.2. First, it will be discussed whether the hard constraints could be fulfilled or not (mainly based on the key statistics in Table 6.1). Afterwards, the fulfillment of the soft constraints will be discussed based on a qualitative evaluation conducted by swisstopo cartographers. In the final section of this chapter, the research questions of Section 2.3 will be answered.

Note that while several authors compare the results of their selection algorithms with a reference dataset (e.g. Yang et al., 2011; Li and Zhou, 2012), such a comparison can be rather problematic (Bard, 2004). Because the generalized reference dataset has been produced manually, it is highly dependent on the cartographer's experience and subjective views (Zhou and Li, 2012). Touya (2010) specifically mentions that if the result is compared to a reference dataset, it is difficult to tell whether the differences are caused by an imperfect process, uncertainties in the specification's translations or errors in the reference dataset. In the specific case of VECTOR200, another difficulty lies in the fact that it represents a fully generalized road network, on which all generalization operators have been performed. This includes displacement, smoothing and a reclassification of the road types. Nevertheless, the VECTOR200 extracts of the same four test regions are shown in Appendix C, as they can be used to compare the overall density of the road network with the results of the basic and improved approaches.

6.1 Results

Table 6.1: Key figures of the final result using the improved approach for the four test areas. Dead-ends at the edges of the test areas were excluded from the analysis, as they are an artefact of the extraction of the different areas and not of the selection algorithm.

	Davos	**Lucerne**	**Langenthal**	**Zurich**
SR[1]	2’496 (25.6%)	4’603 (17.5%)	6’074 (19.4%)	10’091 (15.3%)
HS[2]	0	355	87	853
HS missing	0	0	0	0
HL[3]**[m]**	0	112’580.6	42’797.0	257’112.2
HL missing[m]	0	0	0	0
HE[4]	0	46	16	95
HE missing	0	0	0	0
HE disconnected	0	0 (0)	0 (0)	1 (5)
DP[5]	0	0	0	0
DE[6]	0 (1)	0 (0)	0 (0)	1 (6)

[1] Segments Retained
[2] Highway Segments
[3] Total Highway Segment Length
[4] Highway Exits and Entries
[5] Disconnected Parts
[6] Dead-Ends

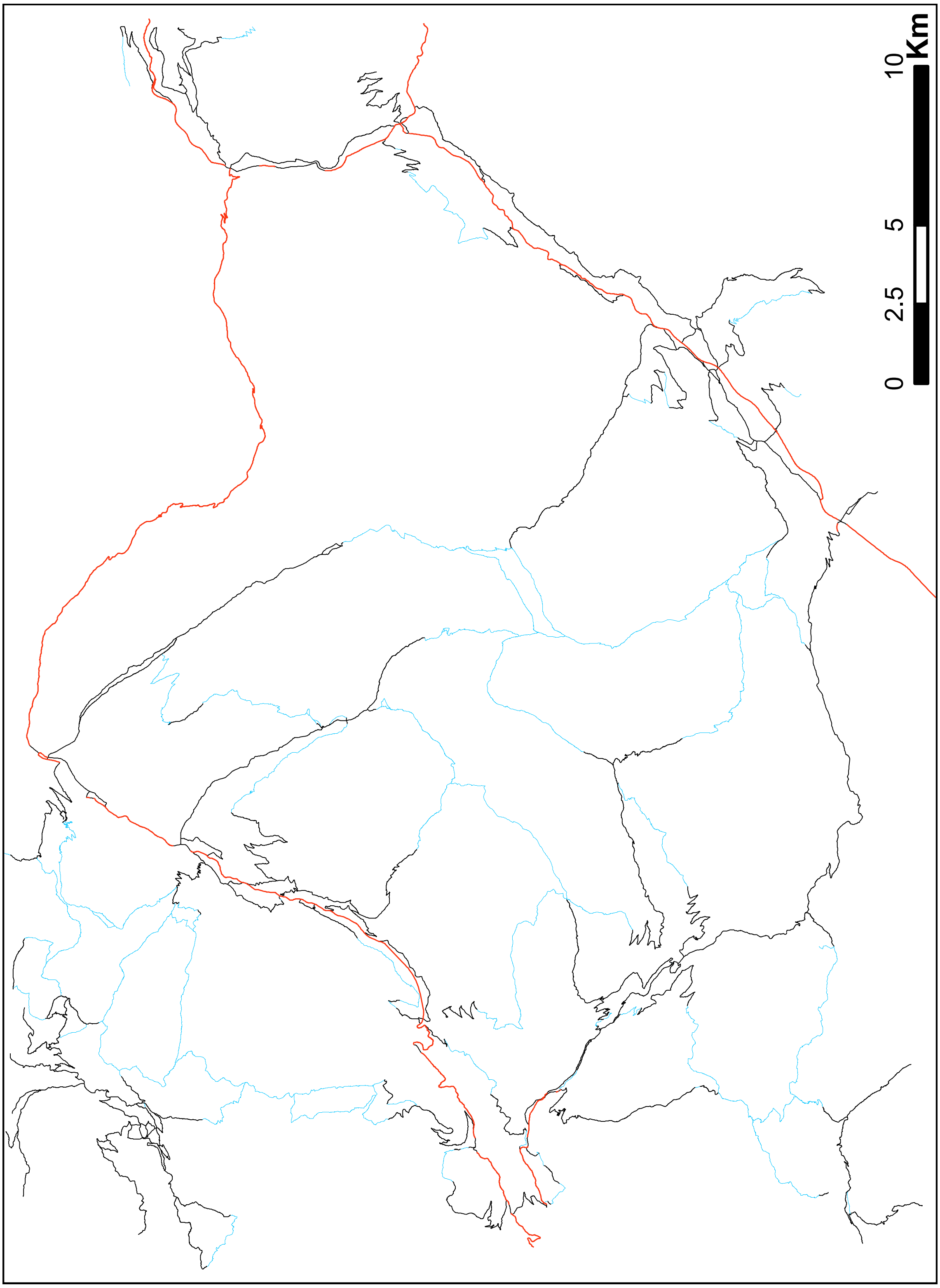


Figure 6.1: Final result of the improved approach for the Davos test area. Major roads (> 4m) are marked as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

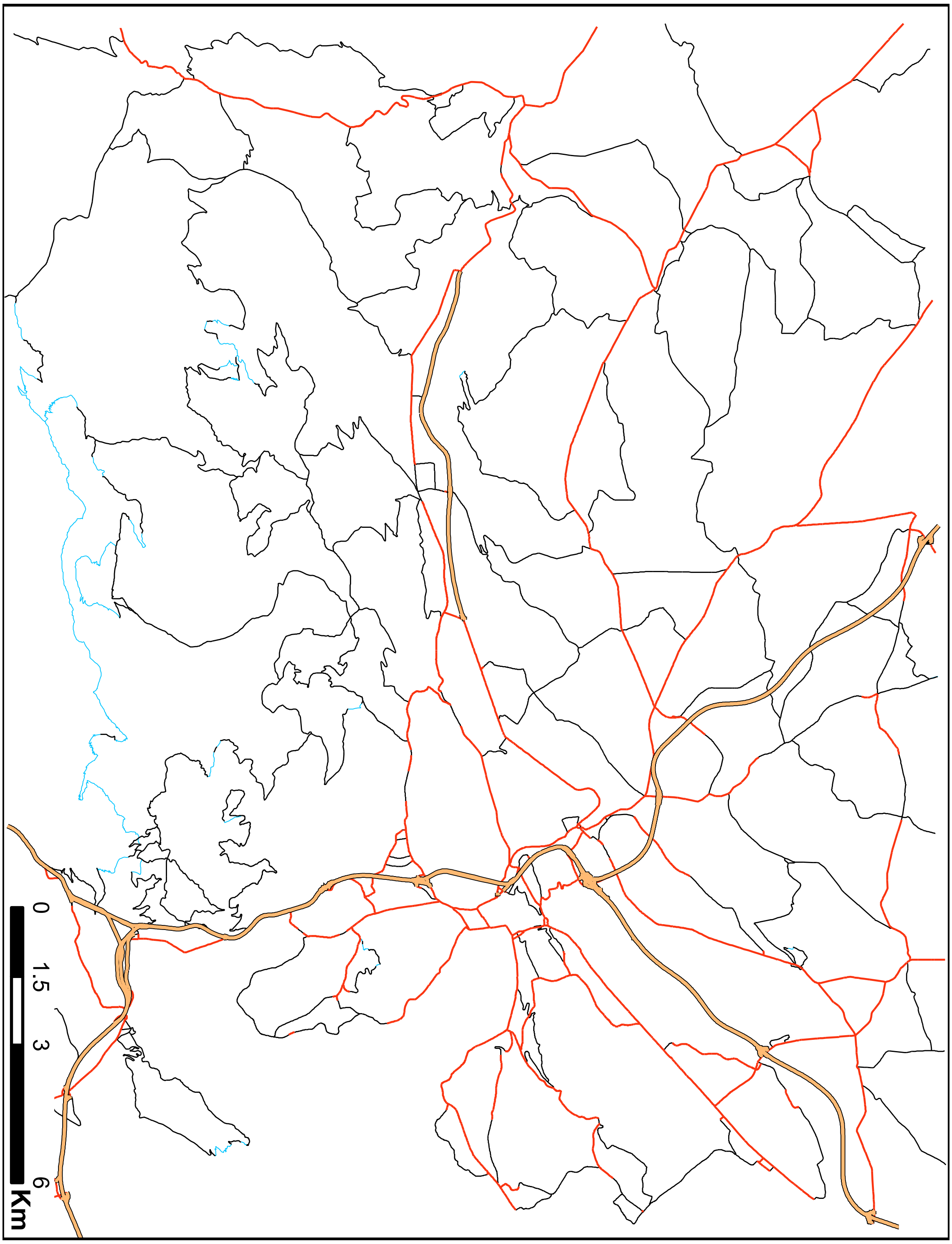


Figure 6.2: Final result of the improved approach for the Lucerne test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

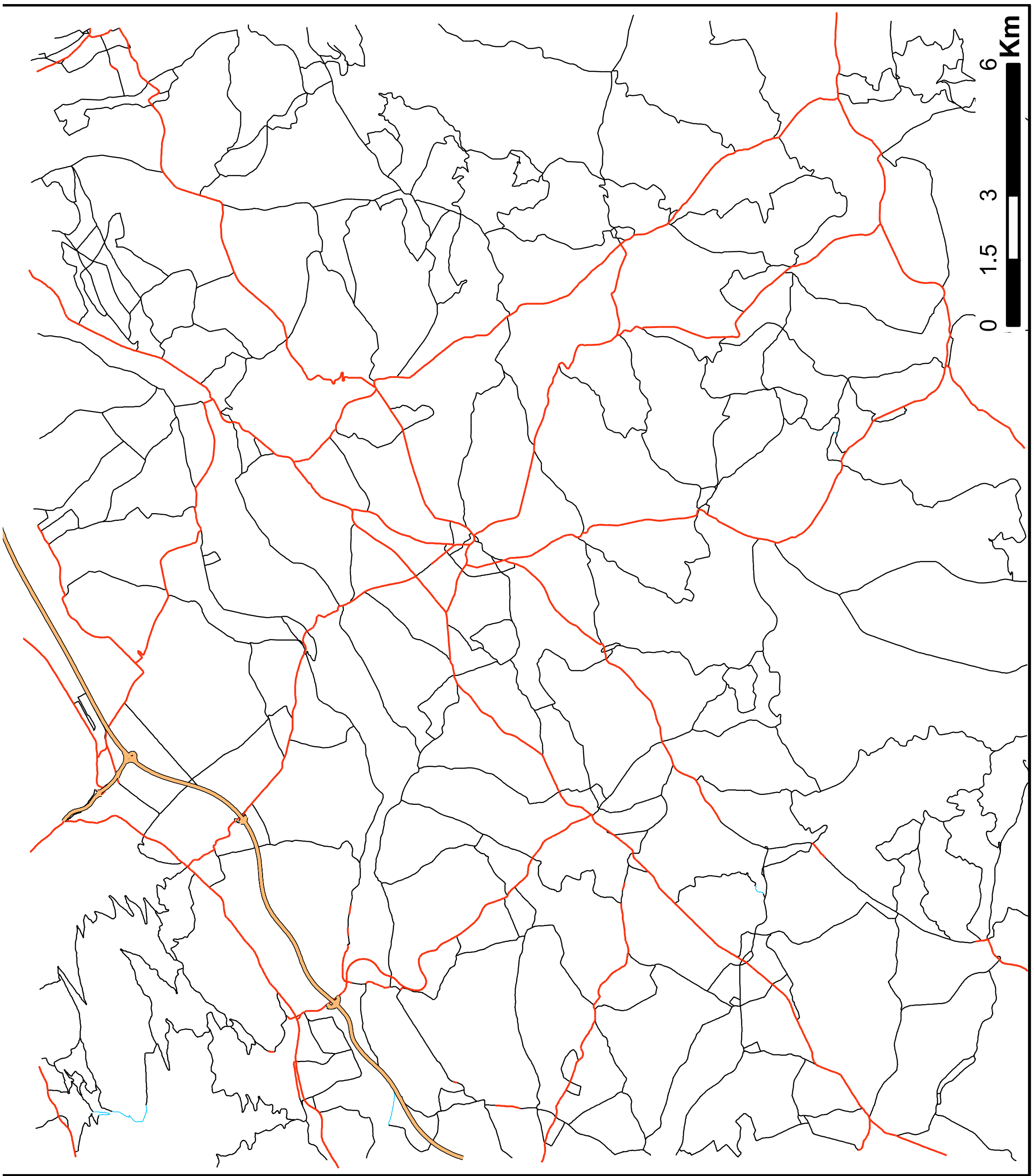


Figure 6.3: Final result of the improved approach for the Langenthal test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

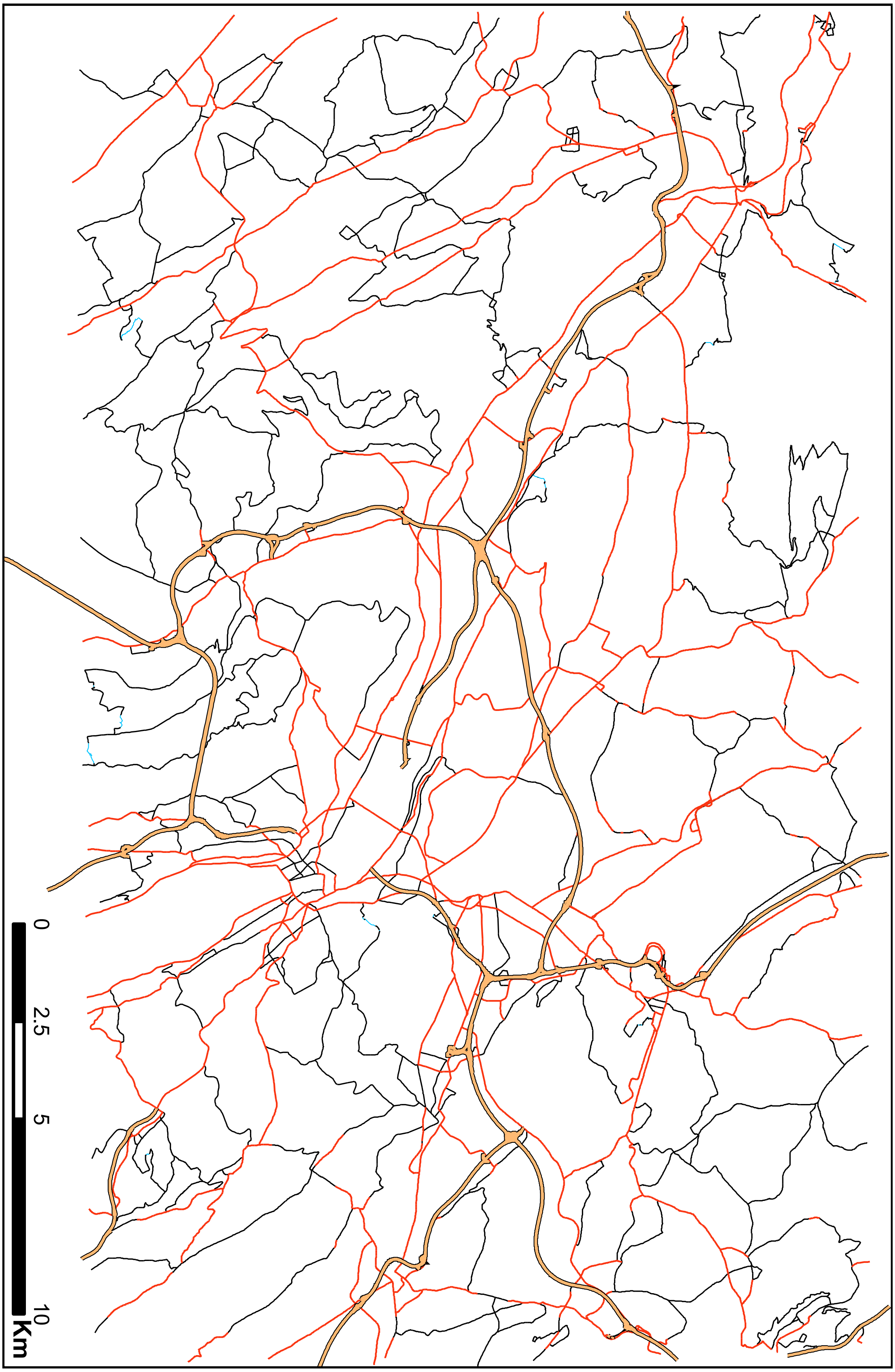


Figure 6.4: Final result of the improved approach for the Zurich test area. Highways are marked as orange wide lines, major roads (> 4m) as red lines, minor roads and paths as black lines (2-4m) and trails (< 2m) as light blue lines. (© swisstopo)

6.2 Discussion

6.2.1 Analysis of the Hard Constraints

The analysis of the hard constraints discussed in Section 3.3.2 can be done by looking at the key figures listed in Table 6.1. In the following, each of the four hard constraints will be discussed in detail for each of the test areas. One exception is the test area of Davos, where no highway segments exist, which is why two of the hard constraints do not apply to this area.

Constraint 1: All highways and expressways and everything related to them (such as entries and exits) must be included in the generalized map.

The first constraint guarantees that all highways are included in the final product, as they are very important features in a map of a scale of 1:200,000. In the basic approach, many highway segments were missing from the result as they do not always exhibit a high betweenness centrality value and are not selected as a result (e.g. all highway segments were missing in the Langenthal dataset). In the improved approach, it was possible to include all segments in all of the four test regions. Thus, this constraint could be fulfilled satisfactorily in all of the regions.

Constraint 2: The entries and exits to and from the highways must be directly connected to the road network.

The second hard constraint ensures that all exits and entries to the highways are also connected to the rest of the road network. It is rather easy to simply include all highway ramps (see analysis of Constraint 1) in the final result, but the real challenge is to also connect them to the road network such that they are also actually accessible.

In the first test area which contains highways, Lucerne, all of the entry- and exit-ramps have been successfully connected to the main network. Visual inspection also reveals that all of the connections were inserted in a sensible way, as Figure 6.5 shows in an example.

In the region of Langenthal, which only contains a relatively small amount of highways, a very similar picture is revealed. All of the entries and exits have been successfully reconnected with reasonable paths consisting of major or minor roads leading to the main road network.

While the Zurich area also improved massively compared to the basic approach regarding the connectivity of highway ramps (see Table 4.2 in Section 4.4.1), some problems could be detected (shown in brackets in Table 6.1), as one disconnected ramp persists. One example is shown in Figure 6.6, where the solution of the improved approach is shown on the left. First of all, one exit is reconnected, but not in a good way (it does not preserve the continuity very well). Secondly, another exit, which is shown as a small stubble in the middle, is not reconnected at all. The reason for this most probably lies in the angle criterion of the reconnection algorithm (see Section 5.6), which limits the possible reconnecting path, which is why none was found. While one may think that there exist better reconnections (for example to the major road to the right or in the bottom of the snippet), this impression is elusive. The reason for this are railway tracks which pass right through the middle of the snippet, separating the upper from the lower part of the road network. One possible better reconnection, marked in green, is shown on the right map. However, it continues more to the right, as it

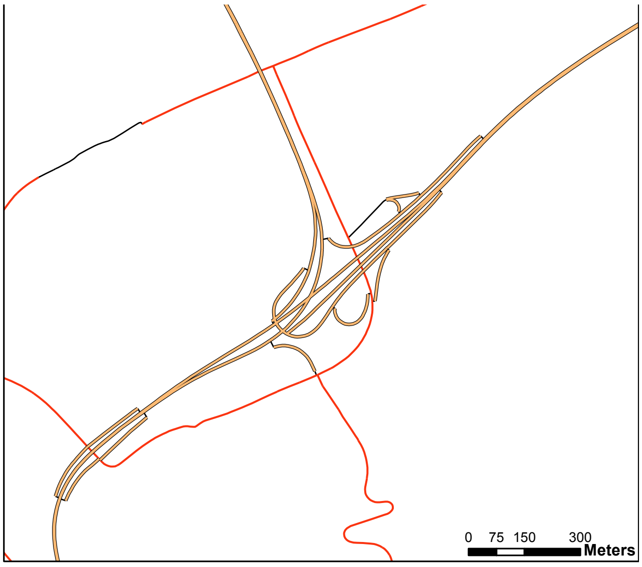


Figure 6.5: Example of the successful reconnection of highway entry- and exit ramps of a complex intersection in the Lucerne dataset. (© swisstopo)

does not connect to the major road (it is an underpass). This example shows that the reconnecting algorithm could probably be further improved. As discussed in Section 5.6, very small dead-ends (up to 400 m) were excluded from the algorithm as they pose some problems to the angle criterion. Exceptions were made for highways, however, which is why the algorithm shows a bad performance in this case. A solution could be to remove the angle criterion for such cases. The other problem, where ramps are only connected insufficiently, could be solved by revising the filtering of the reconnecting paths such that the path shown in green would be included in the result. In addition, instead of simply checking whether a ramp is connected or not, the algorithm could analyze the already existing path and find a more suitable one if the existing one does not meet the criteria.

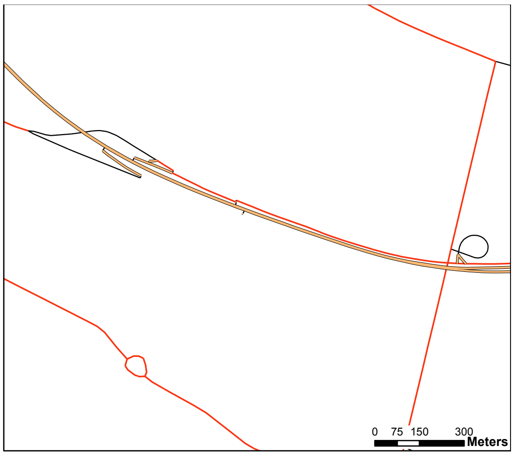
Constraint 3: The road network must be completely connected.

A very important constraint is the necessity of a completely connected network. This constraint could be fulfilled for all of the four test areas, as only one graph component (see Section 4.3.1) remains. This could be achieved by the reconnection algorithm, which was able to reconnect disconnected components.

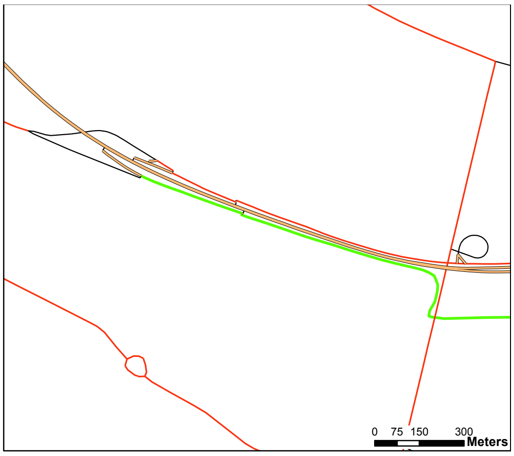
Constraint 4: No dead-ends may exist in the final result.

The dead-end problem (discussed in Section 4.4.3) could be removed nearly completely in a strict sense. The only dead-end in a strict sense (if the dead-ends in the edge-regions are excluded as they are a result of cutting out the test areas) which persists can be found in the Zurich region and was already shown in Figure 6.6. The reconnection algorithm either reconnected or removed all dead-ends, except the ones which are part of highways. Because one highway exit could not be reconnected, the dead-end still exists in the final result. This should be fixed by revising the reconnection algorithm, as discussed in the analysis of Constraint 2.

The other problem, however, are dead-ends in a broader sense: namely, dead-ends which end in a loop, as pictured in Figure 6.7. The Figure shows an example of the Davos area, where no actual



(a)



(b)

Figure 6.6: (a) A snippet of the result of the Zurich test area shows two highway exits which are either connected in a bad way (upper left) or not connected at all (small stubble in the middle). (b) A possible reconnecting path is shown in green. Note that the path does not reconnect to the major road on the right, as the path goes under it and continues a bit further to the right. (© swisstopo)

dead-end in a strict sense exists (see Table 6.1), but if looked at in a broader sense, one dead-end road can be detected which ends in a loop. While the dead-end algorithm would not detect this as a dead-end (i.e. a segment which is only connected at one end), the human eye immediately identifies this as a dead-end road which is not really wanted in the final result. There exist several possibilities to avoid this kind of dead-ends. The problem could be fixed by improving the stroke generator: for example, by prohibiting self-intersecting strokes. This would effectively eliminate such dead-end loops. However, it would also require further changes to the stroke algorithm itself, as one would need to be able to *backtrack* through the strokes if a self-intersection is detected. Another possibility is to make use of the already existing roundabout-detection algorithm. Only little changes would be required to be able to also detect these kind of loops. Smaller loops could be collapsed and larger ones could at least be detected.

In general, this constraint is largely fulfilled, with only small exceptions in Davos and Zurich. The dead-ends in a broader sense in Zurich are mostly located at highway ramps, which were not reconnected in a good way. Again, a revision of the reconnection algorithm could eliminate these problems.

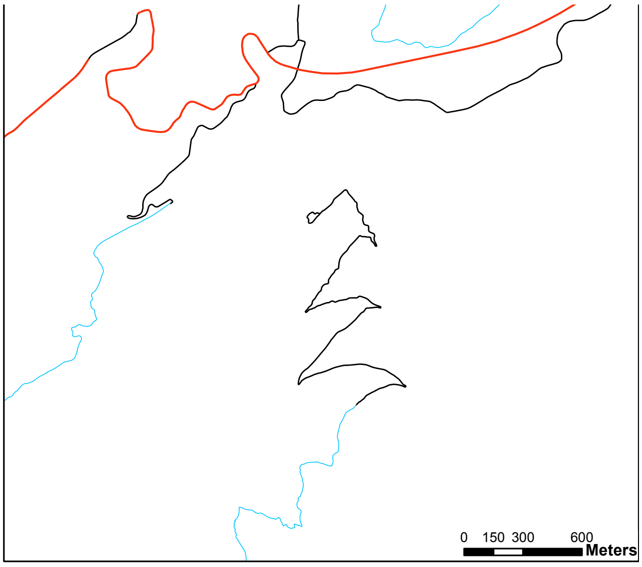


Figure 6.7: Dead-end in a broader sense in the Davos test area. (© swisstopo)

6.2.2 Qualitative Analysis of the Soft Constraints

To assess the final results regarding the soft constraints, a qualitative evaluation was conducted by two swisstopo cartographers. The cartographers were asked to fill out a questionnaire consisting of 6 questions, listed in Table 6.2. To evaluate Constraint 5, a total of three questions were used. The full questionnaire can be seen in Appendix D.

Table 6.2: Questions posed to the swisstopo experts in the qualitative assessment.

Related Constraint	**Topic**	**Question (translated)**
5	**General thinning**	How do you judge the general thinning of the road network for the target scale of 1:200'000?
5a	**Thinning in dense areas**	How do you judge the thinning in denser areas (e.g. urban areas)?
5b	**Thinning in less dense areas**	How do you judge the thinning in less dense areas (depending on the region either rural or mountainous areas)?
6	**Structure preservation**	How do you judge the general preservation of the structure in the final result?
7	**Recognizability of urban areas**	How well are urban areas recognizable in the final result?
8	**Link roads**	How do you judge the preservation of link roads?

For each of the questions, which were based on the soft constraints, they were asked to rate them according to the rating scheme of Ehrliholzer (1996):

- Good
- Acceptable
- Bad
- Unusable

As already mentioned in Section 5.4, a total of three possible solutions were produced for each of the test areas. To determine the best solution, and hence the best threshold values, the opinion of the experts was considered. Only the questionnaire results of the best solutions are presented here. In addition, the experts were asked to highlight the most problematic zones or features in each of the test regions. In the following, each of the test regions will be analyzed in turn and the results of the questionnaire presented, which enables us to also evaluate the fulfillment of the soft constraints.

Davos

General thinning The general thinning of the road network was rated as good by the cartographers of swisstopo.

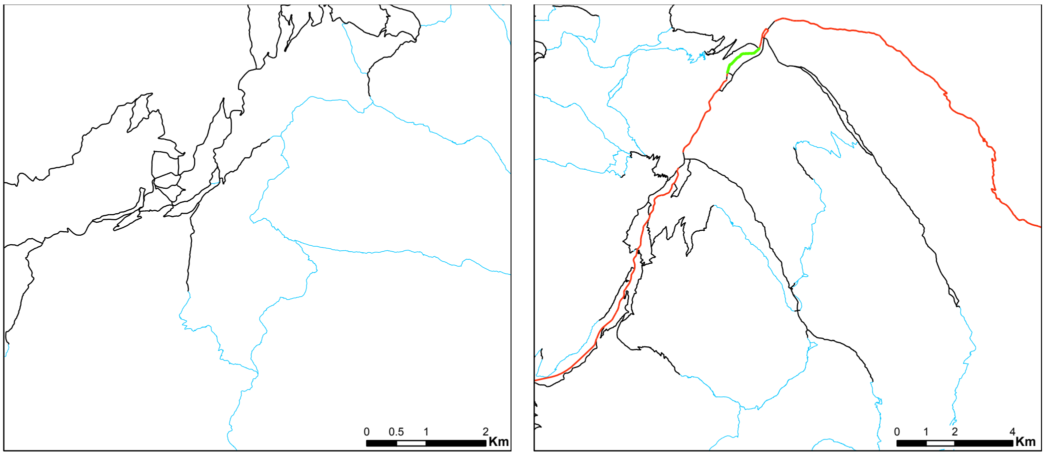
Thinning in dense areas The sparse dense areas in the Davos region were criticized the most. The region around the town of Arosa remained too dense in the end result, which resulted in a cluttered road network (see Figure 6.8(a)).

Thinning in less dense areas The thinning in the less dense region of the mountainous area was praised on the other hand, as nearly all of the important walking trails were preserved successfully.

Structure preservation The general structure of the area was also said to be well preserved, with the exception of the Arosa region, which does not influence the whole test area a lot, however.

Recognizability of urban areas Urban areas were still recognizable (comparable to the VECTOR200 product) according to the experts.

Link roads As a final criterion, the preservation of link roads was rated. In the Davos dataset, this is mainly the major road leading around the whole area (see Figure 6.1). This was rated as very good with one exception: one larger segment was missing near the City of Davos, which is why the preservation of link roads was only rated as *acceptable* (see Figure 6.8(b)).



(a) (b)

Figure 6.8: (a) The region around the town of Arosa in the Davos test area, which should have been thinned out a lot more. (b) Overview of the important link road around the whole test area, with the only missing part near the City of Davos marked in green. (© swisstopo)

In general, the result of the Davos test area was rated as very good. The two problems which were criticized, the cluttered area around Arosa and the missing segment of the link road, did not have a major influence on the map. Regarding the constraints in Table 6.2, one can say that all constraints were fulfilled, except constraint 5a) and 8, which were rated as *bad* and *acceptable*, respectively.

Lucerne

General thinning The general thinning was rated as *acceptable* by the swisstopo experts. The reason for this lies in the rather high density of small roads, paths and trails in the lower part of the test area, which can be rated as a rural or mountainous area (see Figure 6.9). The cartographers stated that too many paths were retained, which gives the user the impression of a dense road network, which is not really the case if it is compared to the rest of the map. Because the area of too high density is quite large, it deteriorates the general impression of the map.

Thinning in dense areas The dense areas, mainly the City of Lucerne, was rated as *good*. It was said, however, that the density is rather at the lower limit.

Thinning in less dense areas As has been mentioned regarding general thinning, the mountainous part of the test area remained a bit too dense. Other than that, the experts were content with the rural areas.

Structure preservation Structure preservation was also rated as *good*.

Recognizability of urban areas While the recognizability of urban areas was rated as *good*, the experts stated that it is rather hard to recognize them. However, they said it was perfectly fine for a map of small scale and that they would not recognize the different towns and cities either in their VECTOR200 road network dataset (without the settlement areas displayed), which is why the recognizability was rated as *good*.

Link roads The link roads were again rated as *good*, as all important roads were retained successfully.

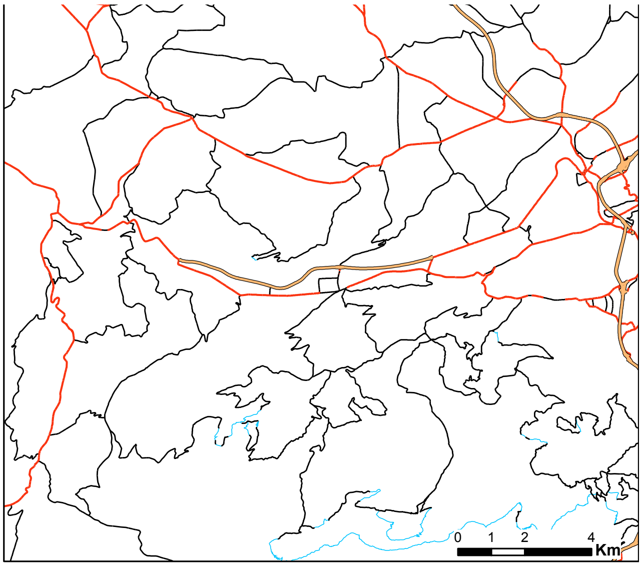


Figure 6.9: The high density of the rather mountainous area in the lower part, compared to the lower density of the rural area in the upper part, becomes evident in this snippet. The lower part should be less dense, as this would reflect the reality more closely. (© swisstopo)

In general, the Lucerne test area was rated as very good. Except for the rather high path density of the mountainous area, the experts were very content with the pruned road network. They especially praised the pruning in settlements and the link roads. Thus, all soft constraints could be fulfilled except the general thinning constraint.

Langenthal

General thinning The swisstopo experts rated the general thinning of Langenthal only as *acceptable*. The main reason for this lies in the presence of a rather high count of unnecessary roads which form loops (not by themselves). Three such examples are shown in Figure 6.10. Because they exist throughout the area, they deteriorate the general impression of the map rather significantly. In addition, it was rather difficult for the experts to decide between the map with the lowest threshold and the one with medium threshold (as explained before, they were asked to decide on the best threshold). They stated that the one having the lowest threshold yielded the best results in general, but that the density in rural areas was too high. Thus, the harmony between urban and rural areas was said to be not very good.

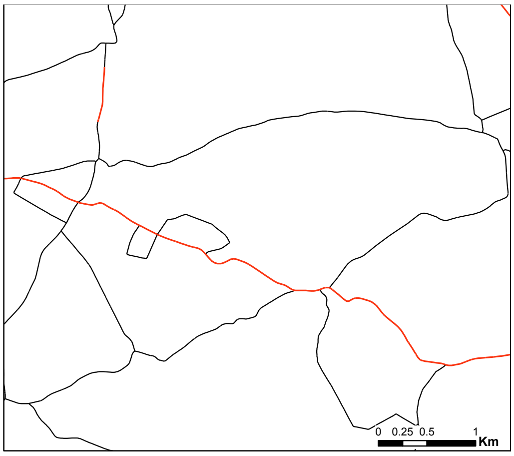
Thinning in dense areas This was rated only *acceptable* as there existed the aforementioned loops in some of the towns throughout the area.

Thinning in less dense areas As already mentioned, because the experts decided that the version with the lowest threshold yielded the best results in general, the rural areas remained too dense, which was rated as *bad*.

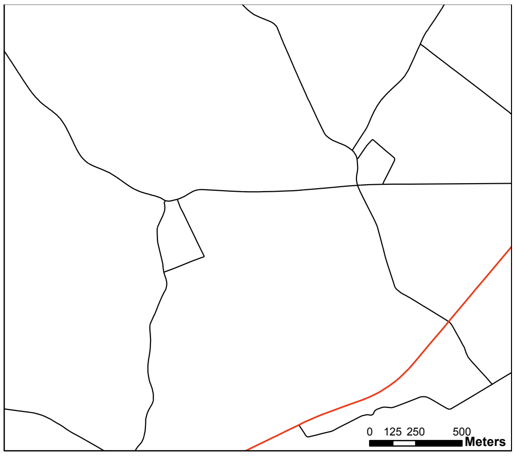
Structure preservation The general structure preservation was rated as *good*.

Recognizability of urban areas This was rated again as *good*, as the urban areas are clearly recognizable (e.g. Langenthal).

Link roads The preservation of link roads was rated as *acceptable* as one smaller road was missing.



(a)



(b)

Figure 6.10: (a,b) Examples of small, loop-like structures which exist throughout the final result of the Langenthal area. Because these streets are rather unimportant and deteriorate the general impression of the map quite heavily, they were rated as the biggest problem in the Langenthal area. (© swisstopo)

The Langenthal area was rated as acceptable in general. The experts stated that the smaller loops (Figure 6.10) posed the most serious problem and deteriorated the pruned road network of the whole region. Regarding the constraints, only the structure preservation and recognizability of urban areas could be fulfilled completely. The general thinning, as well the thinning of urban and rural areas could not be fulfilled satisfactorily.

Zurich

General thinning The general thinning in the Zurich test area was rated as *acceptable*. The main problem in this area were the urban areas, which make up most of the region.

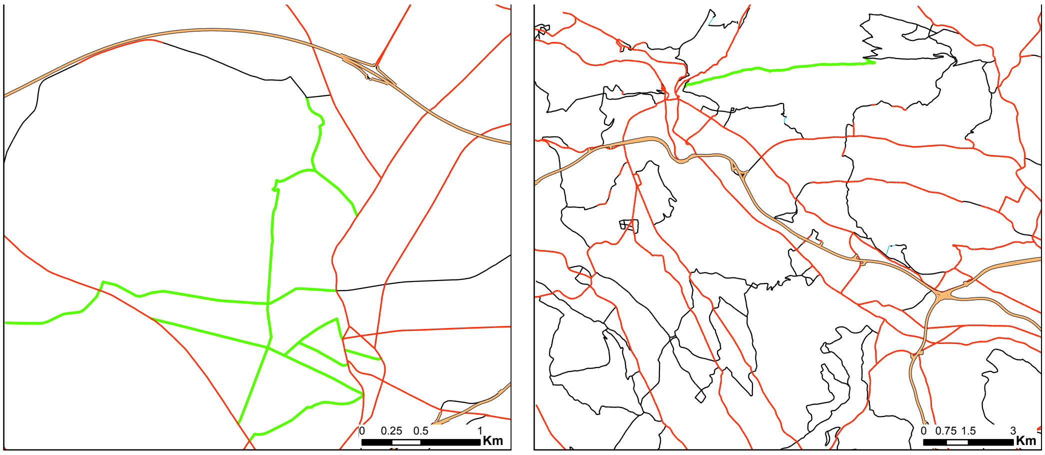
Thinning in dense areas The region around the City of Zurich was rated as rather *bad*, as many important roads were missing. An example is shown in Figure 6.11(a), where a suburb of Zurich is shown. The green lines mark the roads which were missing in the final result. Because this is not the only example and the density in the urban areas, especially around Zurich, was too low, this criterion was rated as *bad*.

Thinning in less dense areas The thinning in the rural regions was rated better. Some roads, especially very minor ones and footpaths were missing, but were rated as quite important by the swisstopo experts. An example is shown in Figure 6.11(b), where the missing footpath is marked as a green line. This is quite an important and well-known hiking trail in the area (the *Jura-Höhenweg*). While the main reason for the missing trail is that footpaths were excluded in the Zurich dataset anyway, as they were rated as too unimportant for the target scale of 1:200’000 in the region around Zurich, the path would have also been missing if footpaths would have been included in the final result, because the they exhibit a too low centrality value.

Structure preservation The general structure preservation was rated as good, with the exception of some urban parts of the region, as mentioned before. In the broad picture, they did not deteriorate the general structure too much, however.

Recognizability of urban areas Because of the bad representation of suburban areas, the recognizability of urban areas was rated as *acceptable*.

Link roads The link roads on the other hand, were rated as very good in the dataset, as only a very small portion were missing (again, in the suburban areas).



(a) (b)

Figure 6.11: (a) Example of a suburban area in the Zurich data set where some rather important roads were missing in the final result (marked in green). (b) The prominent hiking trail, the *Jura-Höhenweg* was missing in the final result, as footpaths were excluded and the trail did not exhibit a high betweenness value. (© swisstopo)

The Zurich area was the test area which was rated the worst, mainly because the large urban area around Zurich was pruned too much. While some problems regarding the density of urban areas also existed in other regions (e.g. Davos), it is much more evident in the Zurich region. In addition, the algorithm failed to include the very prominent hiking trail. Thus, the importance of trails can not be evaluated by simply calculating centrality values, as other factors come into play.

6.2.3 Research Questions

RQ1: Is it possible to generalize the road network of the TLM dataset for a target scale of 1:200,000 in such a way that the requirements of swisstopo (listed in 3.3.2) are fulfilled?

The aim of the first part of this thesis was to evaluate the basic stroke-based centrality approach presented in Jiang and Claramunt (2004a) for the road network selection for a target scale of 1:200,000. The requirements and constraints, which were worked out together with swisstopo, were used to evaluate the acquired results.

The basic algorithm had problems retaining the very important highway segments and the highways were not easily accessible, as many entries and exits were missing. While disconnected parts exist in the result, the other main problem lied in the existence of many dead-end roads, which deteriorated the result heavily. Hence, the hard constraints were clearly not fulfilled.

Because the experts of swisstopo only evaluated the result of the improved version, a detailed analysis of the soft constraints was not possible for the basic approach. Nevertheless, several additional problems inherent to the basic algorithm, which are related to both hard and soft constraints, could be identified: the basic algorithm clearly had problems regulating the density in areas having a heterogeneous distribution in the road network. In addition, several problems of the traditional stroke approach could be revealed. Thus, the basic algorithm also had difficulties retaining important link roads and producing an acceptable density distribution across the different test regions – both criteria that are part of the soft constraints.

In conclusion, one can say that the basic algorithm clearly failed to fulfill the requirements. The main problems regarding the hard constraints lie in retaining highways, as they do not always exhibit a high betweenness centrality value and in the many dead-ends which exist throughout the test areas. Especially the dead-ends posed problems, as they were largely ignored in the reviewed literature. While Yang et al. (2011) improved upon the basic approach by reconnecting completely disconnected parts, they did not solve the problem of dead-ends. Similarly, in other stroke-based approaches, dead-ends still existed in the final results (Chaudhry and Mackaness, 2005; Liu et al., 2010; Touya, 2010). The main problems regarding the soft constraints, while not evaluated thoroughly, were the problematic pruning in heterogeneous areas and the insufficient concatenation of segments during the stroke generation. While Touya (2010) used different approaches for urban and rural areas, the aim of this thesis was to use the same basic algorithm for the whole region. Other authors only evaluated the centrality-based algorithm in relatively homogeneous regions (Jiang and Claramunt, 2004a; Yang et al., 2011). Regarding the concatenation of strokes, most authors have either used the traditional, purely geometric-based approach (Yang et al., 2011), a name-based approach (Jiang and Claramunt, 2004a) or a geometric approach with additional road type information (Touya, 2010; Zhou and Li, 2012). This also seemed insufficient, however, as the identified problems could not be completely solved by simply adding road type information in a simple way.

RQ2: If the the basic algorithm does not fully meet the requirements, how can it be improved such that it performs to specifications?

As the basic algorithm could not fulfill the requirements set by swisstopo, improvements to it were made based on the identified problems (see Section 4.4.3). These improvements and their influence on the constraints will be briefly discussed.

The first improvement seeks to improve the stroke continuity by detecting and eliminating roundabouts. As shown in Section 5.2, the very simple and efficient detection algorithm worked very well in all of the test regions. With the exception of some large roundabouts in a broader sense (squares in cities), all of them could be detected and collapsed. As a result, the continuity of strokes could be improved, which helped to produce an uninterrupted network of important link roads. While other authors have implemented algorithms using meshes (Touya, 2010) or clustering algorithms (Mackaness and Mackechnie, 1999; Yang et al., 2011) it is unclear from the literature how well they performed. As the implemented algorithm is very fast, detected and collapsed all roundabouts in a strict sense (with no false positives) and improved the continuity of link roads, this improvement can be considered as a very good addition to the basic algorithm.

The second improvement has the same aims as the first one: to increase the continuity and correctness of strokes. By changing the rules by which strokes are concatenated, more correct strokes could be generated. While other authors have also included the road type as an additional rule (e.g. Touya, 2010) and it has been shown that the addition of the road type can increase the accuracy (Zhou and Li, 2012), the rules used so far in the literature were relatively simple. By increasing the complexity and also taking into consideration type groups (and not only the exact type), as has already been suggested by Thomson (2006), more accurate strokes could be produced, which again helped to not only retain more link roads, but also produce more accurate strokes and thus better centrality values.

The third improvement, the adaptive threshold based on road density, offers an option to regulate the density separately for areas which feature a dense road network in the original dataset, leading to slightly biased centrality values. This inhomogeneous pruning of the road network using the basic centrality approach has been identified as a problem, which has not been tackled by other authors using the stroke-based centrality approach, mainly because the algorithm was only tested in areas with a relatively homogeneous road network (Jiang and Claramunt, 2004a; Yang et al., 2011). While the addition of this algorithm improved the results of all test areas, there still existed problems regarding the density of urban areas (mainly in the Davos and Zurich data sets). The problems regarding the parametrization of this algorithms have also been mentioned in Section 5.5, which is why the results of this improvement are rather mixed.

The fourth and last improvement handles the reconnection of dead-ends and disconnected parts. As is evident from the results of the basic approach (Section 4.4), dead-ends in particular pose a serious problem. While dead-ends could already be reduced by eliminating roundabouts and using an improved stroke-generator, the reconnection algorithm was able to reconnect a large portion of them. As the rest of the dead-ends were eliminated, the final result of the improved approach is completely free of dead-ends. This can be seen as a big improvement compared to the basic approach and to the stroke-approach in general. While other authors were also aware of the problems that dead-ends generate, only completely disconnected parts were handled (Yang et al., 2011), which proved to be only a minor problem in the four test areas used in this thesis.

In general, both hard and soft constraints could be improved massively using the introduced enhancements. The four test regions were rated differently, however: Davos and Lucerne could fulfill most of the constraints very well and were rated as very good. Langenthal and Zurich on the other hand, had especially problems with the density in urban areas, which is why they were rated worse. The problems can be attributed to the very complex road network structure in urban areas, where the stroke-based centrality approach has difficulties to extract the most important strokes. Furthermore, as already mentioned previously, the adaptive density algorithm proved to produce results of varying quality. Improvements can especially be made by exploring the viability of additional parameters during the stroke concatenation, a way to further enrich the betweenness calculations (for example by using a weighted version of the betweenness algorithm, as described in Brandes, 2008) and by improving the adaptive density algorithm.

7 Conclusion

Because the ultimate goal of many NMAs is to be able to derive all smaller-scale maps from a single, detailed database, map generalization in general and road network selection in particular have been of high interest in the recent past (Beard, 1987; Foerster et al., 2010). In order to save time and costs, the NMA of Switzerland, swisstopo, is interested in an approach to automatically thin out the road network in their large-scale topographic model (TLM3D) for small-scale maps.

The automation of the selection of road networks has been an important part of GIScience research for the last few decades. The selection of road networks is considered to be a very complex process, as one has to deal with a coherent network of roads, each of them having a certain shape, angle, orientation and length (Chaudhry and Mackaness, 2005). As a result, many different approaches to this problem exist and no single algorithm has been able to establish itself. Moreover and very importantly, many algorithms have not been tested for small-scale maps and in particular not for large datasets.

To evaluate whether an existing approach is able to fulfill the requirements of swisstopo for their small-scale maps, the approach presented in Jiang and Claramunt (2004a), which is based on the concept of centrality of strokes (Freeman, 1977; Thomson and Richardson, 1999) was implemented and tested on several extracts of the TLM3D data set. While the results of the basic approach looked promising, they did not meet all of the requirements, which is why the existing problems and the underlying issues of the approach were thoroughly analyzed. Based on the resulting findings, several improvements to the basic approach were developed and evaluated.

7.1 Achievements

In this thesis, the following points have been achieved:

1. Analysis and discussion of the requirements and constraints to which a road network for a target scale of 1:200,000 should adhere.
2. Implementation of the stroke-based centrality approach based on Jiang and Claramunt (2004a) with the following properties:
    - For the stroke building process, the traditional geometric-only approach described in Thomson and Richardson (1999) and Zhou and Li (2012) was used instead of a name-based approach.
    - The betweenness centrality calculation was implemented using the faster algorithm of Brandes (2001).

- By exploiting the properties of this algorithm, a multithreaded implementation could be realized which utilizes an arbitrary number of threads.

3. Evaluation of the efficiency of the basic approach on regions with varying characteristics, as it is a crucial aspect that the algorithm performs well on different road networks. The approach was tested on:

   - a purely mountainous region;
   - a rural region with multiple towns;
   - a mixed region containing mountainous, rural and urban areas; and
   - an urbanized region.

4. Evaluation of the basic approach, which revealed several problems inherent to the basic algorithm:

   - Because of the usage of the purely geometric approach of the basic algorithm, the generated strokes split up unnecessarily at important intersections. At the same time, many incorrect concatenations exist, which connect road segments of very different types.
   - An analysis of the source data set of TLM3D revealed that a multitude of roundabouts exist in three of the tested regions, which cause the strokes to be split up at even more intersections. Hence, the stroke network, which crucially serves as a basis for the centrality calculation, exhibits many poor characteristics.
   - The basic approach is not able to guarantee a completely connected network, which causes potentially important roads to be disconnected from the main network.
   - The existence of many dead-ends in all of the test regions was identified as an even more extensive problem, which was also confirmed by swisstopo experts.
   - While the selection based on betweenness ccentrality was able to retain many important roads, it failed to retain all highways, which are a crucial part of the road network, especially on a small-scale map. This also included many highway ramps, which are either missing or not connected to the main network.
   - Because the purely stroke-based approach does not have a concept of road density, many urban, but also rural regions, retained a too dense network in the pruned result.

5. As the core contribution of this work, the following improvements to the basic approach were developed, based on the above findings:

   - A stroke-based algorithm, which is able to detect and collapse roundabouts, reducing the number of conflicts during the stroke building process.
   - An improved stroke generator, which utilizes additional road type information from the road network by using a custom set of road classes, resulting in a multi-step hierarchy for the concatenation of strokes.
   - A density-based adaptive centrality threshold, which uses a DBSCAN algorithm to find and thin out dense regions, leading to a more balanced road network.

- A reconnection algorithm, which is able to reconnect dead-ends and disconnected parts, by using information of road type, length and orientation.

6. Finally, a thorough quantitative and qualitative evaluation was conducted, which showed the superior cartographic performance of the improved approach. The qualitative evaluation was carried out by swisstopo experts with extensive experience in cartography.

7.2 Insights

The stroke-based centrality approach has shown promising results in earlier work (Jiang and Claramunt, 2004a; Yang et al., 2011). The aim of this thesis was to elaborate whether the approach can be used on the large-scale TLM3D dataset of swisstopo to prune the road network for a small-scale map with a target scale of 1:200,000. Because the approach should work on different kinds of road networks, it was evaluated on four very different test areas.

While the basic approach of Jiang and Claramunt (2004a) clearly exhibited many problematic attributes, it was still relatively successful in retaining many of the important roads, which is why the decision was made to improve upon the basic framework of the approach by developing several enhancements. The introduced roundabout elimination algorithm was presented as a first improvement with the objective to improve the stroke accuracy and to reduce the number of split up roads. The algorithm proved to be very effective, as it was able to detect all roundabouts in the four test areas with the exception of some special cases in Zurich.

To improve the stroke building process even further, it was shown that the stroke building can be improved by introducing several additional rules. In particular, groups of road types were generated, such that the road type is taken into account, while also ensuring that roads which belong to the same groups (i.e. major roads) but to different types, are still allowed to connect to each other. In addition, the angle threshold was adapted depending on individual situations. The new stroke algorithm was implemented in such a way that the different rules can be easily replaced, which is why a slightly adapted version was used for the mountainous test area, which accounts for the different road network characteristics present in such regions.

Because the basic stroke-based centrality approach and stroke-based approaches in general rely on a single, global threshold, it is not possible to regulate dense and less dense regions separately. Therefore, it was shown that the results can be improved by using an adaptive betweenness centrality threshold, which is based on the road density after an initial selection has occurred. The enhancement was able to reduce the density of urban areas for which too many roads were retained. As a result, a more uniform density distribution could be achieved, while at the same time maintaining the characteristics of the road network.

During the analysis of the basic algorithm, it was revealed that disconnected parts and especially dead-ends pose serious problems to which no solution could be found in the literature. Thus, a reconnection algorithm was introduced which was able to reconnect dead-ends and disconnected parts using criteria such as road type, distance and orientation. The algorithm was able to successfully reconnect many important dead-ends and thus increase the general connectivity of the road network.

Finally, the improved approach was evaluated and discussed extensively. The qualitative evaluation by swisstopo cartographers proved to be a valuable instrument to assess the final results. They were able to reveal the clear strengths of the algorithm, but also some weaknesses, which can be used as a guide for future work.

7.3 Future Work

The calculation of betweenness centrality was accomplished using an unweighted betweenness measure (see Brandes, 2001). A drawback of this approach is that it only takes the topological relationships of strokes into account and treats each stroke equally. An alternative would be to use a distance-scaled betweenness measure, which weighs all shortest paths inversely proportional to their length, also known as *length-scale betweenness* (Brandes, 2008). In addition, one could think of additionally weighting the paths according to the contained road types, such that a slightly longer major road would be preferred as opposed to a short minor road.

While the enhancements to the basic algorithm were shown to significantly improve the results, many of them could be further optimized. Especially the semantics for stroke generation and the reconnection algorithm, which heavily rely on a set of rules developed in this work, could be improved further such that even better strokes can be generated, which is essential for the correct calculation of centrality and a balanced, error–free pruning of the road network. In addition to the roundabout elimination, more structure detection algorithms could be implemented to further reduce the conflicts when generating strokes (see Heinzle et al., 2005, 2007). Despite the fact that the adapted threshold based on road density helped to improve the result, it can probably be seen as the least successful improvement, as the most criticized aspects of the results were the dense areas. While better tuning of the density threshold might help, other viable options should also be explored which make a density adaptation possible. One possibility could be to use a mesh-based approach for urban regions (e.g. Chen et al., 2009; Li and Zhou, 2012), which is able to select roads based on density.

Finally, the approach should be tested on more test areas, in order validate its cartographic performance also outside of Switzerland.

Bibliography

Bard, S. (2004). Quality Assessment of Cartographic Generalisation. *Transactions in GIS*, 8(1):63–81.

Bavelas, A. (1948). A Mathematical Model for Group Structures. *Human Organization*, 7(3):16–30.

Beard, K. M. (1987). How to Survive on a Single Detailed Database. In *Proceedings of the International Symposium on Computer-Assisted Cartography*, Baltimore, USA, 29 March - 3 April.

Beard, K. M. (1991). Constraints on Rule Formation. In Buttenfield, B. P. and McMaster, R. B., editors, *Map Generalization: Making Rules for Knowledge Representation*, pages 121–135. Longman Group.

Benz, S. A. (2013). Algorithmen zur automatischen Selektion von Strassennetzwerken: Lösungen für den übergang vom Topografischen Landschaftsmodell (TLM) zu Massstab 1:50'000. Master thesis, University of Zürich.

Brandes, U. (2001). A Faster Algorithm for Betweenness Centrality. *The Journal of Mathematical Sociology*, 25(2):163–177.

Brandes, U. (2008). On Variants of Shortest-path Betweenness centrality and Their Generic Computation. *Social Networks*, 30(2):136–145.

Brassel, K. E. and Weibel, R. (1988). A Review and Conceptual Framework of Automated Map Generalization: International Journal of Geographical Information Systems. *International Journal of Geographical Information Systems*, 2(3):229–244.

Chaudhry, O. and Mackaness, W. A. (2005). Rural and Urban Road Network Generalisation: Deriving 1:250,000 from OS MasterMap. In *Proceedings of the 22th International Cartographic Conference*, La Coruña, Spain, 10-16 July.

Chen, J., Hu, Y., Li, Z., Zhao, R., and Meng, L. (2009). Selective Omission of Road Features based on Mesh Density for Automatic Map Generalization. *International Journal of Geographical Information Science*, 23(8):1013–1032.

Cohen, S. and Ward, L. M. (1989). *Sensation and Perception*. Brace and Jovanovich, Orlando.

Daly, E. and Haahr, M. (2007). Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40.

Diakoulaki, D., Mavrotas, G., and Papayannakis, L. (1995). Determining Objective Weights in Multiple Criteria Problems: The CRITIC Method. *Computers & Operations Research*, 22(7):763–770.

Ehrliholzer, R. (1996). Methoden für die Bewertung der Qualität von Generalisierungslösungen. Diploma thesis, University of Zürich, Zürich Switzerland.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, pages 226–231. AAAI Press.

Foerster, T., Stoter, J., and Kobben, B. (2007). Towards a Formal Classification of Generalization Operators. In *23rd International Cartographic Conference*, Moscow, August 4-10.

Foerster, T., Stoter, J., and Kraak, M.-J. (2010). Challenges for Automated Generalisation at European Mapping Agencies: A Qualitative and Quantitative Analysis. *The Cartographic Journal*, 47(1):41–54.

Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41.

Freeman, L. C. (1978). Centrality in Social Networks: Conceptual Clarification. *Social Networks*, 1(3):215–239.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Goodrich, M. T., Tamassia, R., and Mount, D. M. (2004). *Data Structures and Algorithms in C++*. John Wiley & Sons, New Jersey USA.

Harrie, L. E. (2003). Weight-Setting and Quality Assessment in Simultaneous Graphic Generalization. *The Cartographic Journal*, 40(3):221–233.

Harrie, L. E. and Weibel, R. (2007). Modelling the Overall Process of Generalisation. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pages 67–87. Elsevier, Oxford UK.

Heinzle, F., Anders, K.-H., and Sester, M. (2005). Graph Based Approaches for Recognition of Patterns and Implicit Information in Road Networks. In *Proceedings of the 22th International Cartographic Conference*, La Coruña, Spain, 10-16 July.

Heinzle, F., Anders, K.-H., and Sester, M. (2007). Automatic Detection of Patterns in Road Networks - Methods and Evaluation. In *Proceedings of Joint Workshop Visualization and Exploration of Geospatial Data*, Stuttgart, Germany, 27-29 June.

Hu, Y., Chen, J., Li, Z., and Zhao, R. (2007). Selection of Streets Based on Mesh Density for Digital Map Generalization. In *4th International Conference on Information Science*, pages 903–908, Chengdu China, 22-24 August.

Jiang, B. (2007). A Topological Pattern of Urban Street Networks: Universality and Peculiarity. *Physica A: Statistical Mechanics and its Applications*, 384(2):647–655.

Jiang, B. (2009). Street Hierarchies: a Minority of Streets Account for a Majority of Traffic Flow. *International Journal of Geographical Information Science*, 23(8):1033–1048.

Jiang, B. and Claramunt, C. (2004a). A Structural Approach to the Model Generalization of an Urban Street Network. *GeoInformatica*, 8(2):157–171.

Jiang, B. and Claramunt, C. (2004b). Topological Analysis of Urban Street Networks. *Environment and Planning B: Planning and Design*, 31(1):151–162.

Jiang, B. and Harrie, L. E. (2004). Selection of Streets from a Network Using Self-Organizing Maps. *Transactions in GIS*, 8(3):335–350.

Jiang, B., Yin, J., and Zhao, S. (2008). Self-organized natural roads for predicting traffic flow: A sensitivity study. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(07).

Joy, M. P., Brock, A., Ingber, D. E., and Huang, S. (2005). High-Betweenness Proteins in the Yeast Protein Interaction Network. *Journal of Biomedicine and Biotechnology*, 2005(2):96–103.

Kreveld, M. and Peschier, J. (2000). On the Automated Generalization of Road Network Maps. In *Proceedings of the 5th International Conference on GeoComputation*, University of Greenwich, United Kingdom, 23-25 August.

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W. A., and Weibel, R. (1999). The Application of Agents in Automated Map Generalisation. In *Proceedings of the 19th International Cartographic Conference*, pages 160–169, Ottawa, Canada, 14-21 August.

Li, Z. and Zhou, Q. (2012). Integration of Linear and Areal Hierarchies for Continuous Multi-scale Representation of Road Networks. *International Journal of Geographical Information Science*, 26(5):855–880.

Liu, X., Zhan, F. B., and Ai, T. (2010). Road Selection based on Voronoi Diagrams and "Strokes" in Map Generalization. *International Journal of Applied Earth Observation and Geoinformation*, 12:194–202.

Mackaness, W. A. and Beard, K. M. (1993). Use of Graph Theory to Support Map Generalization. *Cartography and Geographic Information Science*, 20(4):210–221.

Mackaness, W. A. and Mackechnie, G. A. (1999). Automating the Detection and Simplification of Junctions in Road Networks. *GeoInformatica*, 3(2):185–200.

Madduri, K., Ediger, D., Jiang, K., Bader, D. A., and Chavarria-Miranda, D. (2009). A faster parallel algorithm and efficient multithreaded implementations for evaluating betweenness centrality on massive datasets. In *Third Workshop on Multithreaded Architectures and Applications*, pages 1–8, Rome, Italy, 29 May 2009.

McMaster, R. B. and Shea, K. S., editors (1992). *Generalization in Digital Cartography*. Association of American Geographers, Washington DC.

Morisset, B. and Ruas, A. (1997). Simulation and Agent Modeling for Road Selection in Generalization. In *Proceedings of the 18th International Cartographic Conference*, Stockholm, Sweden, 23-27 June.

Newman, M. E. J. (2001). Scientific Collaboration Networks: II. Shortest paths, Weighted Networks, and Centrality. *Physical Review E*, 64(1):016132.

Porta, S., Crucitti, P., and Latora, V. (2006a). The network analysis of urban streets: A dual approach. *Physica A: Statistical Mechanics and its Applications*, 369(2):853–866.

Porta, S., Crucitti, P., and Latora, V. (2006b). The network analysis of urban streets: a primal approach. *Environment and Planning B: Planning and Design*, 33(5):705–725.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581–603.

Spiess, E., Baumgartner, U., Arn, S., and Vez, C. (2002). *Topographic Maps: Map Graphics and Generalisation*. Swiss Society of Cartography, 17 edition.

swisstopo (2012b). swissTLM3D. http://www.swisstopo.admin.ch/internet/swisstopo/en/home/products/landscape/swissTLM3D.html, last accessed on 13.06.2013.

swisstopo (2012c). The Topographic Landscape Model TLM. http://www.swisstopo.admin.ch/internet/swisstopo/en/home/topics/geodata/TLM.html, last accessed on 13.06.2013.

swisstopo (2012d). VECTOR200. http://www.swisstopo.admin.ch/internet/swisstopo/en/home/products/landscape/vector200.html, last accessed on 13.06.2013.

swisstopo (2012e). VECTOR200 Dokumentation. http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/landscape/vector200.html, last accessed on 13.06.2013.

swisstopo (30.10.2012a). Projekt Optina-LK. Presentation, 30.10.2012.

Thomson, R. C. (2006). The' stroke' Concept in Geographic Network Generalization and Analysis. In Riedl, A., Kainz, W., and Elmes, G., editors, *Progress in Spatial Data Handling*, pages 681–697. Springer Berlin Heidelberg.

Thomson, R. C. and Brooks, R. (2000). Efficient Generalisation and Abstraction of Network Data Using Perceptual Grouping. In *Proceedings of the 5th International Conference on GeoComputation*, University of Greenwich, United Kingdom, 23-25 August.

Thomson, R. C. and Brooks, R. (2002). Exploiting Perceptual Grouping for Map Analysis, Understanding and Generalization: The Case of Road and River Networks. In Blostein, D. and Kwon, Y.-B., editors, *Graphics Recognition Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 148–157. Springer Berlin.

Thomson, R. C. and Richardson, D. E. (1995). A Graph Theory Approach to Road Network Generalisation. In *Proceedings of the 17th International Cartographic Conference*, pages 1871–1880, Barcelona, Spain, 3-9 September.

Thomson, R. C. and Richardson, D. E. (1999). The 'Good Continuation' Principle of Perceptual Organization Applied to the Generalization of Road Networks. In *Proceedings of the 19th International Cartographic Conference*, pages 1215–1223, Ottawa, Canada, 14-21 August.

Tomko, M., Winter, S., and Claramunt, C. (2008). Experiential Hierarchies of Streets. *Computers, Environment and Urban Systems*, 32(1):41–52.

Töpfer, F. and Pillewizer, W. (1966). The Principles of Selection. *The Cartographic Journal*, 3(1):10–16.

Touya, G. (2010). A Road Network Selection Process Based on Data Enrichment and Structure Detection. *Transactions in GIS*, 14(5):595–614.

Weibel, R. and Dutton, G. (1998). Constraint-based Automated Map Generalization. In *Proceedings of the 8th International Symposium on Spatial Data Handling*, pages 214–224, Vancouver, Canada, 11-15 July.

Wilkinson, B. and Allen, M. (1999). *Parallel Programming*. Prentice Hall, New Jersey USA.

Yang, B., Luan, X., and Li, Q. (2011). Generating hierarchical strokes from urban street networks based on spatial pattern recognition. *International Journal of Geographical Information Science*, 25(12):2025–2050.

Yaolin, L., Molenaar, M., Tinghua, A., and Yanfang, L. I. (2001). Frameworks for generalization constraints and operations based on object-oriented data structure in database generalization. *Geospatial Information Science*, 4(3):42–49.

Zhang, Q. (2004). Modeling structure and patterns in road network generalization. In *ICA workshop on Generalization and Multiple Representation*, Leicester, UK, 20-21 August.

Zhang, Q. (2005). Road Network Generalization Based on Combined Analysis of Structure and Shape. In Fisher, P., editor, *Developments in Spatial Data Handling*, pages 343–353. Springer, Berlin, Germany.

Zhou, Q. and Li, Z. (2012). A comparative study of various strategies to concatenate road segments into strokes for map generalization. *International Journal of Geographical Information Science*, 26(4):691–715.

A Software Prototype

The software prototype was implemented in Java. Table A.1 contains descriptions of all relevant classes which are used in the prototype. Throughout the development, appropriate design patterns were used (see Gamma et al., 1995), in order to make many implementations replaceable. For example, the DBSCAN algorithm was implemented in such a way that it works with any point type and distance function or a new stroke evaluator with additional rules could easily be added to the prototype. In addition, `AttributeLocator` is used such that the prototype could potentially work on any dataset with only minor adjustments.

Table A.1: Important packages and classes implemented in the software prototype.

msc.algorithms.centrality Classes used to compute centrality measures	
`CentralityBrandesMulti`	Multithreaded version of the centrality algorithm (Brandes, 2001)
`CentralityStruct`	Contains all data of the calculation (used separately by each thread)
msc.algorithms.density Contains the classes used for the adapted density algorithm	
`CentroidPoint`	Represents a centroid of a segment (input for the DBSCAN algorithm)
`DBCluster`	Represents a cluster of points in the DBSCAN algorithm (Ester et al., 1996)
`DBScan`	Implementation of the DBSCAN algorithm
msc.algorithms.loops Group of algorithms used for the roundabout elimination	
`Loop`	Represents a loop or roundabout
`LoopBuilder`	Builder class which specifies and builds loops
`LoopCollapserVisitor`	Collapses a loop to its centroid
`LoopExtractorVisitor`	Extracts loops or roundabouts from a graph
msc.algorithms.reconnection Classes used for the reconnection algorithm	
`DeadEnd`	Represents a dead-end in the dataset
`DFS`	Implementation of a depth-first-search to find reconnecting paths
`Path`	Represents a reconnecting or a dead-end path
`PathComparator`	Comparator used to filter the reconnecting paths
msc.algorithms.strokegen Classes used for the generation of strokes	
`AbstractStrokeEvaluator`	Abstract evaluator class
`AngleStrokeEvaluator`	Geometric-only stroke evaluator/concatenator
`RuralSemanticStrokeEvaluator`	Evaluator used for very rural and mountainous regions
`SemanticStrokeEvaluator`	Enhanced stroke general-purpose evaluator
`RuralSemanticPairComparator`	Comparator used by the rural evaluator
`SemanticPairComparator`	Comparator used by the general-purpose evaluator
msc.data This package contains classes related to data import, export and storage	
`DataStorage`	Handles, stores and grants access to the imported data
`MapProperties`	Contains additional information about the imported data
msc.data.shapefile Classes for shapefile handling (import/export)	
`DualGraphFeatureBuilder`	Creates features from a dual graph, ready to export
`IFeatureBuilder`	FeatureBuilder interface
`PrunedGraphFeatureBuilder`	Creates features from a graph, ready to export
`Shapefile`	Represents a shapefile
`ShapefileWriter`	Class used to export features to a shapefile

Continued on next page

Table A.1 – *Continued from previous page*

msc.exceptions Custom exceptions	
UnexpectedFeatureClassException	Thrown when an unexpected feature class is detected (e.g. in the `TLMClassTranslator`)
msc.interfaces Contains interfaces which can be used globally	
IDBPoint	Interface for a DBSCAN point (the DBSCAN algorithm can be used with other points as well)
IVisitable	Interface to be implemented from classes which are visitable
IVisitor	Interface to be implemented by visitors
msc.structures.dualgraph Contains classes related to the dual graph data structure	
CoordinateMap	Class which keeps track of all the coordinates which are contained in a node of the dual graph
DualEdge	Edge of a dual graph
DualGraph	The dual graph
DualGraphBuilder	Class which specifies and builds dual graphs
DualNode	Node of a dual graph
msc.structures.primalgraph Contains classes related to primal graph data structures	
Edge	Edge of a graph
Graph	Represents a graph containing nodes and edges
GraphBuilder	Class which specifies and builds graphs
Node	Node of a graph
msc.structures.strokes Contains classes related to stroke structures	
Pair	Represents a pair of segments (during concatenation phase)
Stroke	Represents a stroke
StrokeBuilder	Class used to generate strokes from a graph
StrokeCollection	Used to represent a collection of strokes and offers various update and retrieval methods
StrokeSegment	A segment of a stroke (wrapper of an edge)
msc.utils Utility classes are located in this package	
AttributeLocator	Specifies the the location of the attribute ID
EqualsUtil	Utility class for the `equals()` method
HashCodeUtil	Utility class for the `hashcode()` method
IDGenerator	ID generator used by various structures
TLMClassTranslator	Used to translate the 'exact' TLM class to a custom one
Utils	Various additional utility methods
msc.visitors.dualgraph Visitors for the dual graph structure	
BasicCentralitySelectionVisitor	Simple centrality selection of the basic approach
CentralitySelectionVisitor	Centrality selection of the improved approach
CentralityVisitor	Visitor that uses `msc.algorithms.centrality` classes
ConnectivityVisitor	Checks the connectivity of the dual graph and extracts all *graph components*
msc.visitors.primalgraph Visitors for the primal graph structure	
ClusterVisitor	Visitor which extracts clusters from a primal graph (uses classes of `msc.algorithms.density`)
ConnectivityVisitor	Checks the connectivity of a graph and extracts all *graph components*
DensityVisitor	In conjunction with the `ClusterVisitor`, this class extracts dense areas and the related strokes
ReconnectionVisitor	Uses `msc.algorithms.reconnection` and reconnects dead-ends
RemoveDeadEndsVisitor	Used to remove dead-ends from the primal graph
RemoveObjectClassVisitor	Removes edges of certain object classes
RemoveUnconnectedVisitor	Removes unconnected *graph components* (only keeps the largest one)

B Performance

Figure B.1 shows the performance of the algorithm in relation to the number of used threads. Shown in blue is the time it takes to run the whole algorithm in the Langenthal test area (i.e. from graph creation to the final result, without importing and exporting to and from shapefiles). The red line shows the time it takes to run the centrality calculation only. Note that this is the only part of the algorithm that is actually multithreaded. Two things can be noted:

1. The time difference between the centrality calculation and the computing the whole approach is rather small. Because the only the centrality calculation benefits from the additional threads, the difference between the two graphs is larger with 4 than with 1 thread.

2. The algorithm benefits massively from the additional threads. Using 4 threads, the whole approach is over twice as fast as compared to using 1 thread only. As can be expected, the curve gets flatter the higher the thread count gets, as a portion of the centrality algorithm was not parallelized (see Madduri et al. (2009) for a more parallelized version).

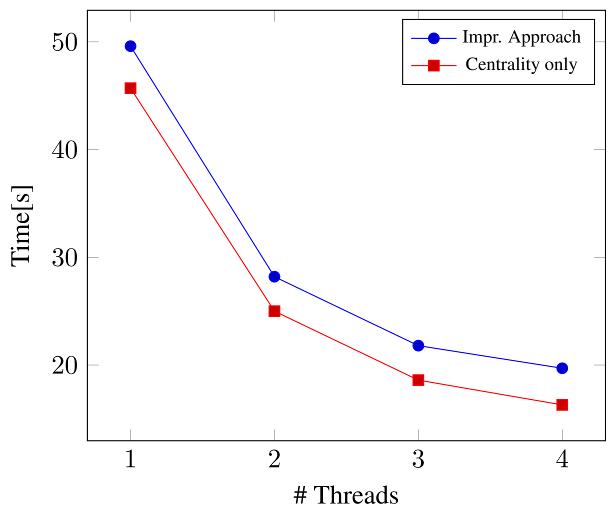


Figure B.1: Performance in relation to the used threads. Shown in blue is the time it takes to perform the whole improved approach and in red the time it takes to calculate the centrality values only (i.e. after the dual graph has been created). The test was performed on an Intel® Q9550 processor running at 3.4GHz and 8GB of RAM.

C VECTOR200

Figures C.1 to C.4 show snippets of the four test regions of the VECTOR200 dataset of swisstopo. Note that the symbolization differs from the one used in the TLM3D dataset, as the transition to a scale of 1:200,000 also includes a major reclassification of the roads. Therefore, only the highways have been symbolized. The other road and path types have not been symbolized. The extracts can be used to compare the overall density with the results from the two algorithms.

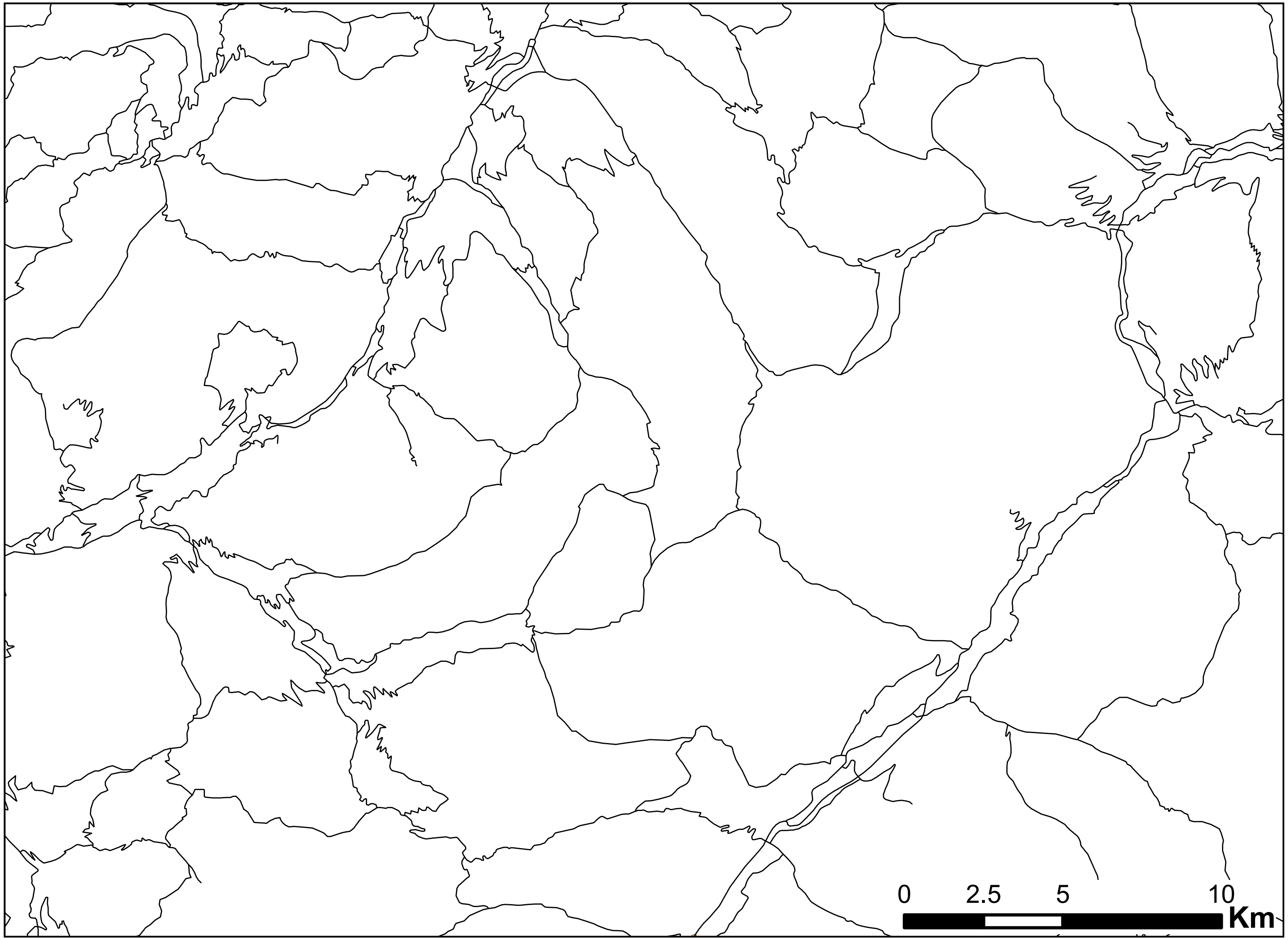


Figure C.1: VECTOR200 extract of the Davos test area. The roads have not been symbolized as the classification differs heavily from the one used in the TLM3D datasets. The snippet is intended to be used for the comparison of the overall density. (© swisstopo)

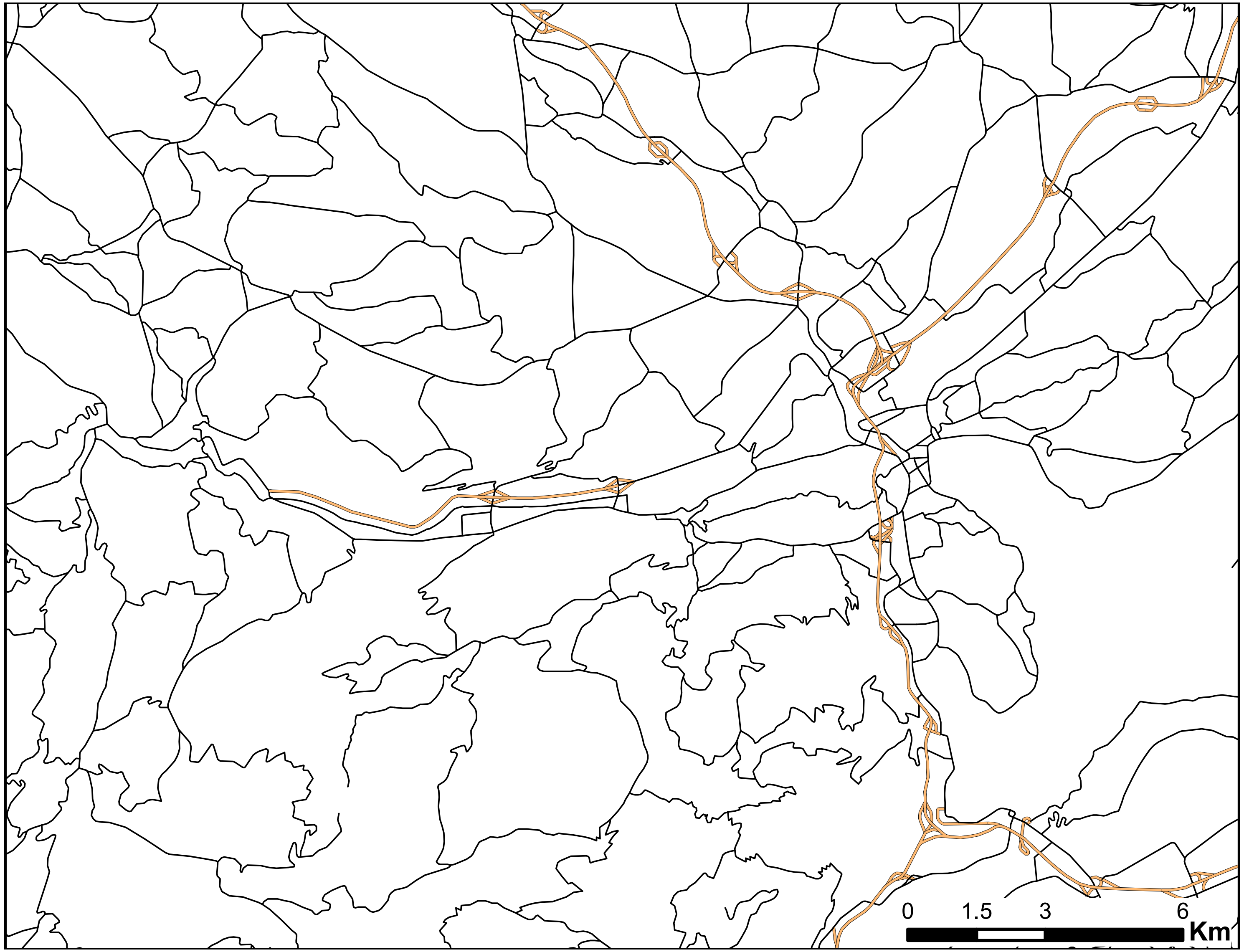


Figure C.2:]
VECTOR200 extract of the Lucerne test area. The roads have not been symbolized with the highways as the only exception, as the classification differs heavily from the one used in the TLM3D datasets. The snippet is intended to be used for the comparison of the overall density. (© swisstopo)

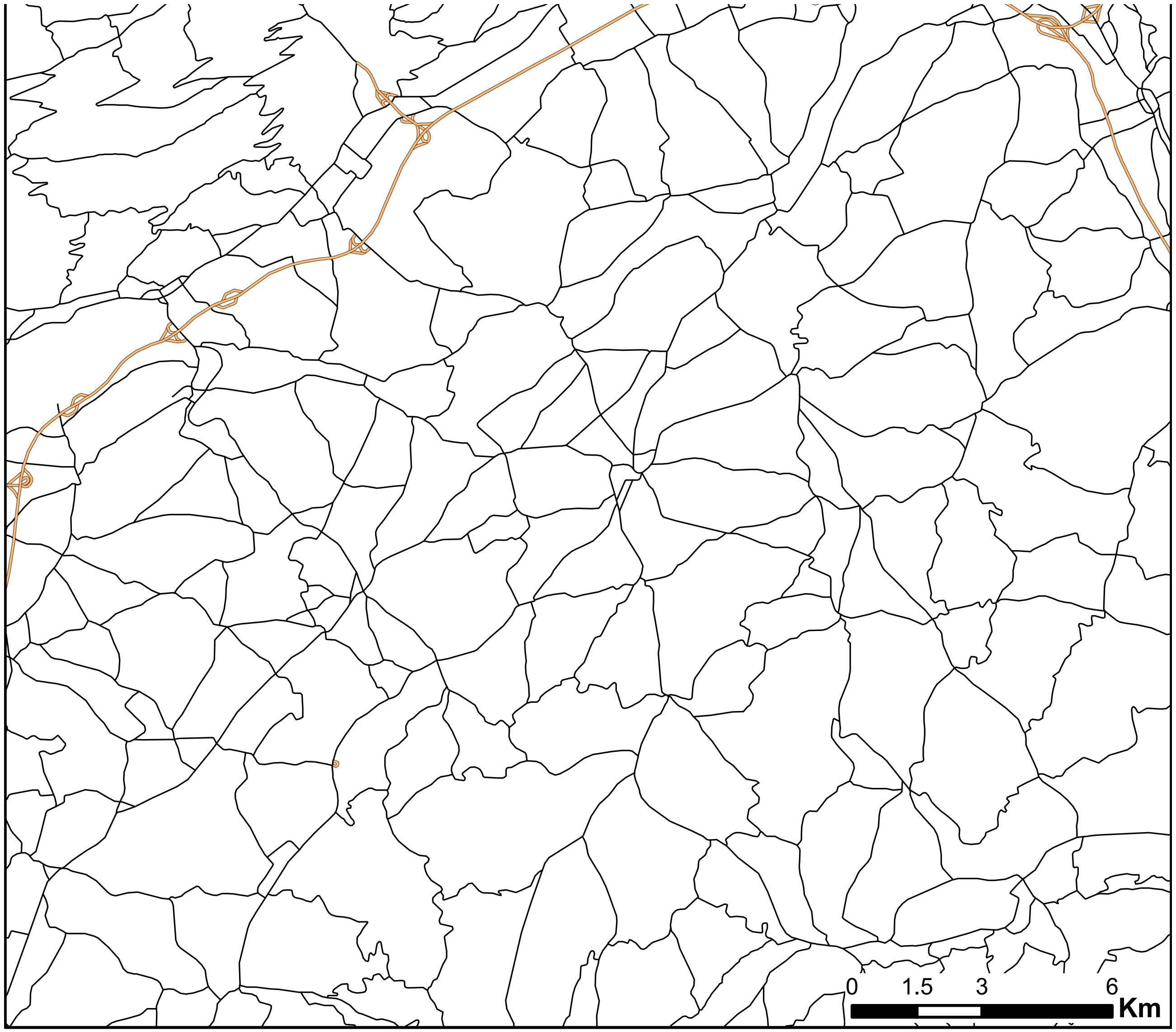


Figure C.3:]
VECTOR200 extract of the Langenthal test area. The roads have not been symbolized with the highways as the only exception, as the classification differs heavily from the one used in the TLM3D datasets. The snippet is intended to be used for the comparison of the overall density. (© swisstopo)

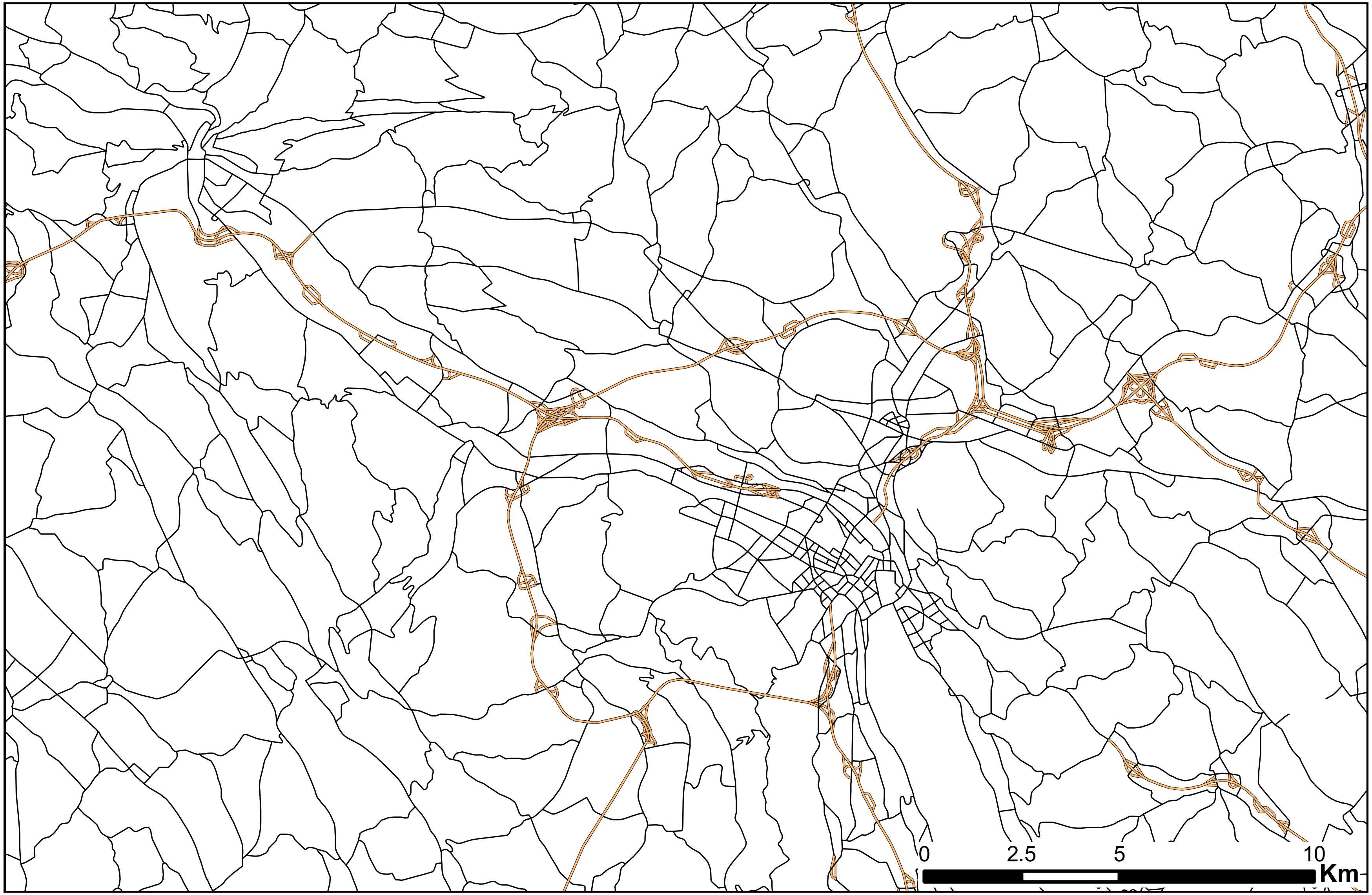


Figure C.4:]
VECTOR200 extract of the Zurich test area. The roads have not been symbolized with the highways as the only exception, as the classification differs heavily from the one used in the TLM3D datasets. The snippet is intended to be used for the comparison of the overall density. (© swisstopo)

D Questionnaire

Expertenbefragung

Einstieg:

Das Ziel meiner Masterarbeit ist die Ausarbeitung eines Algorithmus zur automatischen Selektierung von Strassennetzwerken, ausgehend vom topografischen Landschaftsmodell (TLM) für einen Zielmassstab von 1:200'000. Es umfasst nur die **Selektion** und ist deshalb kein vollständig beendetes Generalisierungsprodukt. Durch diese Expertenbefragung würde ich gerne die entstandenen Resultate von Experten qualitativ beurteilen lassen. Einerseits sollen die subjektiv-qualitativen Aspekte des Anforderungskatalogs und andererseits spezifische Beispiele in den Testgebieten evaluiert werden. Die Ergebnisse dieser Befragung sollten differenzierte Rückschlüsse über die Brauchbarkeit des Algorithmus liefern, sowie über spezifisch gelungene Bereiche und Problemzonen.

Einstiegsfragen:

Wie lange arbeiten Sie schon im Bereich der Kartographie?

____

In welcher Abteilung arbeiten Sie? Was ist Ihre Hauptaufgabe?

____

Arbeiten Sie in der topographischen oder thematischen Kartographie?

____

Wie viel Erfahrung haben Sie im Bereich der Generalisierung von Strassennetzwerken?

____

Testgebiet: ______________________

Es stehen jeweils drei Versionen mit verschiedenen Ausdünnungsgraden der Testgebiete zur Verfügung. Frageblock 1 bezieht sich jeweils auf alle drei Ausdünnungsgrade, Frageblock 2 nur noch auf das oder die am besten bewerteten. H, M und L stehen für hohe, mittlere und niedrige Grenzwerte.

Frageblock 1: Spezifische Fragen

Grad der Ausdünnung

Wie beurteilen Sie gesamthaft den Grad der Ausdünnung für den Zielmassstab von 1:200'000?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Strukturerhaltung

Wie beurteilen Sie die grundlegende Erhaltung der Struktur im Resultat?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Wie gut sind die Siedlungsgebiete auch im Resultat noch als solche erkennbar?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Wie beurteilen Sie den Grad der Ausdünnung in dichteren Gebieten (z.B. Siedlungsgebieten)?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Wie beurteilen Sie den Grad der Ausdünnung in weniger dichten Gebieten (z.B. in ländlicheren Gebieten)?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Inwiefern sind die für Sie intuitiv wichtigen Verbindungsstrassen erhalten geblieben?

H	**M**	**L**
☐ Gut	☐ Gut	☐ Gut
☐ Akzeptabel	☐ Akzeptabel	☐ Akzeptabel
☐ Schlecht	☐ Schlecht	☐ Schlecht
☐ Unbrauchbar	☐ Unbrauchbar	☐ Unbrauchbar

Kommentar:

Frageblock 2: Offene Evaluation

Gibt es im Resultat Bereiche, deren Generalisierung Sie als **besonders gelungen** bezeichnen würden? Entsprechende Zonen können direkt im Resultat **grün markiert** und begründet werden.

Gibt es im Resultat Bereiche, deren Generalisierung Sie als **problematisch** bezeichnen würden? Entsprechende Zonen können direkt im Resultat **rot markiert** und begründet werden.

Abschluss:

Vielen herzlichen Dank für Ihre Beurteilung!

Würden sie für allfällige Rückfragen zur Verfügung stehen? Falls ja, wie lautet Ihre Email-Adresse?

____

Möchten Sie eine Kopie der fertigen Arbeit? Falls ja, wie lautet Ihre Email-Adresse?

____

E Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Hochfelden, 27. September 2013

Roy Weiss