Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)
vorgelegt der
Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich
von

ALESSANDRO CECCONI
VON
VACALLO TI

Begutachtet von
Prof. Dr. Robert Weibel
Prof. Dr.-Ing. Monika Sester
Dr. Dirk Burghardt

ZÜRICH 2003

Die vorliegende Arbeit wurde von der Mathematisch-naturwissenschaftlichen Fakultät der Universität Zürich auf Antrag von Prof. Dr. R. Weibel und Prof. Dr. K. Brassel als Dissertation angenommen.

Summary

This thesis deals with the integration of cartographic generalization and multi-scale databases for enhanced and more flexible web mapping. To generate such web maps the corresponding data set must be available in the requested scale. In most cases the required data do, however, not exist and thus a derivation from existing data sets of a different scale is necessary. This process of adaptation and reduction of the map content to a requested scale is known as *cartographic generalization*. This adaptation allows to maintain the map legible and clear in spite of the smaller map space available and enables to preserve the main characteristics of the original map. Cartographic generalization encompasses several methods and approaches to carry out such an adjustment. However, most of them are mighty complex and thus tend to consume a lot of compute time. Often an interactive intervention is necessary to create a cartographically adequate solution. For these reasons many generalization algorithms cannot be used for real-time web mapping.

The main goal of this work is to develop suitable strategies for Internet cartography by subdividing the whole generalization process into two phases. This allows to keep the overall process more flexible. The first phase takes place only once before an online user requests a web map (off-line phase). The result of this first phase is integrated in a multi-scale database. The second phase is triggered by a user request (i.e. on-demand) to generate in real-time the appropriate map (on-line phase). In the first phase representations are generated at a series of scales (levels of detail, LODs). As they are created in advance (independent of a user request) all possible generalization algorithms and methods can be applied. LODs are incorporated into a multi-scale database (MSDB) which acts as starting point for the second phase. In the second phase an adapted generalization process takes place in real-time for the desired scale. Out of the given representations (LODs) in the MSDB a map is generated. Depending on user requirements the complexity and sophistication of the generalization process can be controlled.

The contribution of this research work for the first phase lies in the design and structuring of the multi-scale database. The focus was on two aspects: an enhancement of the database with additional information (*data enrichment*) and the linking of corresponding objects or groups of objects across the different LODs (*data matching*). By enriching the content of the database with auxiliary information the generalization process of the second phase can be greatly simplified. For the process in the second phase generalization workflows are proposed for several feature classes and two techniques are presented and discussed in detail which allow to generate new maps out of the given LODs in the MSDB: the mesh simplification technique and the morphing transformation technique.

The first one is used for the implementation of the generalization operator 'typification' which transforms an initial set of building objects (or other point symbols) into a new, generalized group. The centers of gravity of the objects define the vertices of the mesh. The original number of vertices is iteratively reduced until the desired number of retained vertices (i.e. buildings) for the requested scale is reached. The elimination process is determined on the basis of the shortest distance between the vertices. The second method generates an intermediate state from two representations given as LODs of the multi-scale database, so-called keyframes, of the same object. This procedure is well known in the area of computer vision, where an intermediate state is computed out of a finite number of steps (frames). This thesis shows how the morphing technique can be used in connection with a multi-scale database in web mapping.

Zusammenfassung

Diese Dissertation befasst sich mit der Integration von mehrfachskalierten Datenbanken in den kartografischen Generalisierungsprozess für eine Verbesserung und Flexibilisierung der Internet-Kartografie. Um eine gewünschte Karte generieren zu können, müssen die entsprechenden Daten im gewünschten Massstab zur Verfügung stehen. Meistens sind diese aber nicht direkt vorhanden, und eine Ableitung aus bestehenden Daten anderer Massstäbe ist nötig. Dieser Prozess der Anpassung des Karteninhaltes an einen bestimmten Massstab, vor allem wenn der Zielmassstab kleiner ist als der Quellmassstab, wird als *kartografische Generalisierung* bezeichnet. Diese Anpassung führt dazu, dass trotz kleinerer Darstellungsfläche die Karte lesbar und verständlich bleibt und die charakteristischen Eigenschaften der Originalinformation adäquat repräsentiert werden. Die kartografische Generalisierung kennt verschiedene Methoden und Ansätze, wie solche Anpassungen durchgeführt werden können. Die meisten davon sind komplex und rechenintensiv und brauchen daher eine grosse Zeitspanne oder unter Umständen einen interaktiven Eingriff zur Generierung einer Lösung. Sie eignen sich daher nicht in jedem Fall für die Internet-Kartografie.

Das Ziel dieser Arbeit ist es, passende Strategien für die Internet-Kartografie zu präsentieren, indem versucht wird, den komplexen Generalisierungsprozess in zwei Phasen zu trennen und damit eine Flexibilisierung des Prozesses zu erreichen. Die erste Phase erfolgt dabei nur einmal und vor der eigentlichen Anfrage von Benutzer/-innen für eine Webkarte und ist dem Aufbau einer mehrfachskalierten Datenbank gewidmet (off-line Phase). Die zweite Phase tritt jeweils ein, wenn auf Benutzerwunsch (on demand) in Echtzeit eine Karte generiert werden soll (online Phase). In der ersten Phase werden Karten für fest definierte Massstäbe (Levels of detail, LOD) abgeleitet. Da diese im Voraus (unabhängig der Anfrage des/r Nutzers/in) generiert werden, können für den Generalisierungsprozess in dieser ersten Phase alle möglichen Methoden und Algorithmen zur Anwendung kommen. All diese Repräsentationen bilden zusammen die mehrfachskalierte Datenbank (Multi-scale database, MSDB), die als Ausgangspunkt für die zweite Phase wirkt. In der zweiten Phase wird entsprechend dem gewünschten Massstab ein adaptierter Generalisierungsprozess in Echtzeit ausgeführt. Ausgehend von den LOD in der Datenbank wird eine Karte mit dem gewünschten Massstab generiert. Je nach Benutzeranforderungen kann die Komplexität beziehungsweise Verfeinerung des Generalisierungsprozesses bestimmt werden.

Der Beitrag dieser Arbeit für die erste Phase liegt im Design und in der Strukturierung der mehrfachskalierten Datenbank, wobei zwei Teilaspekte hervorgehoben werden: eine Erweiterung der Datenbank durch zusätzliche Informationen (*data enrichment*) und die Verknüpfung entsprechender Objekte oder Objektgruppen in den verschiedenen Kartendarstellungen (*data matching*).

Durch die Erweiterung des Datenbankinhalts mit spezifischer, generalisierungsunterstützender Information kann der Generalisierungprozess in der zweiten Phase vereinfacht werden. Für den Prozess in der zweiten Phase werden nach eingehender Diskussion möglicher Generalisierungstechniken zwei Techniken eingehender geschildert, die erlauben, aus den Darstellungen, die in der mehrfachskalierten Datenbank abgelegt sind, neue Karten mit benutzerdefiniertem

Massstab zu generieren: Die Technik der Maschenvereinfachung und die Technik des Morphings.

Die Erstere wird für die Umsetzung des Generalisierungsoperators "Typisierung" eingesetzt, der eine Anpassung der Anzahl Gebäudeobjekte (oder anderer punkthafter Objektklassen) entsprechend dem Zielmassstab ermöglicht. Dabei bilden die Schwerpunkte der Gebäudeobjekte die Maschengitterpunkte. Ausgehend von einer bestimmten Anzahl Originalpunkte, wird durch einen iterativen Prozess deren Anzahl kontinuierlich verkleinert, bis die gewünschte Anzahl entsprechend dem Massstab erreicht wird. Der Parameter, der über die Elimination der Punkte entscheidet, wird durch den kleinsten Abstand zwischen zwei Punkten definiert. Die zweite Methode generiert aus zwei Repräsentationen desselben Objektes, den so genannten "keyframes", eine dazwischenliegende Form. Bekannt ist dieses Verfahren vor allem im Bereich der Computergrafik, wo eine Form durch die Berechnung einer endlichen Anzahl Zwischenschritte ("frames") in eine zweite Form transformiert wird. Diese Arbeit zeigt, wie diese Technik im Zusammenhang mit einer mehrfachskalierten Datenbank für die Kartografie verwendet werden kann und wo dabei Stärken und Schwächen liegen.

Acknowledgments

Prof. Robert Weibel for his encouraging support and serious and open discussions as well the research liberties offered.

Dr. Mathieu Barrault for his everlasting motivation and enthusiasm for cartographic generalization and critical remarks.

Dr. Dirk Burghardt for his ideas and thought-provoking impulses especially regarding the idea of the 'mesh optimization technique' as well as for acting as a reviewer.

Prof. Dr.-Ing. Monika Sester for acting as a reviewer of this dissertation.

My office partner Martin Galanda for all the endless discussions and suggestions, and the cooperation during our time as PhD students.

Alistair Edwardes and Dr. Ross Purves for giving this thesis a face-lift in English.

My parents and brothers for the moral and mental support in good as well as in less good periods of my work.

Friends, colleagues and all others in the Department that expressed academic and friendly interest.

The Federal Office of Topography for kindly providing test data.

This research work is part of the project 'GENDEM: Map Generalization for Thematic and On-Demand Mapping in GIS' supported by the Swiss National Science Foundation under contract 20-59249.99.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The role of cartography for the purpose of visualization

Visualization takes an increasingly important position in the understanding and explanation of phenomena and processes of everyday life. In different research areas and disciplines the importance of visualization can also not be excluded. It helps to make it easier to better understand complex processes and relations. Today one tries to visualize everything and to display it in any graphical form.

Buttenfield and Mackaness (1991) describe the process of visualization as a representation of information synoptically for the purpose of recognizing, communicating and interpreting pattern and structure.

Visualization can be viewed as a logical extension of communication. Maps have been used for a long time for the visualization of spatial data. They particularly help to better understand spatial relationships. The explicit interest in maps as a form of communication has changed the map making or cartography process over the last years (Peterson 1994).

In the last few years the rapid growth of the Internet and distributed systems has allowed easier access of large databases. Especially in connection with the stronger utilization of GIS (in consequence of the use of these databases), the amount of maps created has increased enormously (Longley *et al.* 1999).

Through this enormous demand for graphical representations in different disciplines and fields, efforts have been made to formalize the visualization process and to define rules and principles. Especially in cartography - which is deemed to be the visualization process for spatial data - such rules play a very important role because different types of data and objects must be displayed in different forms and symbols. For all these kinds of possibilities visualization methods and techniques must be found. Bertin (1967) and (1983) defined first principles with his system of **visual variables** for the visualization of qualitative and quantitative features. These principles have later been extended by Dent (1990) whose work was more dedicated to rules of graphic design. Many of these principles are made primarily for paper maps and are a result of empirical tests. For the display of maps on screens, multimedia maps or web maps other and new restrictions and standards must be defined and tested. Mainly in context with GIS, new graphic techniques must be developed especially for the representation of temporal, animated or three-dimensional data (Buttenfield and Mackaness 1991).

From Paper Maps Towards Web Mapping

Maps have been known for a long time only as printed on paper. In the majority of cases these products have been made on paper based on pre-defined specifications depending on thematic content and/or general user requirements. Due to different reasons maps were mainly a static product created once for a special purpose or application. Keates (1996) has defined three categories of functions of maps. First *general maps*, which show selected landscape features. Second *maps with special purpose*, e.g. designed for path finding, and third *maps with a special subject* e.g. hydrography. It is not very simple to make an exact division of these groups as there is no clear border between each of them. Special characteristics of paper maps have been discussed by Goodchild (1999). He described the advantages as **mobile** and in certain aspects **flexible** but not adaptable for the user's purpose.

With the development of better applications and improved hardware for graphical display (better computational performance, higher resolution and more color representation) the demand for digital maps has risen strongly. At the same time new requirements and restrictions for these new kind of displays have to be defined. The limitations of representing maps on screen are given by the nature of computers, such as screen and color resolution or display size (Elzakker van 2001).

Another reason for the strong growth of digital cartography has been the possibility to allow interaction between map and user and give users the facility to create maps temporarily. The option to produce interactive maps instead of static products allows a user to display temporal or dynamic processes or to change the presented information or even the theme without starting from scratch. The user can therefore create their own map according to their requirements. The boom of the Internet in the 1990s and the success of personal computers in many households gave digital maps and mainly web maps a new significance. Peterson (1999) and (2001) noted that the Internet enormously changed the use of maps, how they are made and in particularly their distribution. In contrast to paper maps, digital maps have other advantages and benefits. They can be personalized, updated easily and are interactive but not mobile.

The latest development step in digital cartography – *mobile cartography* – is the combination of the advantages of paper and digital maps. This kind of mapping has received particular importance since mobile information devices (e.g. Personal Digital Assistant) have become available on the market. The presented information is individual for every user (predefined in preferences and user profiles), the content is dynamically generated and thus flexible for any information and location (Reichenbacher 2001). Using this possibility, the user can create his/her own map on-demand, anywhere, anytime.

On-Demand Mapping

At present different mapping applications can be found on the Internet[1] such as route planning services or location finders. Most of them are not flexible as they have been designed for a well-defined purpose with fixed scales and themes. The user has no facility to change the contents, information density or scale according to their individual requirements. In order to avoid these disadvantages new strategies must be found to allow more flexibility for maps produced on user demand.

On-demand mapping is demand-driven, that is, map products are generated according to user needs. Whether the maps are generated on-line or off-line, in real-time or in a matter of days, and whether the resulting products are on screen or paper is a secondary issue. The primary characteristic is that the user specifies when and how the map is produced. Hence, a paper map that is created upon user request and to user specifications (e.g. Ordnance Survey Landplan products (Gower *et al.* 1997) can be called an on-demand map just like a locator

---

[1] www.mapquest.com, www.map24.com, www.expedia.com

map that shows a user's location on the screen of a PDA.

Hence, in contradiction to Jones *et al.* (2000), who equate on-the-fly and on-demand mapping, we suggest that the two terms denote sufficiently different concepts that they should not be confused. The requirements of on-the-fly mapping and on-the-fly generalization, respectively, are more stringent. First of all, as the name suggests, maps have to be generated in real time, making the process time-critical. Secondly, as real-time processes require dynamic graphics media, the resulting products will undoubtedly be screen maps and subject to more stringent technical constraints. Hence, we can define on-the-fly generalization as follows: The real-time creation, upon user request, of a cartographic product appropriate to a scale and purpose.

In principle, there are two fundamental alternatives to implement real-time generalization. The technically simpler, computationally less costly, but also functionally less flexible solution relies on off-line computation (or even manual production) of a multi-scale database containing several levels of detail (LODs). As an alternative to this representation-oriented approach, the process-oriented approach relies on generalization procedures which are carried out in real-time. This approach is technically more demanding, computationally more costly but functionally more flexible, as it is not restricted to given LODs but could generate maps at arbitrary scales. Clearly, there is a trade-off between technological maturity and computational cost on the one hand (which look positive for the representation-oriented approach) and flexibility of product generation and cartographic quality on the other hand (which are the strengths of process-oriented 'true' generalization). Realistically, today, operational applications for real-time mapping utilize multi-scale databases (i.e. LODs). True on-the-fly generalization is still very much a research issue. In the following chapter 2, we will review the requirements and the state of the art for both fundamental approaches, providing also an outlook on currently ongoing research that attempts to combine both multi-scale databases and generalization into a common framework.

The Importance of Cartographic Generalization

Independent from the kind of publication medium – paper or digital – the adaptation of the content of a map is strongly scale dependent. It is evident that a small-scale map contains less detailed information than a large scale map of the same area. Not only the scale but also the theme of the map specifies the density of represented data. The process of reducing the amount of data and adjusting the information to the given scale and theme is called *cartographic generalization* (Müller (1991),Weibel and Dutton (1999)). This procedure abstracts and reduces the information from reality while meeting cartographic specifications, and maintains the significant characteristics of the mapped area. It can be easily seen that this process is very complex and thus time-consuming. Mainly with the spread of new media (such as multimedia, Internet), the speed of the information delivery process has become increasingly important. Today *real-time* generalization is required, since no user expects to wait more then a few seconds for a visualization of a personalized map over the Internet (Feringa 2001). Nevertheless, due to the increased information volume and density that can be delivered by Internet Map Servers today cartographic generalization is becoming more important then ever and must be adapted to the new requirements in the map creation process (Kraak and Brown 2001). Besides this, future map products must still be more service-oriented. One the one hand the user should not be faced too much with defining detailed requests. The system must perceive what type of map the user wants to have by means of pre-set preferences and the request that was generated. On the other hand, the possibility for expert users must be available to define more exact requests (e.g. scale, layout, content, and spatial extends).

1.2 Objectives

As mentioned above, the utilization of on-demand web mapping becomes increasingly important in the domain of cartography. For this kind of application the traditional cartographic approaches for automatic mapping only partly suffice. Thus, new methods and algorithms must be developed to optimize the mapping process owing to these new circumstances. This work focuses on defining strategies and applying non-traditional cartographic generalization operations for on-demand web mapping. The following list gives the general objectives of this thesis:

- Proposing a strategy for improving the generalization process for on-demand web mapping using a multi-scale database;
- Entering alternatives for parts of the generalization process to optimize the mapping process;

The more specific objectives are:

- Defining a data model adapted to the generalization process;
- Development of matching processes to link corresponding objects represented in the different LODs;
- Introduction of techniques for the generalization operators typification and simplification.

The aim of this work is to contribute in this research area and to provide new insights into the interplay of cartographic generalization and multi-scale databases.

1.3 Organization of the Thesis

The research work in front is divided in seven chapters with the following content:

- **Chapter 2** 'State of the Art' of the generalization research. The foci are set on approaches for on-the-fly generalization for on-demand mapping and on the build-up of a multi-scale database. A short summary and assumptions made end the chapter.
- **Chapter 3** Definition of the framework for on-demand mapping. Suggestion of a solution and explanation of the conceptual framework including the different strategies for the generalization process.
- **Chapter 4** Definition of the data model and the structure of the database starting from the given data sets. Besides this added information, known as data enrichment, for speeding up the generalization process will be explained and illustrated with different examples. A second focus is set on the matching process of corresponding objects in different levels of detail.
- **Chapter 5** A description of the generalization process for the selected feature classes *road network*, *building* and *river network* constitute the main part of this chapter. Further, two techniques – mesh simplification and morphing transformation – are presented to replace time-consuming generalizations operators.
- **Chapter 6** This starts with an introduction to the selected platform (Gothic LAMPS2) and the implementation of the multi-scale database in this system. Thereby the matching process for the feature class *building* and the added information to enrich the original database are discussed. The two main points of this chapter are the description of the implementation and evaluation of the proposed techniques presented in chapter 5.

- **Chapter 7** The last chapter contains a discussion of the proposed framework (including the multi-scale database and the generalization process) as well as an outlook for further improvements and possibilities.

Next to these chapters two appendices can be found at the end of the work. Appendix A defines the terminology and describes often used symbols. This work ends with a short curriculum vitae of the author.

Chapter 2

Cartographic Generalization for On-Demand Web Mapping

2.1 Generalization in Digital Cartography

2.1.1 Definition

Topographic maps form a representation of geographical reality. The smaller the map scale[1] The smaller the scale the more the representation is simplified and abstracted. This *process of generalization* concerns itself with the process of abstraction of represented information subject to the change of the scale of a map. The purpose of generalization is to produce a good map, balancing the requirements of accuracy, information content and legibility. It encompasses the modification of the information in such way that it can be represented on a smaller surface, retaining the geometrical and descriptive characteristics. The essence of the original should be maintained at all smaller scales (see Figure 2.1). However, the map scale is not the only factor to determine the generalization process. The purpose of a map is equally if not even more important and influences the selection of the appropriate map scale.

To keep this complex process as general as possible different attempts have been made to achieve a stringent definition. The International Cartographic Association has defined the process of generalization as 'the selection and simplified representation of detail appropriate to scale and/or the purpose of a map' (International Cartographic Association 1973). McMaster and Shea (1992) gives another general definition of generalization: 'Digital generalization can be defined as the process of deriving, from a data source, a symbolically or digitally-encoded cartographic data set through the application of spatial data and attribute transformation'.

2.1.2 Review of Generalization Research

First attempts to automate cartographic generalization tried to find solutions for restricted special problems, in particular for linear (Douglas and Peucker 1973) or point and area features (Töpfer and Pillewizer 1966). On the other hand a number of authors proposed several *conceptual aspects and models* to better understand the elements of the generalization process.

A first conceptual framework by Ratajski (1967) identified two fundamental types of generalization processes: (i) quantitative generalization, which consists of a gradual reduction in map content depending on scale change; and (ii) qualitative generalization, which results

---

[1] The scale of a map is defined as the ratio between an object in reality and its representation on the map. *Small scale* maps are those which depict the mapped region on a small area of map space (e.g., 1:500'000), while *large scale* maps represent the region of interest on a large map space (e.g., 1:10'000).

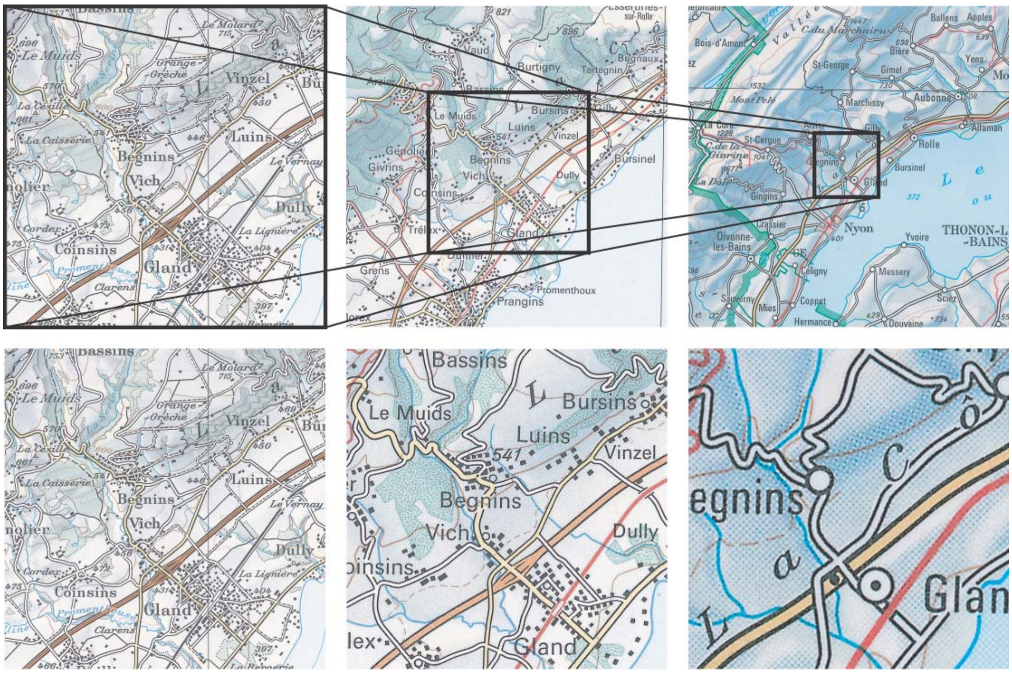


Figure 2.1: Why generalize a map (left: 1:100'000, middle: 1:200'000, right: 1:500'000, not to scale)? Data: © swisstopo (BA034957).

from the transformation of elementary forms of symbolization to more abstract forms. In his framework, Ratajski describes the first generalization steps with the elimination of secondary objects limited by preserving the characteristics of the map. This limit could be overcome only if a certain sum of elements and observations can be replaced by new cartographic methods. Ratajski defines these limits as **generalization points**. The changing capacity of a map can be represented by a triangle, the base of which illustrates the maximum capacity and the top of which illustrates the limits of map capacity. When the map capacity is near the top of the triangle a new cartographic method must be applied in order to initiate a new generalization cycle. Figure 2.2 illustrates the generalization model proposed by Ratajski.

Several other authors proposed models for the digital generalization process (Brassel and Weibel (1988), McMaster and Shea (1988, 1992)). All these models offer a basis for a better understanding of the complex and comprehensive process of generalization.

Brassel and Weibel (1988) developed one of the most detailed conceptual models of map generalization. To be able to simulate the procedure the process of recognizing essential features must be better understood. Understanding generalization means, on the one hand, the extraction of the characteristic structures of the spatial information to identify the process for modifying these structures and on the other hand, to formalize the process of modification as a number of operational steps. The conceptual framework proposed five separate processes of generalization including: (a) structure recognition; (b) process recognition; (c) process modelling; (d) process execution; (e) data display.

An early effort towards automated cartographic generalization was the conceptual generalization model proposed by (McMaster and Shea 1992). According to McMaster and Shea the

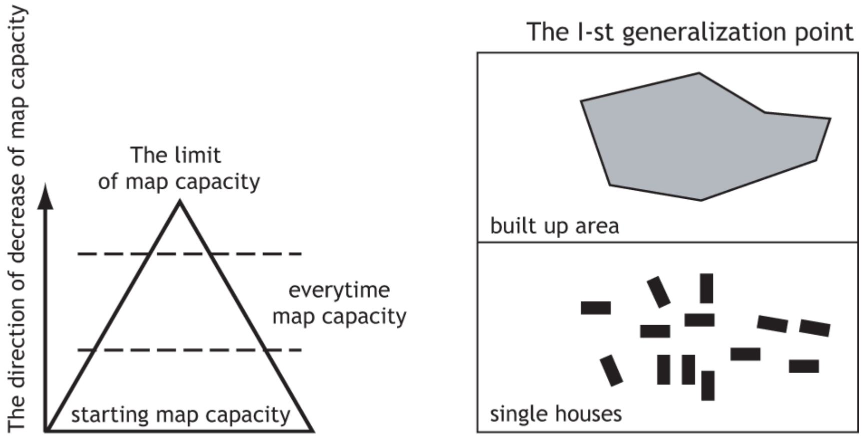


Figure 2.2: The changing capacity of a map may be represented by a triangle (left). On the right hand side, an example of the first generalization point. Near capacity limit, single houses must be replaced by whole settlement (from Ratajski (1967)).

entire generalization process can be divided into three tasks: 1) a consideration of the philosophical objectives of *why* to generalize; 2) a cartometric evaluation of the conditions which indicates *when* to generalize; and 3) the selection of the appropriate spatial and attribute transformations which provide the techniques on *how* to generalize.

Generalization according to Müller (1991) is the application of a transformation to spatial data and not only motivated by a reduction of scale representation initiated by four main requirements: (a) economic requirements; (b) data robustness requirements; (c) multipurpose requirements; (d) display and communication requirements.

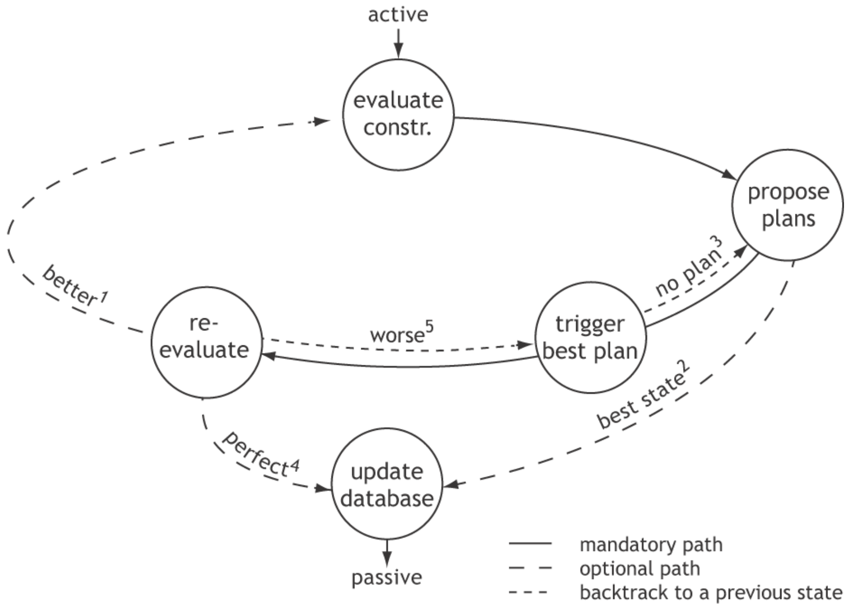
Ruas and Plazanet (1996) proposed a framework controlled by a set of constraints. The dynamic generalization model is based on constraint violations and on the local qualification of a set of objects, represented by means of an object situation. A *situation* is described by 1) the geographical objects involved, 2) their relationships, and 3) the constraint violations (see also 2.1.5).

The object of the AGENT project[2] was to model the holistic nature of generalization, by means of multi-agent technology. Instead of using a centralized plan for the generalization process, it uses local, regional and global constraints, which are stored in agents. These agents are cartographic objects (micro-agents), group of objects (meso-agents) or feature classes (macro-agents) which are able to communicate and affect other agents. The aim of each of these agents is to improve the situation with respect to the attached constraints. Each constraint provides a 'goal value' which each agent attempts to reach by generalizing itself in concert with other agents. The state of an agent is characterized by all of its constraints integrating their values and importance to establish its own 'happiness'. When unhappy, the agent proposes a generalization plan to remedy a constraint violation. In order to improve the search for the best sequence, the agents sort the available plans with respect to ranking and constraints' priority and severity. The agent then triggers the first plan and re-evaluates its own happiness. Depending on the success of the plan (see Figure 2.3) it may backtrack

---

[2] AGENT (Automated generalisation new technology) was a research and development project funded by the ESPRIT 4 programme (EU information technologies programme) between December 1997 and December 2000 (http://agent.ign.fr).

and trigger the next plan, stopping if a perfect state is reached or starting a new cycle if the happiness has improved until no further plans are available.



[1]better	current state improves the agent's happiness; new list of plans to enhance further the current state;
[2]best state	the best compromise (state) of constraint satisfaction is passed to [update database];
[3]no plan	no alternative plan is left to improve the current state; return to a previous state (list of plans);
[4]perfect	all constraints are completely satisfied; current state is passed to [update database];
[5]worse	current state degrades agent's happiness; another plan of the current list of plans is tried out;

Figure 2.3: The AGENT cycle proposed by Galanda and Weibel (Galanda and Weibel 2002).

A more detailed description of the AGENT project can be found in Lamy *et al.* (1999), Ruas (2000) and Barrault *et al.* (2001).

2.1.3 Models of Generalization

In cartography and GIS, the generalization process has attained a broader importance. It can be understood as a process of representation of the real world by different models. Generalization takes influence through the building of a first model of the landscape as a DLM (digital landscape model), in which real objects are represented abstractly, a process which is also known as *object generalization*. Moreover, it then takes part in the derivations of secondary models, involving a reduction of the first model depending upon a map purpose (*model generalization*). Finally, it is responsible for adequate cartographic visualization of the primary or secondary models depending on scale and map purpose (*cartographic generalization*) creating a digital cartographic model (DCM). The terminology of these three models has been developed by Grünreich (1985) and adopted by other authors (Weibel and Dutton 1999).

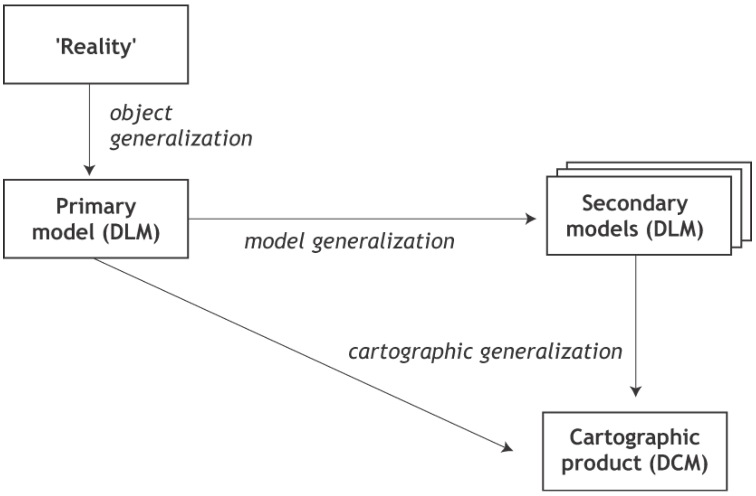


Figure 2.4: Generalization as a sequence of modelling operations (after Grünreich (1985)).

Object Generalization

First the definition and the structuring of the original database takes place. This attempts to achieve a complete data structure of real world objects as well as possible (see Figure 2.4). Generalization occurs here in the sense of abstraction, selection and reduction of given circumstances. These steps are needed, because only determined information and object classes, which are relevant for the intended purpose, are integrated in the database (primary model).

Model Generalization

The process of object generalization is used as a preparatory step for data for traditional as well as digital maps, while *model-oriented generalization* is concerned solely with digital map creation. According to Brassel and Weibel (1988) it is never used for display, but exclusively for data reduction. In digital systems memory and storage can be saved by this generalization process. The main objective here are the application of manageable simplification algorithms[3], for different uses, to minimize memory requirements and increase computer efficiency in analytical procedures. Another objective is the reduction of the resolution for analytical purposes in GIS. In addition data transfer in networks, like the Internet, can be accelerated. The output data from model generalization are not simple map graphics but have to be fit for usage in GIS. Model oriented generalization can be regarded as a preprocessing step for cartographic generalization, but contains only completely formalizable processes with no abilities for subjective graphic decisions.

---

[3] The term *algorithm* is used to describe problem-solving methods suitable for implementation as computer programs (Sedgewick 1992). A generalization algorithm is a mathematical construct that solves a generalization problem.

Cartographic Generalization

The *cartographic generalization process* is connected most frequently with the issue of the generalization of spatial data for representation in maps (Weibel and Dutton 1999). The differences to model generalization lie particularly in the representation of objects through graphical symbolization. The generalization is based on the semantical evaluation and on conditional graphical restrictions. Of the three generalization processes the cartographic one can be considered as the most investigated to date.

2.1.4 Generalization Strategies

In contrast to the previous section, which focused on understanding the generalization process as transition between different models, this section deals with the definition of strategies. These strategies try to comprehend the whole generalization process and start with different conditions. Based on Weibel (1997a) who presents two different views of strategies, a third one is proposed:

- **Process-oriented view**: Deriving generalization from a detailed database. This view understands the generalization process as a derivation of data departing from a single detailed database to an arbitrary map with respect to scale and theme and depending on the requested purpose. Due to the complexity of the generalization process, complete solutions for the automated process are still not available yet (but partial solutions do exist).

- **Representation-oriented view**: An alternative approach is the development of a multi-scale or multi-representation database[4] including maps with different fixed defined scales. This approach is called a 'representation-oriented view' because different *representations* of different scales are stored in one database. While this strategy overcomes the problems of generalization methods, it poses maintenance problems, especially in updating the different scale levels (inserting, deleting or moving objects) without introducing inconsistencies between the levels.

- **Derivation-oriented view**: This strategy merges the two strategies above. Similar to the representation-oriented view, this strategy is composed of different levels of detail, which are, however, derived from one base data set by applying a generalization process. Thus, the data set is consistent and the corresponding objects at the different levels are connected, building a hierarchical structure.

All three views possess advantages and also disadvantages (see section 2.3) which may make the mapping process inflexible concerning the requested map scale and theme. As neither one of the views entirely fulfills all the requirements for use in web mapping, a new strategy for creating maps must be developed including the benefits of all presented views (see chapter 3).

2.1.5 Generalization Constraints

The automation of the generalization process requires an understanding of the relationship between the different map objects and their informative value. Thus, as in every automation of a process, formulations and definitions of conditions are needed to formalize the process. This procedure, formalized as a *set of rules*, can be used either to generate knowledge or to manage some procedural choices. For map design, simple rules may not accommodate the flexibility required. Beard (1991) introduced the notion of **constraints** which she claims to correspond

---

[4] In other research works the term *multi-scale* is sometimes replaced by the synonymous *multi-resolution* (Kilpeläinen and Sarjakoski 1995).

to the rule predicate. According to her the distinction between rules and constraints is that "a constraint is not bound to a particular action. The overall rule is that all constraints must be satisfied or resolved, but any number of actions can be applied to resolve them" (Beard 1991, p.122).

These constraints are translations of the required conditions that should take account of not only the objects but also the purpose and the final state of a map. Different authors have made classifications of types of constraints. Beard has organized the constraints in four types: *graphic*, *structural*, *application*, and *procedural* constraints. Ruas and Plazanet (1996) concentrated only on constraints related to objects and have tried to understand the ontology of these constraints in combination with the generalization process. They differentiate four types: *legibility* (define the perceptibility threshold of objects), *shape* (define important shape information to be preserved), *spatial* (define the position of an object or the position of objects in relation to one another), and *semantic* (used to represent indicators). Weibel and Dutton understood a constraint as a design specification to which a solution should adhere and defined it as "a limitation that reduces the number of acceptable solutions to a problem" (Weibel and Dutton 1998, p.215). They classified constraints into five categories: *graphical* (specify the minimum size and width and are directed by graphical limits), *topological* (ensure that the existing relationship between features are maintained), *structural* (define criteria describing spatial and semantic structure), *Gestalt* (relate to aesthetics and visual balance), and *process* (mainly reflect how operators are selected and sequenced).

Other authors, such as Mackaness (1995) and Berg de *et al.* (1995), have dealt with constraints and defined other categories and types. Feature class specific work on constraints has been made for building generalization (Regnauld 1998) and for line simplification (Weibel 1997b).

2.1.6 Generalization Operators and Algorithms

Generalization Operators

The generalization process involves a great deal of human interpretation of the geographical data and decisions about what and how to generalize and how to solve potential generalization conflicts. The manual process is thus of an intrinsically holistic nature. For automated generalization it is almost impossible to develop such a holistic solutions due to its the subjective nature and the lack of well-defined rules to guide decision making. The alternative is to decompose the overall generalization process and automate it as much as possible as a computational process. In order to define generalization constraints, it is necessary to understand exactly what happens when a cartographer generalizes a map, and to make the operations explicitly defined for digital implementation. Hence, the generalization process must be subdivided into a set of **generalization operators** which are able to solve very specific problems of the generalization process.

Significant differences can be found in the literature in both the number of generalization operators and in the terminology used to describe them (Rieger and Coulson 1993). For example, Robinson *et al.* (1995) lists four procedures while McMaster and Shea (1992) listed 12 different terms. McMaster and Shea's typology was the first detailed one which more or less achieved the requirements of digital cartography. Their classification is divided in two types of transformation, ten operators with a spatial character and two with an attribute character (see Table 2.1). Interestingly the often used operators *selection/elimination* are missing from their set.

Another set of generalization operators were presented by Beard (1987) for application to categorial data coverages. It should be noted that different definitions for the same term may be used (see Weibel and Dutton (1999) for a review). This led other authors to refine definitions or add new operators (Ruas and Lagrange 1995). Bader *et al.* (1999) shows the

Spatial transformation	simplification	smoothing	aggregation
	amalgamation	merging	collapse
	refinement	exaggeration	enhancement
	displacement		
Attribute transformation	classification	symbolization	

Table 2.1: Types of the generalization operators proposed by McMaster and Shea (1992).

operator classification proposed for the AGENT project (see Figure 2.5), which is based on a multi-agent system (Lamy *et al.* 1999). This classification builds the base of our work concerning the terminology of generalization operators.

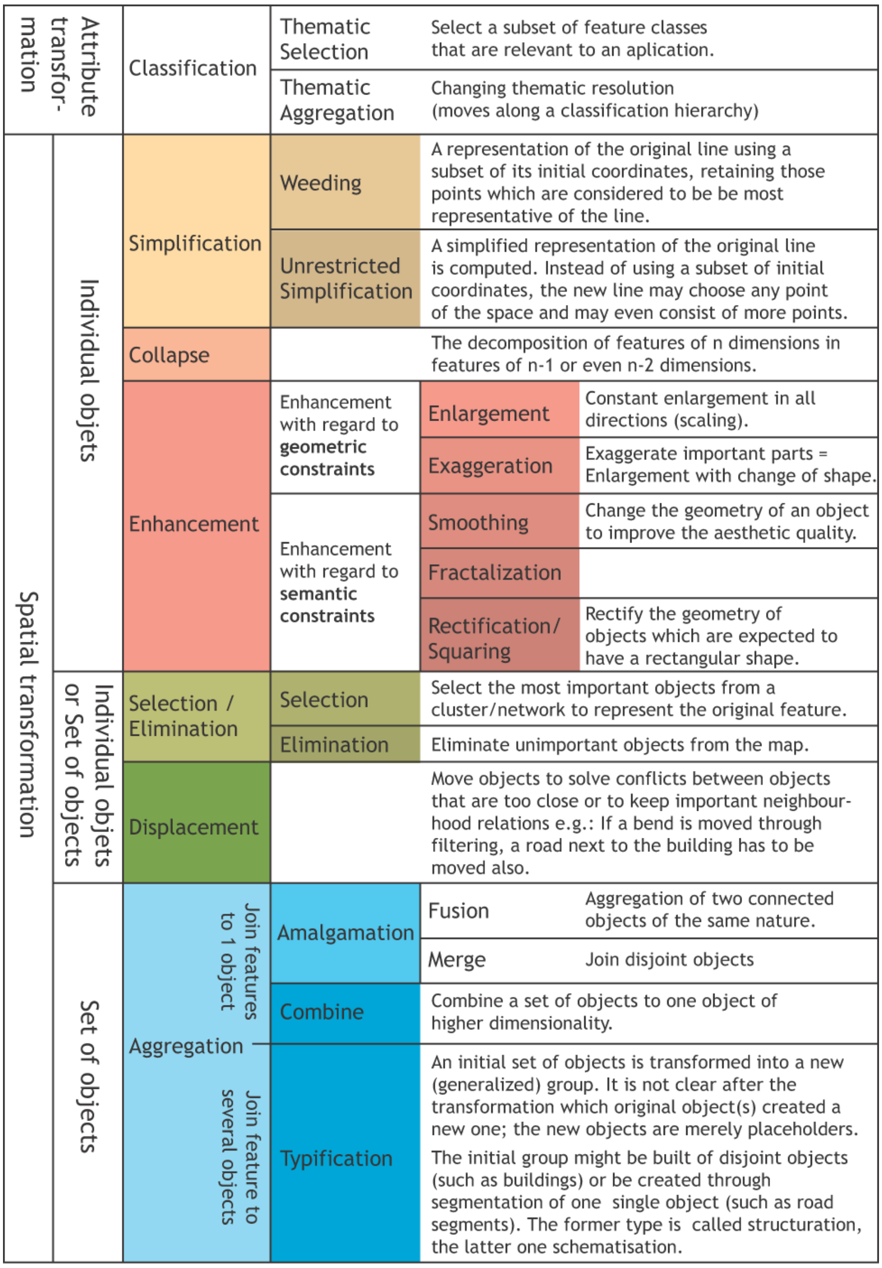
The main task of a generalization operator is to solve a specific generalization problem. Using such operators it is possible to break down the overall generalization process into smaller sub-processes which are much less complex to solve than the whole process. The combination of these generalization operators can then be used to build an entire generalization process or workflow. Different factors determine which operator must be applied for a given situation. Three main elements influence the use: the *preceding situation analysis*, *the feature class* (e.g. road, river) and the *map scale*. Additionally, it must be noted that generalization operators do not all behave equally. Especially for the utilization in on-demand web mapping, it would be helpful and important to distinguish between two types of operators acting differently (Ruas 1999):

- **Independent** (also termed context-independent by Weibel (1997a)) This kind of operator is applied to individual objects or groups of objects independent of their spatial context. No spatial analysis must be performed to determine, for instance, the nearest objects or what effects they have after a transformation on other objects. Examples for this type are *simplification* and *smoothing*.
- **Contextual** On the other hand context-dependent operators, like *selection*, *aggregation*, *displacement* or *typification* can only be triggered and controlled following spatial analysis of the context. Especially for the operator displacement, which is used to move objects to solve conflicts between objects that are too close or to keep important neighborhood relations, an inspection of the surrounding objects is needed.

While generalization operators designate abstract transformations in the generalization process, generalization *algorithms* put the geometric and semantic transformation of the data set into execution. This ensures that, in contrast to generalization algorithms, operators are independent of a particular data model (e.g. vector or raster). The development of adequate generalization algorithms and the integration of them into a comprehensive workflow is thus more important.

Generalization Algorithms

Independent of how the composition and the order of the operators are defined, the relationship between *generalization operators* and *generalization algorithms* is hierarchical. An operator defines the transformation that is to be achieved; a generalization algorithm is used to implement the particular transformation. This fact implies that operators are, in contrast to algorithms, independent of the data model (e.g. vector or raster data). Two different types of algorithm can be distinguished in terms of how they are to be used. We can distinguish algorithms that have been developed to be used for specific feature classes (e.g. buildings, roads)



Attribute transfor-mation		Classification	Thematic Selection		Select a subset of feature classes that are relevant to an aplication.
			Thematic Aggregation		Changing thematic resolution (moves along a classification hierarchy)
Spatial transformation	Individual objects	Simplification	Weeding		A representation of the original line using a subset of its initial coordinates, retaining those points which are considered to be be most representative of the line.
			Unrestricted Simplification		A simplified representation of the original line is computed. Instead of using a subset of initial coordinates, the new line may choose any point of the space and may even consist of more points.
		Collapse			The decomposition of features of n dimensions in features of n-1 or even n-2 dimensions.
		Enhancement	Enhancement with regard to **geometric constraints**	Enlargement	Constant enlargement in all directions (scaling).
				Exaggeration	Exaggerate important parts = Enlargement with change of shape.
			Enhancement with regard to **semantic constraints**	Smoothing	Change the geometry of an object to improve the aesthetic quality.
				Fractalization	
				Rectification/ Squaring	Rectify the geometry of objects which are expected to have a rectangular shape.
	Individual objets or Set of objects	Selection / Elimination	Selection		Select the most important objects from a cluster/network to represent the original feature.
			Elimination		Eliminate unimportant objects from the map.
		Displacement			Move objects to solve conflicts between objects that are too close or to keep important neighbour-hood relations e.g.: If a bend is moved through filtering, a road next to the building has to be moved also.
	Set of objects	Aggregation Join features to 1 object	Amalgamation	Fusion	Aggregation of two connected objects of the same nature.
				Merge	Join disjoint objects
			Combine		Combine a set of objects to one object of higher dimensionality.
		Join feature to several objects	Typification		An initial set of objects is transformed into a new (generalized) group. It is not clear after the transformation which original object(s) created a new one; the new objects are merely placeholders. The initial group might be built of disjoint objects (such as buildings) or be created through segmentation of one single object (such as road segments). The former type is called structuration, the latter one schematisation.

Figure 2.5: Generalization operators defined for the AGENT project (simplified after Bader *et al.* (1999)).

from generic kept algorithms (e.g., generic line simplification) that can be used by modifying user-definable parameters. Most operators can be implemented by several algorithms with special characteristics. An overview is given by Figure 2.5 (Bader *et al.* 1999). McMaster and Shea (1992) determined two factors important in making the appropriate algorithm selection:

process efficiency and **accuracy**. It must be noted that for our work only algorithms for vector data are considered. A review of raster-based algorithms can be found in Schylberg (1993) and Jaakkola (1997).

For a long period research focused primarily on the development and comparison of algorithms for the *simplification operator* rather than on other aspects of the generalization process. For line simplification different authors proposed local (Reumann and Witkam (1974), Lang (1969)) or global (Douglas and Peucker (1973), Cromley and Campbell (1992)) solutions. Li and Openshaw (1992) describes an algorithm that allows the geometry of an object to be changed which they claim to preserves the original form (termed unrestricted simplification). Most of these proposed solutions do not analyze and consider the characteristics and the generating process of a line but treat the object as a simple geometric primitive. Especially in cartographic generalization, however, the meaning of a line plays a decisive role for applying an algorithm. Depending on the feature class (e.g. road, coastline, river, administrative boundary) the corresponding algorithm or parametrization of an algorithm must be applied. Algorithms which take the semantics into account have been presented for coastlines (Wang and Müller 1993), buildings (Lichtner 1979) and roads (Mustière 1998).

The *smoothing operator* is often understood as a cosmetic operation, but it can also be used to remove small crenulations. One of the most well-known methods are the Bezier curves (Bézier 1968). These routines try to approximate a sequence of points with a mathematical function. Their advantage is that the shape can easily be deformed using few vertices and control points. Therefore they are particularly useful for smoothing of road objects (Plazanet *et al.* 1995). Li and Openshaw (1992) proposed a 'natural principle' of line generalization and presented three different algorithms: vector-mode algorithm, raster-mode algorithm and a raster-vector algorithm.

Because of their *context independence* and thus *low computational costs*, these generalization operators will represent the ones to be included in an on-demand web mapping environment.

For the *selection operator*, as a process for reducing the number of objects, several solutions have been presented. Töpfer's radical law (Töpfer and Pillewizer 1966) was one of the first attempts to conceptualize different rules in a mathematical form. The formula can be applied mainly to the generalization process of topographic maps. For the selection of river networks Rusak Mazur and Castner (1990) proposed using the Horton order (Horton 1945) which gives a relatively simple and effective ordering of river segments. Thompson and Richardson (1995) presented a prototype for a network analysis that allows an efficient extraction and handling of the road network and thus support in finding the essential or representative characteristics. Concerning the feature class *building*, Ruas (1999) proposed a method which progressively removes the 'worst building' by means of a cost function based on size, density and directional proximity.

The operator *typification* reduces a set of objects to a new group which has to show similar characteristics concerning density and orientation. Sester and Brenner (2000) proposed an approach for typification of 2D-structures of similar type and size (e.g. buildings). The method is based on Kohonen Feature Maps, a neural network learning technique. The prominent property of this unsupervised learning method is the fact that the neurons are adapted to a new situation, while keeping their spatial ordering.

More algorithms for contextual generalization, such as displacement, aggregation and typification are described in Bader *et al.* (1999).

It is not simple or even possible to define a fixed sequence or order for the application of generalization operators which is universally valid for every map scale and purpose. For each situation or problem the operators and the algorithms used must be re-arranged. Another fact is also that two algorithms used in reverse order do not lead to the same result (Monmonier and

McMaster 1991). Thus, it is very important to fine-tune the interaction between algorithms by changing and adjusting their parameters for specific generalization problems. Some pragmatic guidelines for operator sequences can be driven from cartographic practice (Weibel 1997a):

1. **Selection** and **elimination**: Increases available space by removing non relevant information.
2. **Aggregation**, **amalgamation**, and **merging**: Combine selected features and possibly induce topological changes (e.g. river: polygon to polyline).
3. **Simplification**: Reduces detail and allows a more efficient calculation.
4. **Smoothing**: Follows simplification and allows a aesthetical refinement.
5. **Displacement**: Employed to resolve topological conflicts caused by previous operators.

This solution must allow the possibility to make additional modifications at run time. It is important to coordinate and prioritize which operators should be used for the requested type of map (e.g. detailed paper map, or on-demand map). An example of an application which allows a cycle in a process chain is the AGENT project described above (see therefor section 2.1.2).

2.2 On-Demand Web Mapping

In parallel to the rapid growth of the Internet, web mapping also experienced a swift evolution, given the high reliance of Internet applications on graphics. According to van Elzakker (2001) web mapping offers two main benefits: **accessibility** and **currency**. Accessibility means that through the use of web mapping, the user gains access to an enormous body of information and maps. Currency refers to the fact that updates in a database can be immediately reflected in web maps, that is, no delay takes place between database updates and map updates. The lack of currency was one of the main weaknesses of conventional mapping; due to the length of the conventional production process, maps were often out of date by the time of publication.

2.2.1 Definition of On-Demand Web Mapping

As introduced in section 1.1, on-demand web mapping is a concept resulting from the increase of power and speed of global data networking and transfer capabilities as well as increased awareness of data users, who want to produce their own maps according to their own specifications. *On-demand* or *on-the-fly* web mapping describes the process of mapping done by the user at the time of interest for a specific map. Certain authors do not distinguish between these two terms (e.g., Jones *et al.* (2000)). But since they rely on different concepts, it will be good to differentiate between them to prevent confusion. The term **on-the-fly** emphasizes the *immediacy* and *user interaction* with data and processing methods, whereas on-demand mapping accentuates more on user selection of *mapping techniques* and source data (NOAA National Geophysical Data Center 1997). **On-demand web mapping** can thus be defined as: the creation of a cartographic product upon a user request appropriate to its scale and purpose. The main characteristics of on-demand web mapping are:

- A temporary, reduced scale data set is generated for visualization purposes from the database (Oosterom van 1995) in order not to store data redundantly and thus to use memory efficiently;
- The visualization data set has to meet the user's preferences (which can be stored in the predefined user profile or selected by the user) and the technical display specifications;

- The scale and the theme are arbitrary and not predefined to ensure the optimal flexibility;
- The visualization is accomplished automatically with no possibility of checking the result before publishing it on the display;
- The resulting map must appear on the display within a certain time slot depending on the user preferences (Feringa 2001);
- On web and mobile devices, there is an additional problem of network bandwidth.

2.2.2 Requirements and Constraints

At the beginning of its evolution, web mapping was restricted to the delivery of static and simple maps in raster formats (GIF, JPEG or PNG)[5]. Since those early days, the technical capabilities (graphics capabilities, network bandwidth) on the one hand and the user requirements on the other hand have increased continuously, giving rise to a continuous evolution in the direction of true on-demand web mapping. The current state of the art of commercial web sites providing web mapping functionality is perhaps best illustrated by sites such as MapQuest[6] and MS Expedia Maps[7]. For more specialized map purposes, the user has the option to choose the window that is displayed, the scale, and possibly also the data layers that are displayed. Maps are then 'cut out' from the appropriate scale level from a multi-scale database that consists of several, fixed levels of scale (or levels of detail, LODs) that are not connected to each other. The pre-defined map is then sent as a raster image to the web client. So, the flexibility of 'on-demand' mapping is relatively low in terms of the choice of map theme and scale. However, if the user stays within the pre-defined purpose of the map server (e.g. route mapping) the results are acceptable, witness the huge number of hits that sites of the above kind receive every day.

Independent of the approach that is used for implementation, several technical constraints influence the presentation of maps on the web. The most important ones are (for more details and constraints, see Arleth (1999)):

- **Transfer rate**: web mapping has to deal with limited transfer rates. Presently, 52-128 kbps can be expected using modems (in industrialized countries);
- **Color depth and quality**: 16 to 24 bits color depth are common used today and sufficient for most purposes. However, depending on the screen, colors will be rendered quite differently (i.e. the same color will look different on different screens);
- **Screen resolution**: although screen resolution has been continuously increasing and is now commonly at 1024x768 pixels, we will have to live with resolutions that are dramatically below that of paper-based products in the years to come.

A further difficulty of web mapping is that the above technical constraints are largely beyond the control of the map maker, since the precise characteristics of the display and data transfer may vary significantly between different clients. Also, as the web is literally global and can be accessed by anyone (unless restrictions are deliberately imposed), it is difficult to

---

[5] The Graphics Interchange Format (GIF98a) defines a format intended for the on-line transmission and interchange of raster graphic data in a way that is independent of the hardware used in their creation or display. JPEG stands for Joint Photographic Experts Group and is designed for compressing either full-color or gray-scale images of natural, real-world scenes. It works well on photographs, naturalistic artworks, and similar material. The Portable Network Graphics (PNG) format was designed to replace the older and simpler GIF format and has three main advantages: alpha channels (variable transparency), gamma correction (cross-platform control of image brightness), and two-dimensional interlacing (a method of progressive display).

[6] www.mapquest.com (accessed: February 2003).

[7] www.expedia.com (accessed: February 2003).

make any assumptions about the prospective readership of web maps, and their map reading abilities, cultural background, etc.

In terms of workload distribution between server and client, two main approaches exist today. *Thin clients* permit running software on a remote server rather than running it locally on the desktop. This saves overhead and expense because the desktop machine that accesses the server doesn't have to be as powerful as a machine capable of running the entire software locally. This allows control of the characteristics and security of displaying. The result will be sent to the user as a raster graphic referenced in HTML-code. The alternative approach, moves more functionality to the client which now becomes a *thick client* (typically a Java applet or specialized plug-in to a web browser) that can handle some of the requests locally (e.g. zoom, pan, switching on/off data layers, modification of symbology). Requests to the server are reduced to a minimum and the flexibility of local display generation is increased, but this strategy also implies that not only the requested map but also further information (e.g. additional levels of detail for zooming) must be downloaded to the client. Client-side solutions typically only make sense when vector formats for web graphics such as SVG[8] (Cecconi and Galanda 2002) are used.

2.3 Approaches to On-Demand Web Mapping

For the flexible creation of maps on-demand at arbitrary scales and for different themes, **cartographic generalization** is a necessity. However, cartographic generalization is also known to be a time-consuming process, which runs counter to user expectations for Internet services, where the time factor is critical. The above point shows why for web mapping in particular, real-time generalization would be such an important utility and has become one of the essential factors.

In principle, there are three fundamental alternatives to implement on-demand web mapping (cf. section 2.1.4 Generalization Strategies):

- **Representation-oriented approach**: The technically simpler, computationally less costly, but also functionally less flexible solution relies on off-line computation (or even manual production) of a multi-scale database containing several *independent* levels of detail.

- **Process-oriented approach**: As an alternative to the above strategies, this approach relies on generalization procedures which are carried out on-the-fly.

- **Derivation-oriented approach**: Similar to the representation-oriented approach this strategy relies on computing LODs of a MSDB by off-line generalization from a detailed data set and thus maintains consistency between LODs.

Clearly, there is a trade-off between technological maturity and computational cost on the one hand (which look positive for the representation- and derivation-oriented approach), and flexibility of product generation and cartographic quality on the other (which are the strengths of process-oriented 'true' generalization). Realistically, today, operational applications for real-time[9] mapping utilize multi-scale databases. True on-the-fly or real-time generalization is still very much a research issue (Lehto and Kilpeläinen (2001), Oosterom van (1995), Harrie *et al.* (2002)).

---

[8] SVG (Scalable Vector Graphics) is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text.

[9] The term *real-time* can be used as a synonym for *on-the-fly*. In other disciplines 'real-time' is used for the transmission of information within milliseconds. In this context it will include also *near* real-time processing.

Representation-Oriented Approach

The representation-oriented approach is the most widely used approach in association with on-demand mapping applications in the Internet environment. A variety of advantages make it possible to deal with easily for every kind of user. Mapping applications relying on this approach are often developed for a particular purpose or requirement like location requests, route guidance and navigation, nearest point search, road trip planner, or search of points of interests. The setup of such an application can be very easy, if the precalculated LODs are available. Upon every user request, the appropriate window of a predefined LOD[10] is sent back to the client allowing change between the fixed scales. Table 2.2 compares the advantages and disadvantages of these types of applications.

Advantages	**Disadvantages**
Since in most cases the different LODs are available (e.g. from national mapping agencies) the setup of such an application is simple.	Updating the LODs of a multi-scale database is not easy, because every level must be updated separately, and updates propagated across scales.
Everything is pre-calculated and must not be done at run-time. This allows real-time mapping.	Corresponding objects in the different LODs are not hierarchically connected.
Applications are for a particular purpose and work very well for this one (but not for others).	As everything is stored at fixed levels and contents in the database, applications are not flexible, according scale and theme.

Table 2.2: Comparison of properties of the representation-oriented approach.

Process-Oriented Approach

In contrast, process-oriented approaches are sparcely widespread in the context of web mapping. The main reason for this situation is the long response time of a user request. It is not that the systems are poorly equipped, but the calculation time for a result still needs too much time to achieve real-time applications. Since an on-the-fly generalization process must be carried out, this approach is technically more demanding. The complex functionality of the procedure increases the computational cost immensely but allows wide flexibility according to the map scale and theme. As it is not restricted to a given LOD but can generate arbitrary scales, it is adaptable to every user request. Advantages and disadvantages are summarized in Table 2.3.

Derivation-Oriented Approach

The derivation-oriented approach is a further development of the idea of the representation-oriented approach, where a MSDB with several LODs builds the core of the system. In this case, however, the scale levels are not independently generated but derived from one detailed base data set applying a semi-automatic generalization process[11] to get the new scale levels. The idea of building a MSDB with derived and consistent data levels arises from different National Mapping Agencies (NMA) that want to maintain only a single detailed base data

---

[10] A good example on the web is the application of MapQuest (www.mapquest.com). MapQuest offers *ten* different predefined zoom levels from continent level to street level.

[11] A semi-automatic generalization process allows some steps to be accomplished automatically, but not the whole process. The remainder must be done by hand by a cartographer.

Advantages	**Disadvantages**
The application is extremely flexible and allows the production of scale or theme dependent outputs.	Computational costs are very high, limiting the use in real-time contexts.
The database consists only of one detailed data set at one scale (e.g. 1:1'000), facilitating the updating process enormously.	It is strongly time-sensitive depending on the involved number of objects.
Duplication of the stored data can be avoided.	A fully automated generalization process is needed but not existing yet.
The structure ensures consistency and integrity between the various scale outputs.	

Table 2.3: Comparison of properties of the process-oriented approach.

set and not have to update or modify all existing LODs each time a change is introduced. This strategy is scarcely used until now because the creation of the different levels needs a lot of time, human resources and thus production costs. Advantages and disadvantages of this approach are summarized in table 2.4

Advantages	**Disadvantages**
The LODs can be pre-computed by applying a generalization process and are thus not time-sensitive.	Production costs regarding time and human resources for the creation of a data set are very high.
The hierarchical structure between the LODs simplifies the generalization process.	The approach is not flexible enough due to the LODs defined at fixed levels.
Integrity and consistency are ensured by the hierarchical structure.	Since several levels are stored the data set contains redundant information.

Table 2.4: Comparison of properties of the derivation-oriented approach.

It can be seen that for the development of a strategy for on-demand mapping none of these approaches achieves the requirements and constraints defined in section 2.2.2. Thus a new procedure must be found which combines the advantages of the methods fulfilling all tasks of on-demand web mapping. This new strategy based on a fusion of MSDB and generalization, will be presented in detail in chapter 3.

2.3.1 Generalization Algorithms for On-Demand Web Mapping

Until today, this part of the research in cartographic generalization has not been attended as well as other topics (e.g. line simplification). Only a few studies of real-time generalization have been published so far.

One example of such fast algorithms are simple *selection algorithms* that merely rely on pre-computed attributes, such as the Horton stream ordering scheme used for river network selection (Rusak Mazur and Castner 1990); another example of attribute-based selection is given in Lehto and Kilpeläinen (2001). Another category of fast algorithms are the many line simplification algorithms available with a simple computational logic, such as the Douglas-

Peucker algorithm (Douglas and Peucker 1973). Glover and Mackaness (1999) provide an example where several simple algorithms and a database with pre-computed scale-related attributes are combined into a solution for on-the-fly generalization. Algorithms that exploit hierarchies are also well suited for time-critical applications. Hence, the algorithm by Douglas and Peucker (1973) can be pre-computed and a tree data structure of the resulting simplification tolerances can be built, as was done in the 'binary line generalization' (BLG) tree described by Oosterom van (1995). Van Oosterom also describes a tree that allows a more complex generalization operation, called the 'generalized area partitioning' (GAP) tree. This tree defines successive hierarchies of aggregations of adjacent polygons in a polygon mosaic.

As was noted in section 2.3, on-the-fly generalization is a time-critical task. Besides this more generic fact, generalization algorithms that are suited for this purpose must achieve at least one of these two factors:

- **Fast and effective** In principle, all known generalization algorithms that run in linear or logarithmic time can be used for on-the-fly generalization.
- **Must be supported by pre-computed data structures or attributes** Using precalculated information can help to run an algorithm faster or to leave out some calculation parts needed for the process.

In general it can be ascertained that some generalization operators are better qualified for on-demand mapping than others, with respect to the definition in section 2.2.1 (Lehto and Kilpeläinen 2001). Table 2.5 gives an overview of a possible classification of some important operators.

Suitable	**Less suitable**
Selection/Elimination	Displacement
Aggregation	Typification
Simplification	

Table 2.5: Classification of well and less suited generalization operators for on-demand web mapping.

Since *true* on-the-fly processing is not possible yet, alternative approaches relying on pre-computation and database representation have been explored as well and will be presented in the next sections.

2.3.2 Other Approaches for On-Demand Web Mapping

Another term associated with real-time generalization is *progressive vector data transmission*. The first work on progressive transmission of vector data on the Internet has been by Buttenfield (1999) and (2002). She argues that progressive data transmission is non-existent for vector data, while it is a well known and successfully applied principle for raster-based images (e.g. in the GIF and PNG format), where coarser versions of the data are displayed before the complete image is downloaded. This technique is mainly of importance for large data sets that must be transmitted across slow communication links, and could be helpful for vector data too. The research by Buttenfield focused on the hierarchical subdivision of vector data by using tree structures based on the Douglas-Peucker algorithm (Douglas and Peucker 1973), similarly to the BLG tree by Oosterom van (1995).

More work in this area was done by Bertolotto and Egenhofer (2001). The focus of their research was mainly on progressive transmission of vector data in conjunction with on-the-fly mapping over the web. They propose the creation of multiple representations of the data set corresponding to different levels of detail. These LODs are then sequentially transmitted to the client in the order of increasing detail. In fact, only the increments are transmitted and added to the currently displayed representation. While the proposal is conceptually interesting, the main problem that remains to be solved in practical terms is the integration of the transmitted data on the client side, not only with respect to graphics but also in terms of spatial relations, so that the displayed representation is complete and correct at each step.

The work by Lehto and Kilpeläinen (2001) is more dedicated to cartographic generalization in the narrower sense and concentrates on the real-time generalization processes for the visualization of spatial data in different environments, such as the web or mobile devices. The idea is the ad hoc display of maps out of a single detailed database, given a strong service and user orientation. As Lehto and Kilpeläinen argues, this approach needs mechanisms for real-time generalization which are drastically different from the traditional process. Since the possible scenarios also include the use of wireless devices (e.g. PDAs), the generated map must also take into consideration the location of the user. This means that only fast and easy to handle generalization operators can be implemented in such a context. In this case only selection, building simplification and area aggregation were implemented and tested on topographic data.

Harrie *et al.* (2002) proposed methods for presenting geodata for personal navigation in real-time to a mobile user. Real-time generalisation methods are described which are used to adapt the original cartographic data to the small-scale areas of the map. A prototype system of a variable-scale approach was created using the emerging XML-based vector-data standards (GML[12] and SVG), where the generalisation and scale-variations were performed in an XSLT[13] transformation.

Recently, two main EU projects[14] funded by the IST programme (Information Society Technologies Programme) have been started researching on on-the-fly generalization for displaying maps on small devices (e.g. PDA, Smartphones) in context with location-based services. The above projects and studies show that the topic of on-the-fly generalization become increasingly important for cartography and the map making process.

2.4 Multi-Scale Databases in Cartography

In the first part of this section a short introduction about database technologies and types will be given (2.4.1). The second part of this section discusses the use of multi-scale databases[15] in cartography (2.4.2).

2.4.1 Types of Databases

During the last decades, databases constantly enlarged their role as a key technology for the efficient realization of complex information systems. The background for this development

---

[12] GML (Geography Markup Language) is an XML based encoding standard for geographic information developed by the OpenGIS Consortium (OGC).

[13] XSL Transformations (XSLT) is a standard way to describe how to transform (change) the structure of an XML (Extensible Markup Language) document into an XML document with a different structure. XSLT is a Recommendation of the World Wide Web Consortium (W3C).

[14] *WEBPARK*: Geographically relevant information for mobile users in protected areas. More information can be found in Burghardt *et al.* (2003) or http://www.webparkservices.info/.

GIMODIG: Geospatial info-mobility service by real-time data-integration and generalisation, for more information see Sarjakoski *et al.* (2002) or http://gimodig.fgi.fi/ .

[15] The term *multi-scale database* will be introduced and illustrated in section 2.4.2 .

was the fact that progress in database technology allowed systems

- to administer larger assets of integrated information (megabytes or even gigabytes of information);
- to represent new information structures (commercial, scientific or statistic information);
- to access existing information in new modes (background processing, interactive search or remote database access in client/server architectures).

A database can be seen as a collection of logically related data stored together in one or more computerized files (Date 2000). The first concept to understand is the role of the *database management system* (DBMS). The DBMS handles the management of the underlying database, handling all communications between the user and the database. Requests for data are sent to the DBMS where they are processed and turned into actual reads and writes of the database engine. This makes the DBMS moderately independent of the underlying database engine. DBMSs can perform different tasks and accomplish various types of process. The primary function of any DBMS is to provide access to the huge amount of data found in separate databases. It is the job of the DBMS to make all these separate and distinct databases appear as though they were one to the user. DBMSs allow users to search for data, add, update and even delete data. Besides these basic functions, DBMS are also responsible for ensuring data security and integrity by blocking access to unauthorized sections of databases and logging all activities conducted by users on the databases (Date 2000). Figure 2.6 illustrates such a database system environment.

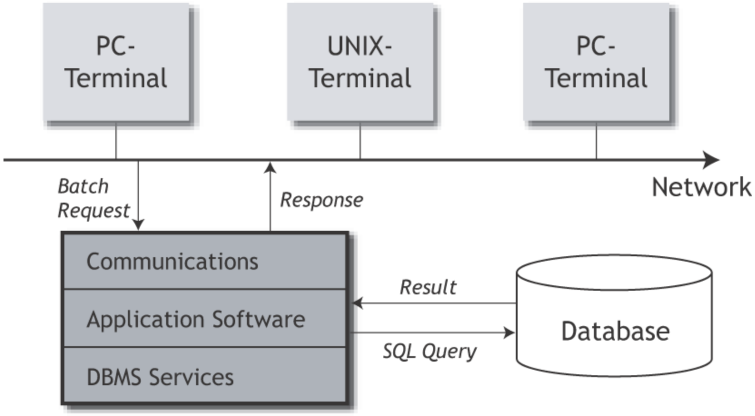


Figure 2.6: A simplified database system environment.

The data model determines the structure of a database management system and can assume different forms. Four main examples of data models can be defined: The *relational model*, *entity-relationship model*, *object-oriented model* and *object-relational model* will shortly be described in the next paragraphs.

Relational Model

The relational model was introduced by Codd (1970). It is based on a simple data structure, the relation. Within a relation, data is structured as a table of values. Each row of a table consists of a collection of related values and represents an entity of the universe of discourse. In relational terminology, the table is used as a synonym of a *relation*, where data is represented as a matrix of values. A row of a table is called a *tuple*, a column of a table is called an *attribute*, and the intersection of a row and a column is called *cell*.

Entity-Relationship Model

The Entity-Relationship (ER) model was introduced by Chen (1976), and describes data as entities, relationships and attributes. An *entity* is a 'thing' in the real world with an independent existence, for example, a department, a project or an employee. Each entity has *attributes* - the particular properties that describe it. An *entity type* defines a collection of entities that have the same attributes. The *relationship* type is a concept to define associations between two or more entity types, and is formally defined as a subset of the product of the participating entity types.

Object-Oriented Model

Object-oriented databases (OODB) have been influenced mainly by programming languages[16] and database systems. They are based on the object model and use the same conceptual model as object-oriented analysis, object-oriented design and object-oriented programming languages. The main characteristics of the conceptual model of object-orientation are (Martin 1993):

- **Object**: The concept of an object is the central basis for object-oriented systems. Although multiple definitions of the term object exist, it is normally defined by two parts: data and operations (whereby the term 'operation' and 'method' are equal). The data of an object is sometimes also referred to as its state and is implemented through attributes. The state of an object can only be changed via its operations. The effect of an operation depends on the current state of the object and the parameters.

- **Classes**: Objects are not described individually, but are categorized into classes instead. Every object is an instance of one class. A class description represents the template for all its instances, including the structure of the data and the implementation of operations.

- **Inheritance**: A class can be a subclass of another class. That is, the derived subclass inherits the attributes and operations of the base class, and can extend it with additional attributes and operations. Some systems support modification of the inherited attributes and operations in the subclass.

- **Polymorphism**: Polymorphism is a concept in type theory wherein a variable may contain instances of many different classes, as long as they are related by some common superclass. Any object denoted by this name is thus able to respond to some common set of operations in different ways. Another form of polymorphism, also known as parametric polymorphism, allows the declaration of multiple methods having the same name for a certain class, as long as their invocations can be distinguished by their signatures.

The fundamental argument in favor of OODBs, as compared to relational databases (RDBs), relies on a measure of data complexity (Desanti and Gomsi 1994). The need for OODB representations quickly increases with data complexity, which is (a) the absence of a

---

[16] Object-oriented languages can be divided into two categories, hybrid languages and pure object-oriented (OO) languages. Hybrid languages are based on some non-OO model that has been enhanced with OO concepts. C++ (a superset of C) is a hybrid language. Pure OO languages are based entirely on OO principles; Java and Smalltalk are pure OO languages (Baudoin and Hollowell 1996).

permanent and unique identification of the represented objects, (b) the existence of many-to-many relationships, (c) a complex relationships between object instances and object features, (d) the existence of a large variety of data types, (e) complex hierarchical structures between data types.

Object-Relational Model

Several database companies are promoting a new, extended version of relational database technology called object-relational database management systems also known as ORDBMSs. The main objective of ORDBMS design was to achieve the benefits of both the relational (RDBMS) and the object models such as scalability and support for rich data types. ORDBMSs employ a data model that attempts to incorporate OO features into RDBMSs. All database information is stored in tables, but some of the tabular entries may have richer data structures, termed abstract data types (ADTs). An ORDBMS supports an extended form of SQL[17] called SQL3 that is still in the development stages. The extensions are needed because ORDBMSs have to support ADT's. The ORDBMS has the relational model in it because the data is stored in the form of tables, having rows and columns, and SQL is used as the query language, with the result of a query being a table or set of tuples (rows). But, the relational model has to be drastically modified in order to support the classic features of object-oriented programming. Hence the characteristics of an ORDBMSs are:

- Base datatype extension;
- Support complex objects;
- Inheritance;
- Rule Systems.

A more detailed discussion of ORDBMS can be found in Stonebraker *et al.* (1999).

In the database community, GISs are primarily associated with spatial databases, and therefore a large amount of the research effort in databases for GIS is related to spatial data structures, access methods and data modelling (Worboys *et al.* (1990), Raper and Maguire (1992)). Concepts of object-oriented modelling applied to geographic data have been presented by Egenhofer and Frank (1992).

The rapid growth of GIS in recent times has resulted in a large number of systems, each of it with its own data handling and storage characteristics. Examples of these architectures for different types of DBMS are:

- *relational*: Data is usually stored in vector format and the data management is handled by two systems: the relational DBMS supports alphanumeric data, another system processes spatial data. An example is the ARC/INFO System[18].
- *object-oriented*: They are based on variations of the object-oriented paradigm, either using an object-oriented database or an object-oriented programming environment. An example is the Gothic LAMPS2 System[19].

Further information about databases, database management systems, and research issues in database topics can be found in the database literature (e.g. Date (2000)). For more GIS and database related issues Adam and Gangopadhyay (1997) and Rigaux *et al.* (2002) can be consulted.

---

[17] SQL (Structured Query Language) is a standard interactive and programming language for getting information from and updating a database. Queries take the form of a command language that lets you select, insert, update, find out the location of data, and so forth.

[18] Environmental Systems Research Institute, Inc. (ESRI).

[19] Laser-Scan Ltd.

2.4.2 Use of MSDB for On-Demand Mapping

As introduced in section 2.1.4, two main generalization strategies can be defined which are used as a basis for the generation of maps: a *process-oriented* and a *representation-oriented* approach. Methods and techniques for the first approach have been presented above, this section deals with the second.

Multi-scale databases are an alternative or complementary approach to solve the very complex process of deriving a generalized map from a detailed database. They can be described as a combination of different data sets representing objects at several map scales (see Figure 2.7). The term *multi-representation database* (MRDB) is often used in a similar manner by other researchers. However, since there are slight differences a short explanation must be given. Kilpeläinen (1997) defined a multi-representation database (MRDB) as consisting of different representation levels of the same area and objects but in different themes. Multi-scale databases, in contrast, store the same themes at different scale levels[20]. MRDBs may tailor data to meet particular thematic purposes while MSDBs are concerned more with topographic contents and can be considered as a subset of multi-representation databases. In this work the term of **multi-scale databases** (MSDB) will be used.

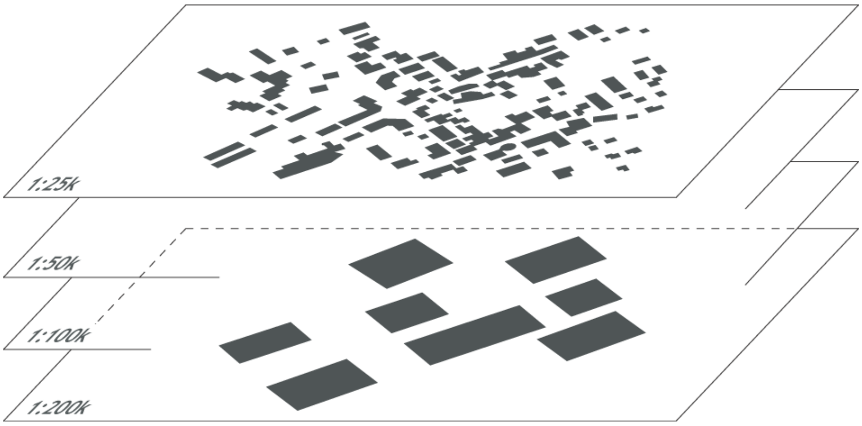


Figure 2.7: Schematic depiction of a multi-scale database with four LODs, representing the feature class *Buildings* (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

For the purposes of this thesis a multi-scale database is defined as consisting of several data sets, where these data sets contain objects stored at different pre-defined scale levels (e.g. 1:25'000, 1:100'000, 1:200'000) (Cecconi and Weibel (2001), Devogele *et al.* (1996)). These levels – commonly termed *levels of detail* (LODs) – are derived beforehand (off-line) by means of interactive generalization of the base data set. We distinguish two phases of the mapping process in regard to the generalization process: a) a preprocessing phase (carried out off-line), and b) an 'ad hoc' phase (carried out online):

- **Preprocessing**: The whole generalization process is accomplished off-line in the preprocessing phase. This allows the use of all possible methods and algorithms to solve the specific problems related to the scaling process. Examples of expensive methods are optimization techniques proposed by Bader (2001) or multi-agent systems such as proposed in the AGENT-project (Barrault *et al.* 2001).

---

[20] *Levels of scale* or *levels of detail* (LODs) can be used synonymously.

- **Ad hoc**: In the 'ad hoc' phase, no generalization processing takes place. Since the whole generalization has been carried out in advance (off-line) the mapping process is speeded up.

Next to the advantages of using a MSDB, the main disadvantage is keeping the database updated for all LODs. If the base data set is changed (e.g. new buildings, new roads are added) all the updates must be propagated to the other LODs. Methods and algorithms for doing this procedure automatically are proposed by Kilpeläinen and Sarjakoski (1995) and Harrie (1998) but have so far only be implemented to a limited extent.

In the late 1980s, the NCGIA[21] started a research initiative on multiple representation (Buttenfield and DeLotto (1989)) through a three-year research program (National Center for Geographic Information and Analysis 1989). The central issues were, the need to organize multiple topological and metric versions for efficient access and the implementation of linkages between representations. Beside this, other research topics like new strategies for maintenance of realized database views have been suggested (cf. closing report composed by Buttenfield (1993)). A special case study in combination with the research initiative has been made by Bruegger and Frank (1989) in relation to multipurpose GIS. For the display of spatial objects at several resolutions topological data structures are not enough and additional structures are needed. Bruegger and Frank proposed a dimension-independent approach for building multiple, hierarchically related representations of spatial objects. These representations are formed of layers of topological data structures which are connected by hierarchical relations[22].

Other reasons for forcing the research topic of multiple representation are given by Jones (1991). In many applications of GIS spatial data at different scales are required. But, since the full automatic generalization process is not sufficiently advanced yet, (to derive spatial data from a single large-scale data set) multiple representations must be maintained. Jones presented a concept of database design, based on a deductive knowledge-based system architecture, which provided a basis for building an experimental multi-scale spatial information systems.

Kilpeläinen (1997) was engaged with the problem of updating the data set of a multi-representation database (MRDB). She proposed the approach of *incremental generalization* derived from the so-called modular approach (Ledgard 1983) to resolve the task of updating MRDBs. The idea was to make the necessary updates at the base level of a MRDB and to propagate these updates to the other levels. The main principle is based on two assumptions (Kilpeläinen and Sarjakoski 1995): i) the generalization task must be divided into modules, where modules build a subset of geographical data; and ii) in an MRDB, each object should be defined as belonging to a conflict. Essential for the success of incremental generalization, is the ability to consider a module in isolation to the rest of the data set. She concluded that the division of the generalization task into modules and the question of how to find the borderline for the generalized modules to be updated still remains an open question (Kilpeläinen 1997). Harrie and Hellström (1999) presented a prototype for propagating updates between cartographic data sets in a multiple representation database, based on the conceptual framework of incremental generalization proposed by Kilpeläinen (1997) defining four steps: examination (checking the current status of the data and determining how the update will affect the target data set), propagation (execution the examination step), generalization of updates (generalization of new or modified objects), and solution of spatial conflicts.

The task of updating was also researched by Uitermark *et al.* (1998). They propose a solution involving two steps. First a selection of candidates of equivalent object representa-

---

[21] National Center for Geographic Information and Analysis. More information can be found under *http://www.geog.buffalo.edu/ncgia/* (accessed: February 2003).

[22] Hierarchical structures have often been proposed by several authors, e.g. strip trees (Ballard 1981) and quadtrees (Samet 1989).

tions on the basis of the thematic component should be made. Second, after synchronizing the databases, a geometric search is maded for equivalent instances using methods from the computational geometry, such as overlay operations. This presented search procedure has been implemented only for building objects.

Today, in most cases, MSDBs are built by integrating existing representations that have been captured independently at different scales (termed *data integration* (Uitermark 1996)). There are two main reasons for building a MSDB with existing data. On the one hand, creating derived data at arbitrary scales from existing data is associated with great effort in time and costs. On the other, the data volume is typically so large that it makes sense to re-use available data. This implies that originally no types of correspondence were defined between the various levels since the different data sets used to form the levels have been collected and built independently of one another. Devogele *et al.* (1996) pointed out that, for the design of a truly integrated MSDB, the transitions between scales must be modelled. This entails three types of problems:

- *Correspondence between abstractions*: Database schemata translate phenomena of the real world into abstracted instances in a database, by focusing only on relevant parts of these phenomena; integration of abstractions thus requires methods for schema integration at the semantic level.
- *Correspondence between objects of different representations*: Data models are required to describe the links between corresponding objects of the different representations.
- *Defining the matching process between objects*: In order to identify corresponding objects, two sets of geographical data must be searched to identify objects that represent the same real-world objects; methods for this are subsumed under the term *data matching.*

The work of Devogele and co-authors was focused on the first and last point (Devogele (1997) and Devogele *et al.* (1998)). In their work they focused on one critical issue raised by the integration of data sets of different scales. It concerns the aggregation conflicts that arise when one object of one level corresponds to a set of objects at another level. In their research, they intended to build a multi-scale database from two road databases[23]. In a first step they tried to achieve a common integration schema, since the two databases employ different definitions of feature classes. In a second step, the individual objects are matched geometrically. The matching problem is carried out in two steps. First by semantic criteria and then by a metric search using the Haussdorff distance.

The EU-project *MurMur*[24] was not directly concerned with cartographic generalization and MSDBs but more with the definition of a flexible manner to conceptually describe MRDBs. Parent *et al.* (2000) defined the objective of the project as to enhance GIS (or DBMS) functionality so that, relying on more flexible representation schemes, users can more easily manage information by using multiple representations. The added functionality would then support multiple coexisting representations of the same real-word phenomena (semantic flexibility), including representations of geographic data at multiple resolutions (cartographic flexibility).

More work on the geometric linking process of corresponding objects has been done by Sester *et al.* (1998). They showed that such a linking or matching process has many benefits. First, it helps to limit redundancy and inconsistencies, which helps take advantage of the characteristics of more than one data set and therefore greatly supports complex analysis processes. It thus opens the way to integrated data and knowledge processing using whatever

---

[23] The databases are from the IGN (Institut Géographique National): BDCarto (scale of 1:100'000) and GéoRoute (roughly equivalent to 1:10'000).

[24] MurMur (Multi-representations and multiple resolutions in geographic database) was part of the IST-programme of the European Union (http://lbdwww.epfl.ch/e/MurMur/).

information and processes are available in a comprehensive manner. Sester *et al.* proposed a solution to match two data sets of similar scales and used the functionality for deriving a representation of lower resolution from a highly detailed data set to perform database generalization.

The solutions presented by Buttenfield (1999) and Bertolotto and Egenhofer (2001) placed the focus of their research more on developing techniques for the transmission of vector data than on on-demand web mapping. But since they were working with fixed pre-defined levels, which are transmitted progressively to get a more detailed map, their solution is partially applicable here. The advantage of their work is that the full data set need not be sent immediately and then displayed, but instead coarser maps are sent the progressively get more and more precise.

Jones *et al.* (2000) focused their research on two major problems in context of dynamic map generalization, thus pursuing similar objectives to this thesis. The first is the problem of the design of multi-scale spatial data access schemes. The other is the design of online symbolization procedures that can render the data and ensure a legible visualization for the given display. The assumption is that well-designed multi-scale database schemes allow balancing the access speed with the storage requirements and organizing the data in a manner that avoids duplication of geometry components. A restriction of these multi-scale schemes, however, is that the representation has been 'pre-generalized' independently of other features, which can create spatial conflicts when the scale or symbology is changed. Jones *et al.* (2000) proposed applying triangulated-based techniques (Delaunay triangulation) to detect and solve spatial conflicts in a real-time process and thus preserve correct topology.

Another approach has been presented by Bédard *et al.* (2002). They developed a new solution for the modelling of spatial databases that support multiple representations and generalization processes in a spatial data warehouse. The tool used in their project, called *perceptory*, is a conceptual spatial data modelling application for designing object class models based on the UML object-oriented formalism and includes several components that allow the modelling of the spatial characteristics of cartographic objects in a simple manner. For visualization, the 'vuel concept' (view element) is introduced. Vuels are defined as the primitives of spatial database views similarly to the way pixels (picture elements) are defined as the primitives of digital images. It allows a system to go further than only taking into account the geometric aspect of the data without considering their graphic or semantic aspects.

Recently, a new project named *GEMURE*[25] has been started by the same group at Université Laval in Québec as well as other groups including our research group at the University of Zurich as international collaborator. The global objective of this project is to develop new methods and tools that better support the combinations of cartographic generalization operations with multi-resolution databases for easier on-demand cartographic information delivery, whether for Web mapping or paper products. The expected results are new data structures, techniques and tools and the introduction of cartographic patterns in the cartographic generalization process to facilitate on-demand cartographic information delivery.

The question now is no longer *if* MSDB are necessary for on-demand mapping but *how* they can be used for the real-time generalization process.

---

[25] GEMURE (Multi-scale generalization and multiple representation techniques for efficient production and delivery of cartographic data) is a project that started in 2002 within the GEOIDE Network (Geomatics for Informed Decision), funded by NSERC, the National Sciences and Engineering Research Council of Canada.

2.5 Conclusions and Assumptions

2.5.1 Conclusions

Multi-Scale Database

As outlined in this section, MSDBs do not solve the problem of automatic generalization. As with other techniques they possess advantages and disadvantages. On the one hand, they include prepared maps at different scale levels, which makes the generalization process unnecessary because it has already taken place before, but on the other, this fact implies that the fixed defined levels are not flexible in creating other map scales to those already existing in the MSDB, let alone creating maps for purposes other than those originally envisioned. Not to be underestimated is the updating and update propagation process of the several levels. Because changes of objects occur regularly, this leads to high costs in the maintenance of the various map scales.

In the context of on-demand (web) mapping, it is not sufficient to build a special multi-scale database with a many LODs as it is the case in many different Internet applications found today (e.g. www.mapquest.com or www.mapblast.com). The user's request is generally more complex and flexible than predefined layers allow for. To achieve the best flexibility in map theme and scale *combined strategies* using other techniques must be found.

On-the-fly Generalization

The presented definition of on-demand mapping in section 2.2.1 shows why on-the-fly generalization is such an important utility and will grow to an increasingly important method in the web mapping process. Two main points are essential: on the one hand, the better utilization of the client-/server-architecture and on the other hand, the possibility of being able to display and also manipulate vector graphics.

In the research area of on-the-fly or real-time generalization, little work has been done yet and it is still very much a research issue. Really successful solutions are not available. First approaches have been presented by Oosterom van (1995) and Rusak Mazur and Castner (1990). Starting from the existing methods the focus must mainly be set on approaches based on *hierarchical data structures*. More recent research, above all in the area of *location based services*, shows first basic approaches that could appropriate possible solutions.

Since automated generalization is still in an immature state and cannot be used for on-the-fly generalization alternatives must be found. As shown in table 2.5 some generalization operators are more likely to be suitable for that purpose than others. Especially for the ones not particularly suited for this process alternative solutions must be found.

2.5.2 Summary and Assumptions

The different examples show that a fully automated generalization process, especially in the context with on-demand web mapping, is not yet feasible, because various corresponding tools and algorithms do not exist at this time and will probably not be developed in the near future. Thus, it is important to focus on new approaches by trying to substitute missing algorithms through other methods and possibilities.

The approaches discussed above all give preference to either multi-scale databases or on-the-fly generalization. Advantages and disadvantages of the process-oriented and representation-oriented views are listed in section 2.3. However, in order to better exploit the advantages of both approaches, it would make sense to develop a combined solution that lies between these two poles. These two poles can be set by defining:

- a structure for a MSDB and hierarchical linking;
- an adaptation of the generalization process.

The aim of this work is to develop methodologies and techniques that allow the generation of a user-defined map from a multi-scale database. The main focus is on the *generalization component* in the map creation process, meaning that the quality of the maps must be adapted to the requirements of the user requests. The user themself can influence – by the selection of different attributes such as scale or definition of the symbolization – the complexity and cartographic quality of the target map, and hence the time required to generate the result.

Assumptions

To simplify the conditions for this work, a few main constraining assumptions have been made. Firstly, the provided data set from the Federal Office of Topography has been cleaned up and processed to be used for further applications. Secondly, only one base database (defining a multi-scale database) is used. No other auxiliary databases are combined for the mapping process. The data models VECTOR25 and VECTOR200[26] compose the base data set and need to be consistent. So only two data sets are defined to build up the multi-scale database, whereby for all feature classes are represented for the same levels (describing the same scale). So as not to complicate the work too much, only the feature classes *road network*, *building* and *river network* are taken into account.

---

[26] VECTOR25 and VECTOR200 define the data models for the scale of 1:25'000 and 1:200'000 and are provided by the Federal Office of Topography.

Chapter 3

A Conceptual Framework for On-Demand Web Mapping

3.1 Overview

A description and a definition of on-demand web mapping were given in section 2.2.2. In this chapter a *conceptual framework* for on-demand web mapping will be presented.

The main idea is to combine two well known approaches in digital cartography – the use of multi-scale databases and the application of a generalization process – into one system. Different research foci can be set to optimize the interaction between these two methods to obtain a smooth workflow. Primarily, however, it is important to find the trade-off between these two poles to look for an optimized solution for on-demand web mapping.

In the first section of this chapter an overview of different scenarios occurring in the web mapping context will be given. The conceptual framework and the units pertaining to the main *GENDEM unit*[1] are discussed in sections 3.2 and 3.3. Section 3.4 investigates the behavior of the generalization operators used in different scale bands for different feature classes. The strategy of the generalization process in context with on-demand web mapping is treated in the last section of this chapter (3.5).

3.1.1 Introduction

As was discussed in section 2.3 different approaches for on-demand web mapping are known and used today. However, all of them have considerable limitations regarding flexibility and are not adaptable enough to fulfill the varying requirements of potential users. They also pay little attention to aspects of scale changing and map generalization. To remedy this drawback a new strategy must be found which allows more flexibility for the map creation process.

Cartographic generalization plays a very important role in the overall map-making process so that it critically determines how long it takes to create a map. But since fully automated generalization is only partially solved with respect to available methods and algorithms other solutions must be found a) to substitute missing automated generalization operations and b) to speed up cartographically appropriate yet computationally intensive generalization algorithms.

The selected approach – the combination of multi-scale databases and cartographic generalization – offers a possible solution combining the advantages of both methods (Cecconi and Weibel 2001).

---

[1] GENDEM: *Gen*eralization for on-*de*mand *m*apping.

Using a multi-scale database the automated generalization process can be divided into two phases:

Multi-Scale Database ⟶ i) Pre-computed Generalization
ii) Real-time Generalization

First, time consuming generalization algorithms can be **pre-computed** in advance and off-line where the computation time is more or less irrelevant. Every feasible and existing algorithm can be included in the generalization process to achieve the best possible cartographic visualization of the data set for different pre-defined scales. The results are then stored as levels of detail (LODs) in a multi-scale database. On the other hand non-existing generalization algorithms can be substituted (i.e. simulated) by means of interactive operations used to build the multi-scale database.

Second, those generalization algorithms that are computationally efficient can be used to compute **real-time** (on-line) generalization and to refine the nearest LOD to the requested intermediate map scale. By means of this combined process on-demand map creation can be speeded up and made more flexible at the same time.

In summary it can be said that the use of such a multi-scale database allows to compensate for the drawbacks caused by the automated generalization process not yet entirely solved. But since the interests and the purposes of the users are strongly variable, different types of scenarios concerning the mapping process and thus the 'quality' of the generalization process must be defined in a first step.

3.1.2 Definition of Usage Scenarios

Different requirements and requests for web mapping on the part of the users constrain the definition of scenarios for cartographic purposes (Gardels 1998, Elzakker van 2001). For instance, some users prefer more cartographically pleasing maps at a certain scale while others are satisfied with a simple map which appears in the shortest possible time. All these options and possibilities must be condensed to a small number of scenarios with typical characteristics. The focus of this arrangement has been set on the *complexity of the generalization process* and hence in the time required for map creation. Based on this, three main scenarios can be distinguished and are summarized in Table 3.1.

The difference between these three types is based on the kind of the map creation, focusing especially on the generalization process. The distinction is made between an 'on-the-fly' and 'on-demand' sequence of the mapping process. They all start from existing LODs in the multi-scale database followed by an adaptive generalization process generating the requested map. The first and third scenario represent the 'worst' and the 'best' case in terms of cartographic quality in on-demand web mapping, while the second one lies somewhere between these two extremes. On the basis of *parameters* and *preferences* defined in advance by the user the desired visualization will be generated.

Scenario 1: *'Quick and Dirty'*

Scenario 1 describes the easiest way of creating a map within the Internet context. It assumes that a lot of users want to have an initial coarse map to give an overview about the requested information. Starting from the nearest LOD a map with the demanded scale will be produced by a simple scaling transformation. The process takes place on-the-fly *not* including cartographic generalization operations besides the symbolization process and allows one to display the result within a few seconds of the transmitted request. By means of the initial result a more generalized map, scenario 2 or 3, can be requested if desirable.

Scenario	Description	Map generation type	Flexibility
Scenario 1	On user request an initial map must appear. The compute time must be very short.	on-the-fly	less flexible
Scenario 2	The created map includes methods, which means good quality in a limited amount of time.	on-the-fly and on-demand	↓
Scenario 3	The time component plays a minor role. A high-quality map is requested on users requirements.	on-demand	more flexible

Table 3.1: Description of different scenarios for the on-the-fly and on-demand web mapping process depending on the complexity of the generalization process.

Scenario 2: *'Map for the Web'*

This scenario combines the advantages of scenarios 1 and 3. On the one hand a map has to be created in a limited time and on the other hand generalization is applied to obtain an appropriate cartographic visualization. It is important to balance between these two elements to create cartographic products at varying scales.

Scenario 3: *'Map for Print'*

In this scenario the obtained result is a 'perfect' cartographic product generated especially to be printed on paper. Because the user wants to have a high quality map and thus can be expected to be willing to accept a longer production time, the computation time component plays a minor role[2]. That implies that the entire generalization process can be applied starting from one detailed data set to obtain a high-quality product including all existing generalization algorithms. For very complex situations which cannot be solved by the automated process an expert with cartographic knowledge can even be consulted.

All three scenarios have their justification in the context of on-demand web mapping whichever is tailor-made to user specifications. Scenario 1 and 3 describe two extreme poles where on one side nearly no generalization takes place and on the other side a full process is applied. These two scenarios will not be further discussed in this thesis. The focus is placed on the second scenario optimizing the generalization process according to the specific requirements of web mapping.

---

[2] The idea is that the user has not to wait in front of the screen for the result but that he/she gets informed by a message as soon as the map is finished.

3.2 Conceptual Framework for On-Demand Web Mapping

In the next two sections (3.2 and 3.3) the complete conceptual framework for on-demand web mapping will be presented. Starting from the architecture of the system the environment and the different units and modules of the framework are introduced.

3.2.1 Architecture of the System

On-demand mapping is essential in distributed systems such as the Internet or local area networks. The main reason is that different data sources must be used and referred to for the generation of maps (OpenGIS 2000). For the purpose of having access to different databases a convenient solution for the architecture of the environment must be adopted. The *client/server* architecture accomplishes all the required conditions for on-demand web mapping. A client is defined as a requester of services, while a server is defined as the provider of services. A single machine can be both a client and a server depending on the configuration; in most cases, however, they are placed on different machines connected by a network (e.g. Internet). Some of the most popular applications on the Internet following the client/server design are: Email clients, FTP clients or web browsers. A detailed description can be found in Gallaugher and Ramanathan (1996).

In contrast to a two tier architecture, where client and server build a layer the **three tier architecture**[3] includes a third tier (middle tier) between the user interface (client) and the data management (server) components (see Figure 3.1).

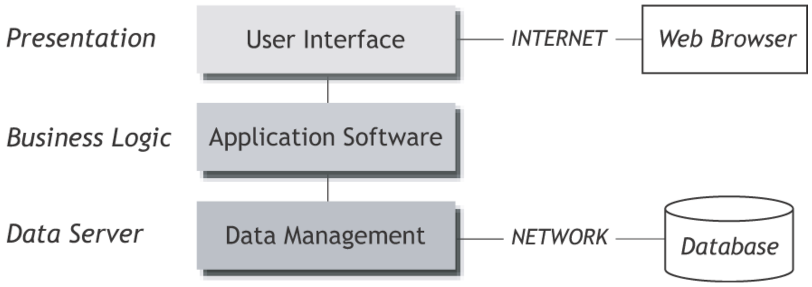


Figure 3.1: Example of a three tier distributed client/server architecture.

The user interacts with the graphical user interface (*presentation tier*), an integrated collection of menus, toolbars, and other controls. This enables to see requested results and/or fill in forms on a client software (e.g. web browser). The middle tier provides process management where *business logic* and rules are executed by providing functions such as queueing, application execution, and database staging (Edelstein 1994). The third tier is usually best satisfied with a database on which the different data sets are stored and organized by the *data management software*. This data management component ensures that the data is consistent throughout the distributed environment (Szyperski and Murer 2002).

---

[3] Three tier architectures have been used successfully since the early 1990s on thousands of systems of various types and are well known in information technology. Further multi tier architectures have been widely and successfully applied in some of the biggest Internet servers (Edelstein 1994).

3.2.2 Embedding of the Framework

On the basis of a three tier architecture a conceptual framework for on-demand web mapping will be presented. For this work all elements of the conceptual framework – the presentation, the business logic and the data server – are located on one server called *MapGenServer*.

The interaction with the user/client is guaranteed by a so-called *client viewer*, which establishes the communication with the user and provides a visualization capability. This viewer is located on the MapGenServer but can otherwise also be connected through a network. In this combination the client viewer can assume quite different forms (e.g. a web browser or a local application, java applet) and offers the possibility to display the desired information in form of a cartographic product. In addition there is the possibility to form a module of a distributed GIS, which would generate maps or visualizations for on-line display. A further possibility for the client viewer is given by downloading the created result in a format chosen by the user in advance (e.g. postscript, pdf, tif).

The application software located on the MapGenServer possesses the facility to access different databases processing the data for the creation of a requested product. The success of this process relies on the GENDEM Unit (see Figure 3.2 introduced in the next section) which handles the information requested by the user with his/her preferences defined beforehand, generalizing the data set to the desired cartographic product.

3.3 GENDEM Unit

The principal item of the conceptual framework is formed by the GENDEM Unit which is itself composed of a collection of several minor units focusing on special functions and services. The different units interact with each other to respond to a user request by extracting the necessary data from the multi-scale database. Depending on their function and based on the three tier architecture three units can be distinguished: the FrontEnd Unit, the Multi-Scale Database Unit, and the MapMaking Unit. Illustration 3.2 provides an overview of the minor units of the GENDEM Unit and how they provide access to other components as well as the interaction with the different databases.

All of them are specialized and engineered to accomplish assigned functions and services. Additionally the MapMaking Unit contains different components needed to run services to create the requested product out of the data set accessed in the database. Two modules build the core of this MapMaking Unit: the ScaleChanging, and the Generalization Module. In the following the different units and modules will be presented respective to their functionality.

3.3.1 FrontEnd Unit

The FrontEnd Unit forms the connection between the user interface and the system, taking care of the entire communication flow. On the one hand it takes the queries and requests defined by the user as well as preferences and technical constraints influencing the final map product. In a further step all the gained information will be processed and prepared to be passed to the next unit, the Multi-Scale Database Unit. On the other hand it is also responsible for the correct output of the results, depending on the specifications set by the user in advance. In principle, the result can take different forms, from a print-prepared map to a coarse cartographic visualization or a vector data product. In the final step the output will be sent to the user and displayed in the client viewer. The implementation of this unit can be based on the specifications defined by the Open GIS Consortium (OGC)[4], in which way maps are requested (see more in OpenGIS (2000)).

---

[4] OGC is an international industry consortium of more than 230 companies, government agencies and universities participating in a consensus process to develop publicly available geoprocessing specifications (http://www.opengis.org).

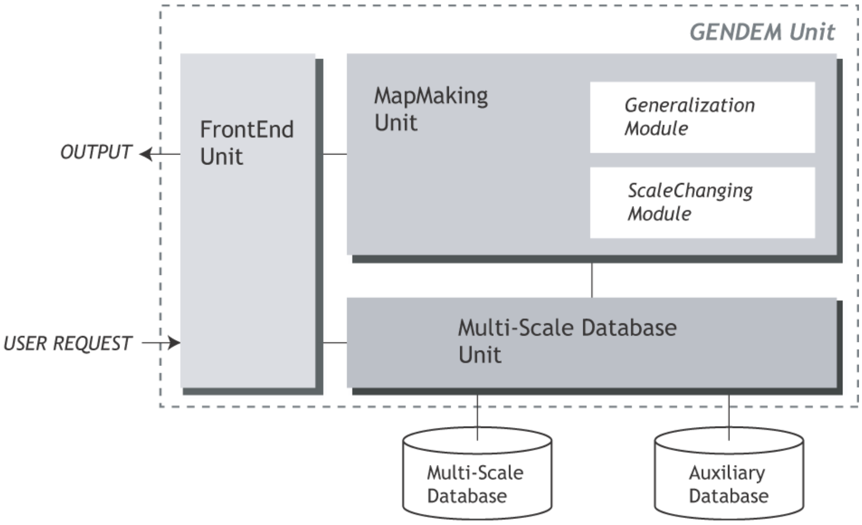


Figure 3.2: The GENDEM Unit located on the MapGenServer is composed of three minor units: the FrontEnd Unit, the Multi-Scale Database Unit, and the MapMapking Unit and based on a three tier architecture commonly used in information technology.

3.3.2 Multi-Scale Database Unit

The main task of the Multi-Scale Database Module is the handling of the requested data from the multi-scale database. For this step the information sent from the FrontEnd Unit must be handled to retrieve the appropriate data from the database. The example depicted in Figure 3.3 demonstrates schematic of such a possible workflow. Next to the 'main' database the possibility must be available to access other multi-scale databases connected through a network.

Figure 3.3 shows the simplest case, where the requested map scale is equivalent to a LOD stored in the multi-scale database. In most cases, however, a different situation occurs where the requested map scale is not directly available as a LOD in the database but situated between two other LODs. In this case the LOD with the next smaller and next larger scales will be accessed and processed to generate the demanded map product. For example, if the desired map scale is 1:75'000 the two next LODs (concerning the map scale) are taken to be processed. In this case the LODs of 1:50'000 and 1:100'000 have to be activated and returned as result for further processing.

$$\text{User Request}\{1{:}75'000\} \quad\longrightarrow\quad \text{Response }\{\text{LODs } 1{:}50'000 \text{ and } 1{:}100'000\}$$

The number of LODs stored in the database for on-demand web mapping cannot be fixed for all possible cases but must be adapted to different scenarios and purposes. Three main criteria do affect the choice of an appropriate number:

1. On the one hand the number must be as *large* as possible. This allows minimization of computation and thus of the mapping process time;

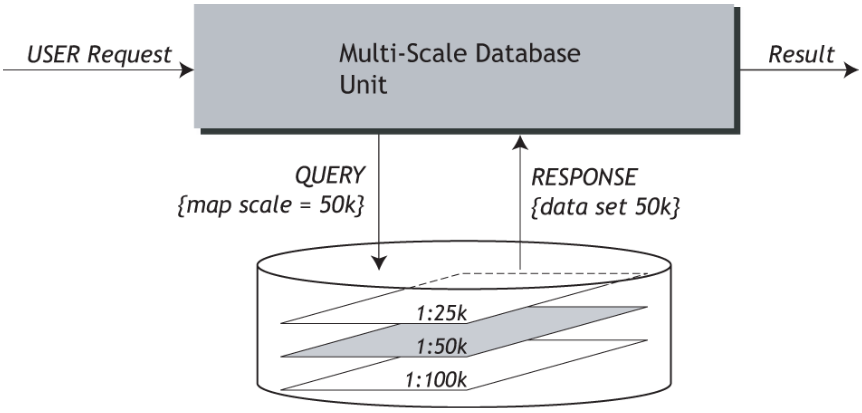


Figure 3.3: The illustration shows the tasks of the Multi-Scale Unit. The multi-scale database consists of data sets with different map scales, so-called 'Levels of Detail' (LODs).

2. On the other hand the number must be as *small* as possible to reduce redundant information and thus simplify the database updating process;

3. The third point is the fact that the number of LODs must be optimized to not have too large ranges between each LOD, so that the creation of a map scale lying in between is possible.

Obviously, these criteria are in conflict with one another and illustrate the strong case dependence of the choice of the number of levels. Another fact that should not be underestimated is the restriction of available data sets. In fact most of the National Mapping Agencies offer only a small number of data sets which rarely agree with the preferred LODs of web mapping[5].

For this work the smallest possible number of levels has been selected to build up a multi-scale database prototype. Based on the data sets available from the Swiss Federal Office of Topography the scale levels of 1:25'000 and 1:200'000 are taken to be implemented in a multi-scale database (for more detailed information about the data sets see section 4).

3.3.3 MapMaking Unit

The third and last unit, the MapMaking Unit, represents the core of the whole conceptual framework and thus the main unit of the system. In this component the cartographic product specified by the user is generated from the selected data set retrieved from the multi-scale database. A first step is the definition of a scale dependent *symbolization*.

For the second step, the processing of the data concerning the target map scale, two modules are responsible: on the one hand a simple scale changing transformation defined by the ScaleChanging Module, on the other hand a generalization process realized by the Generalization Module. Depending on the situation one of these two modules is used.

---

[5] The Swiss Federal Office of Topography, for instance, only have two landscape models in vector format: *VECTOR25* based on the national map of 1:25'000, and *VECTOR200* based on the map of 1:200'000.

ScaleChanging Module

As shown in section 3.3.2 a multi-scale database consists of several LODs, containing a minimum of two. Thus it is not always necessary to invoke a generalization process to create the demanded map. Two different situations can be defined where this module can be employed:

- The ScaleChanging Module is applied if the user-selected scenario (see Table 3.1) is equivalent to the first one (Scenario 1: 'Quick and Dirty'), where a map product is requested which is not cartographically generalized.
- If the requested map scale RS_U is within a defined tolerance ε to one of the given LOD_i (n_l is the number of LODs) where

$$\varepsilon < |RS_U - LOD_i| \qquad \forall i \leq n_l, n_l \in \mathbb{N}, \tag{3.1}$$

a simple scale change not including a generalization process is sufficient. In both cases a scale-dependent *symbolization* is applied to display the demanded information.

Generalization Module

For all other situations where the ScaleChanging Module is not applied, including scenario 2 and 3 (see Table 3.1), and all cases where the equation (3.1) is not fulfilled the Generalization Module comes into operation. Depending on the selected scenario an adaptive generalization process takes place to create the requested product. This can be on the one hand a simplified automated process (scenario 2) or on the other hand an extended generalization process not limited by the time component or the restriction of available generalization algorithms. For the following discussion of this work the focus is on the second scenario.

This module is, as the name implies, responsible for the generalization process of the original data set according to the scale requested by the user. On the basis of map purpose, user requirements and technical restrictions defined beforehand the data of the selected LODs must be generalized under the retention of the minimum separability distances (Spiess 1990) of the target map scale. Everything that is done in connection with cartographic generalization happens in this module. Generalization strategies for the different feature classes are discussed in section 3.5. First, however, we will discuss the issue of defining appropriate levels of detail.

3.4 Definition of Levels of Detail

The conventional manual generalization process is of a holistic nature. In contrast to the conventional process it is almost impossible to develop an operational solution for automated generalization, at least for the time being. Hence, the whole process must be subdivided into a sequence (allowing also feedback loops) of generalization operators (cf. section 2.1.6). As several authors propose different generalization operators it is not possible to define a perfect set. This research work is based on the operators defined by McMaster and Shea (1992) and Bader *et al.* (1999).

The use of generalization operators for the creation of maps is fundamentally dependent on the scale(s) and feature classes involved (e.g. road network, hydrography). Dependent on a certain scale different operators (e.g. selection, simplification, typification and displacement) are applied in various sequences. There is no fixed order to using operators, but only some reasonable successions from the point of view of a cartographer. Starting from studying an existing map series representing different scales within the same area[6], the generalization

---

[6] The Federal Office of Topography provides a series of six different map scales of the National Maps of Switzerland as so-called pixel maps (pixel maps are scanned originals of the printed maps in a raster format): 1:25'000, 1:50'000, 1:100'000, 1:200'000, 1:500'000 and 1:1'000'000.

process accomplished and especially the operators used between each of these scales can be extracted. The knowledge gained thereby can be used to design the contents of a multi-scale database. In particular it should be attempted to answer the following questions:

- **Levels of detail**: Which and how many levels are necessary to guarantee an operational on-demand mapping process?
- **Limits of applicability**: Up to which scale can the defined LOD be used?
- **Generalization operators**: Which operators can be used to transform a LOD into another?

Obviously, all these decisions must be made separately for each feature class (e.g. road network, building), depending widely on the symbol specifications used (e.g. whether built-up areas are shown by individual buildings or simply as a tinted area). So, how can the above elements be defined in a meaningful way? The idea is to minimize the number of LODs, and extend the limits of applicability as far as possible. Each additional LOD implies significant additional costs, not only during database creation but even more importantly during database updates (which need to be consistently propagated across LODs). To answer this question the underlying idea is to analyze the *generalization complexity* as well as the application scope of the generalization operators over the desired range of scales.

3.4.1 Definition of Limits of Applicability

As explained in the previous section the multi-scale database (MSDB) is used as a base element for the generation of on-demand maps. The MSDB includes at a minimum two data sets of geographic objects represented at two different levels of detail or scales. In order to create a map at the scale desired by the user the appropriate LOD has to be determined. To this end, so-called **limits of applicability** (LA) are defined. These pre-defined limits can vary from feature class to feature class and depend on the given levels of detail. Figure 3.4 shows the selection of the corresponding data set for the desired map scale on the basis of these limits.

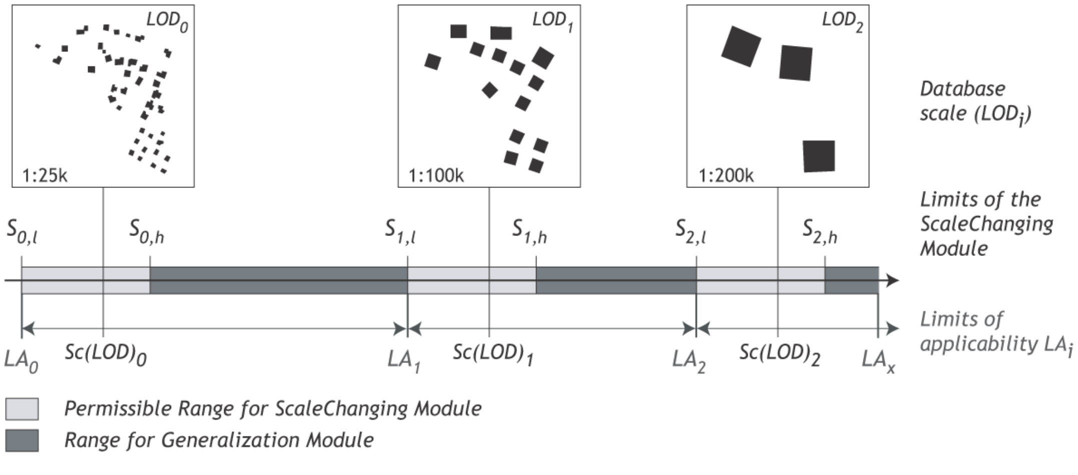


Figure 3.4: Definition of limits of applicability illustrated for the LODs of the feature class *building*. Data: VECTOR25/200 © swisstopo (BA034957).

Figure 3.4 illustrates an example with three LODs representing the feature class *building*: 1:25'000, 1:100'000 and 1:200'000. Based upon the assumption that the depiction of individual buildings is possible to a scale of 1:300'000[7] the scale range defined for this example is limited from 1:25'000 to 1:200'000 and illustrated by the light gray ranges, where the ScaleChanging Module is applied and in dark gray bands where the Generalization Module operates. For example the data of the LOD_0 1:25'000 can be used from LA_0 to the scale LA_1. Within this range two sections are distinguished: between the value of $S_{0,l}$ $(=LA_0)$ and $S_{0,h}$ the LOD can be directly submitted to the ScaleChanging Module (see also (3.1)). From $S_{0,h}$ to $S_{1,l}$ $(=LA_1)$ the Generalization Module transforms the data of the given LOD. The same partitioning is repeated for all other LODs (1:100'000 and 1:200'000) by defining the adequate limits LA and S.

This classification allows one to define a workflow for the selection of the LOD and the module corresponding to the scale requested by the user (RS_U):

$$\begin{array}{lll} \text{Selection LOD}_i\text{:} & LA_i \leq RS_U < LA_{i+1} & \forall i \leq n_l, n_l \in \mathbb{N} \\ \\ \text{Selection Module:} & \text{if } (S_{i,l} \leq RS_U < S_{i,h}) & \rightarrow \textsf{ScaleChange Module} \\ & \text{if } (S_{i,h} \leq RS_U < S_{(i+1),l}) & \rightarrow \textsf{Generalization Module} \end{array}$$

The definition of the range of these limits of applicability is important for the selection of the best available LOD. Note that the LA_i are asymmetrically defined about the corresponding LOD, with a bias towards using high-resolution data as far as possible. The underlying idea is to transmit a map with the highest possible information density to the user.

3.4.2 Studying the Complexity of the Generalization Process

The question of which scale and how many LODs must be defined to obtain the best possible system for on-demand web mapping is still unanswered. A possible approach to determining the number and the scales of the LODs is to study the generalization process over a large scale range. The idea behind this study is to better comprehend the generalization work performed by cartographers and to describe which operator acts in the different steps of the procedure. Dividing the full range into smaller scale bands would allow to define zones where specific generalization operators act. With this arrangement it should be possible to answer the questions of where to fix the different LODs and hence determine the appropriate minimal number.

The behavior of the generalization process over a larger scale range can be illustrated by means of the *generalization complexity.* This generalization complexity combines all the 'costs' involved in the process and provoked by the deployed operators. Low values indicate simple operators can be used (e.g., smoothing and simplification) while high values hint at major modifications requiring complex contextual operators such as typification or displacement. Analyzing a sequence of maps representing different scales of the same area (cf. Figure 3.5) and evaluating them according to the cartographic literature allows to establish the generalization operators used to describe the complexity of the process. While this is merely a qualitative method, it is nevertheless systematic and bears some resemblance with the *points de généralisation* observed by Ratajski (1967).

The starting point of the study builds the paper map sequence of the Federal Office of Topography representing different map scales: 1:25'000, 1:50'000, 1:100'000, 1:200'000, 1:500'000. For the examination of the generalization process and to understand what kinds of transfor-

---

[7] In the Swiss National Map series the layer *building* represents building objects up to a scale of 1:300'000. For smaller scales a settlement area is depicted.

mations are carried out between each scale level several test areas were defined in advance. Figure 3.5 illustrates a clip of an investigated area showing maps in the four different scales.

Figure 3.5: Maps from the Federal Office of Topography representing different scales. Upper left: 1:25'000; upper right: 1:50'000; lower left: 1:100'000; lower right: 1:200'000 (not to scale). Data: © swisstopo (BA034957).

The selected areas represent very general situations showing as many feature classes as possible. By means of the evaluation of the map series it has been tried to define the complexity of the generalization process for every feature class over the full scale range. For the five feature classes *road network*, *building*, *river network*, *lake* and *railroad* an overview is given below illustrating the behavior of the generalization process over the different scale bands. Figures 3.6, 3.7 and 3.8 show the costs arising from creating the smaller scale out of the larger one and describe the possible operators needed to obtain the result. The following equation formalizes the sum of the costs accruing in one scale band, where n is the number of operators involved in the scale band SB:

$$Costs_{SB} = \sum_{i=1}^{n} Cost_{Operator_i}.$$

Road network The complexity curve[8] for the *road network* feature class suggests three scale bands (see Figure 3.6). In the first scale band (large scales down to 1:50’000) the selection and simplification operators are sufficient because there is usually enough room for displaying all road objects without much modification. In the intermediate scale band (1:50’000 to 1:300’000) important shape changes become noticeable. Because there is still enough map space most road objects can be displayed (except in built-up areas), but must be transformed due to increased symbol width. This will be achieved by the operators typification (transformation of an initial set of objects into a new set, maintaining the typical arrangement) and displacement. Hence, the generalization complexity increases. In the third scale band the available space is so small that only the most important road objects appear; as the number of objects is lower, typification and displacement are becoming less important and generalization complexity decreases again.

Building The scale range for the feature class *building* can also be divided into three main scale bands (cf. Figure 3.6). The main break point (between scale bands 2 and 3) is the transition from individual buildings to tinted polygons for the display of built-up areas. On Swiss topographic maps this happens at 1:200’000. A further break point can be found at 1:50’000. Up to that scale, the main operators are elimination of small buildings (i.e. selection of large buildings) and simplification of building outlines, while typification of groups of buildings and important displacements only come into play later, significantly increasing the generalization complexity.

River network For the feature class *river network* (see Figure 3.7) most objects can be drawn as polygons up to 1:50’000. From this point on nearly all objects need to be collapsed to single lines. This step frees up map space near the line objects and thus decreases the potential of overlap conflicts. As the collapse operation actually creates space the complexity in the intermediate scale band is lower than for other feature classes. In the last section the displacement caused by overlapping objects becomes more important whereas most objects are collapsed by now.

Lake A special behavior is exhibited by the feature class *lake* (see Figure 3.7) with high complexity at large scales, low complexity at medium scales and large complexity at small scales. The large complexity in the first scale band is due to collapse operations of small inlets of rivers leading into the lakes. The complexity of the last band originates from the enlargment/exaggeration of smaller lakes which must be represented by reasons of importance. In the middle section the generalization complexity is influenced only by selection and simplification operations which results in lower values.

Railroad The feature class *railroad* shows a different behavior with a narrow band at large scales (cf. Figure 3.8). The large complexity of the first scale band is due to merge and collapse operations in station zones and switchyards, while mainline tracks do not pose a problem due

---

[8] The curve illustrates the first derivative of the summarized costs over the full scale range.

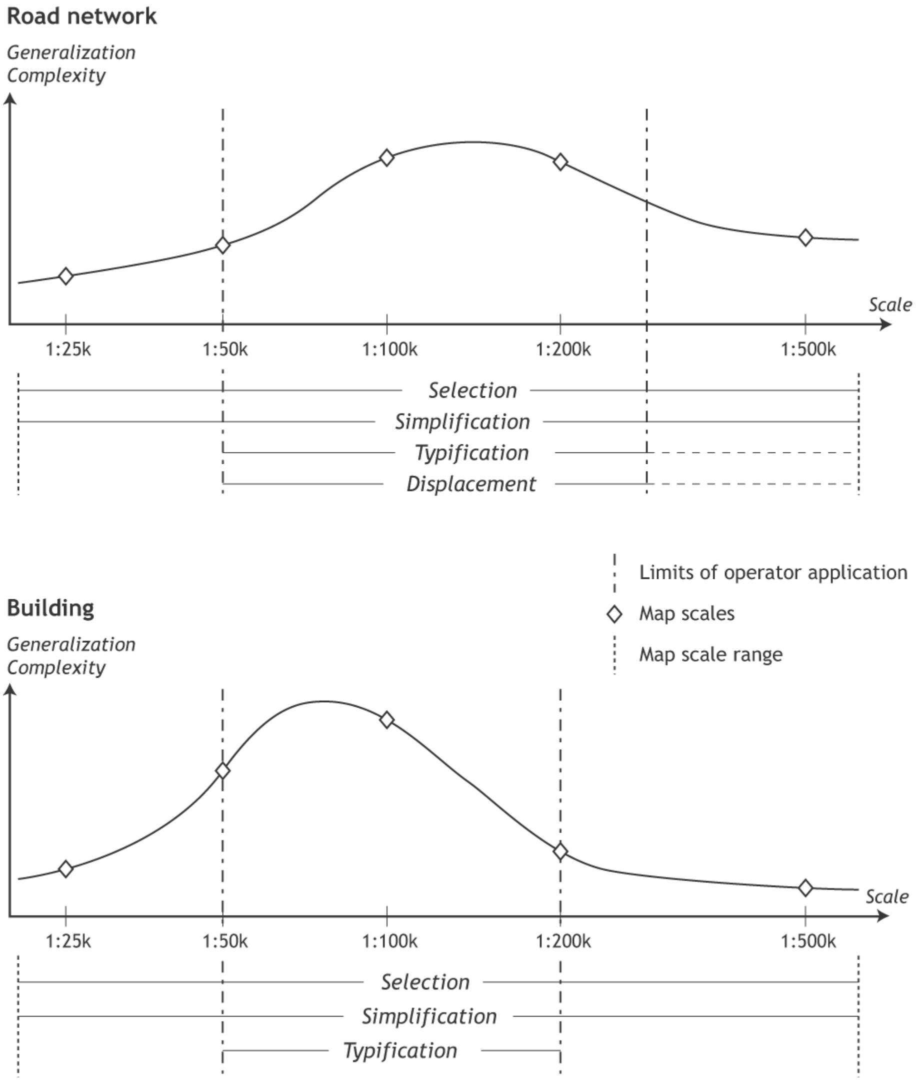


Figure 3.6: The 'generalization complexity' and the application scope of generalization operators for the feature classes *road network* and *building*.

to large curve radii. It is only later, in the intermediate scale band, that mainline tracks need to be generalized by various operators. Finally, at small scales beyond 1:200'000 only a small number of objects remain, decreasing the generalization complexity.

3.5 Strategy of the Generalization Process

Figures (3.6)-(3.8) in the last section show that the generalization process for each feature class does not only include one but several generalization operators over a large scale range. Since for some operators corresponding algorithms are available other solutions and approaches

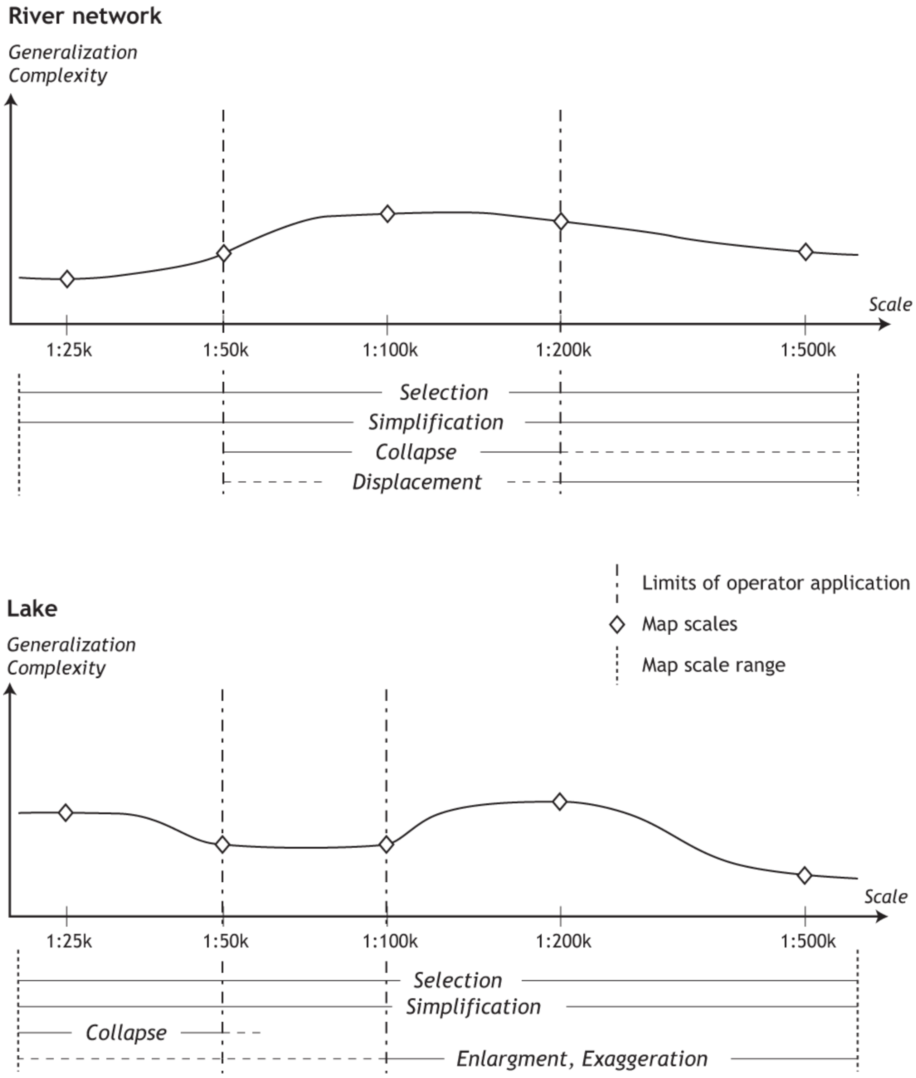


Figure 3.7: The 'generalization complexity' and the application scope of generalization operators for the feature classes *river network* and *lake*.

must be found to substitute the missing ones. Especially for on-demand web mapping where computing time restrictions are more constraining, less operators and algorithms are available to accomplish the requirements defined in section 2.2. Even though for every operator corresponding algorithms are not yet available the long-term task of web mapping and of course of cartographic generalization is to integrate newly developed algorithms to optimize the overall mapping process.

As presented at the start of this section three different scenarios concerning on-demand web mapping are conceivable (see section 3.1.2): a) *Quick and Dirty*; b) *Map for the Web*; and c) *Map for Print*. For scenarios two and three an adaptive map making and thus cartographic generalization process must be defined first. As the focus of this work is set on the second

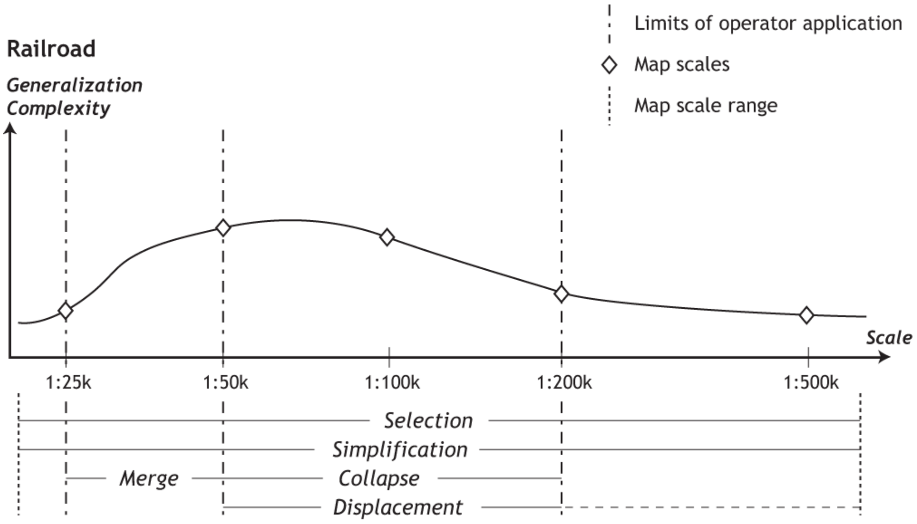


Figure 3.8: The 'generalization complexity' and the application scope of generalization operators for the feature class *railroad*.

scenario only for this one an appropriate procedure will be proposed.

A possible delimitation consists in the choice of operators pertaining to the context-independent group (see more in section 2.1.6). These operators are applied in most cases to objects or groups of objects without an antecedent examination of the spatial context. A first selection of generalization operators for the on-the-fly generalization is proposed by Lehto and Kilpeläinen (2001). Based on the context-independent group and on Lehto and Kilpeläinen's choice an enhanced list has been determined suitable to be used in on-demand web mapping and illustrated in Table 3.2.

Must be included	Nice to be included
Selection	Typification
Simplification	Displacement
Smoothing	
Morphing	

Table 3.2: Generalization operators and other methods suitable for the on-demand web mapping process.

On the left hand side of Table 3.2 the four mentioned operators *must* be included and applied in the generalization process. For the two on the right side it would be nice if they could be available since no specifically adapted solutions for on-demand web mapping are available at the present time. A possible approach for realizing *typification* for the feature class 'building' will be presented in section 5.3.2. Implementations for *displacement*, such as the energy minimization methods presented by Bader (2001) are of high cartographic quality but extremely time-consuming and thus not feasible in this context. Possible solutions are to optimize the selection/elimination operation to avoid right from the start possible overlap conflict potential. This table does not only contain traditional operators known in cartographic generalization (Bader *et al.* 1999, McMaster and Shea 1992) but also an approach which is used to transform the geometry of objects: **morphing**[9]. The morphing procedure computes an intermediate state from several parameters, including a starting and an ending state. A multi-scale database where objects are stored in different states (LODs) seems well suited for using such a morphing process. Potentials and benefits of using this approach are presented later in section 5.4.2.

A more detailed discussion of the generalization process can be found in chapter 5. The next chapter, however, will focus on the other main building block of our approach, the multi-scale database.

3.6 Summary

This chapter has presented a full comprehensive framework for on-demand mapping with a focus on on-demand *web* mapping. Starting from a definition of different scenarios of relevance in Internet mapping an architecture of such a system has been proposed. To bypass the missing operators and algorithms used for the on-the-fly generalization a special multi-scale database is built-up. This database consists of a minimum of two LODs, each representing the same area in different map scales. The map making process is controlled by various units and modules whose task is to create the requested map out of the data sets contained in the database. For the decision of how many and which LODs are needed a solution was proposed which is based on the study of the behavior of the generalization process.

The basic idea of this framework is the extension of the generalization process by an enriched multi-scale database for on-demand web mapping. These two components of the work will be presented in the following chapters.

---

[9] The term *morph* is derived from the Greek word *morphe* meaning "shape" and has in recent years acquired the meaning of **shape-changing**.

Chapter 4

Design of a Multi-Scale Database

As discussed in the previous sections, the multi-scale database plays an important role for on-demand web mapping. On the one hand, it replaces missing generalization algorithms for the automated map creation process and on the other, it speeds-up the whole procedure through the use of pre-computed information stored for several feature classes. Another task that assists web mapping is the construction of a *hierarchically* structured MSDB, where corresponding objects or group of objects represented in various LODs are linked. To take advantage of this a well structured data model is necessary.

This chapter presents a solution to build a data model that achieves the requirements of on-demand web mapping. In the first section (4.1) the data sets used to define a multi-scale database are introduced. The various possibilities of data enrichment, which help to make the database more valuable are illustrated by means of generalization operators in section 4.2. Linking[1] of corresponding objects or groups of objects represented in different LODs is described in the section 4.3. Hierarchical linking facilitates the generalization process because changes can be partially accomplished locally. The data model defined for the conceptual framework and used to build-up a MSDB is presented in the following section (4.4). A short discussion about the design of the MSDB will end this chapter.

4.1 Data Source

The foundation of the system is a multi-scale database with predefined LODs, the ideal number of which is determined by the different characteristics presented in the section 3.3.2. In practice, this is mainly determined by the data sets available. Many National Mapping Agencies (NMAs) at present only own a few data sets[2] deriving all other products out of them. For this work the landscape data models of the Federal Office of Topography (Switzerland) have been chosen. They offer two products in vector format: *VECTOR25* represents the National Map 1:25'000 and *VECTOR200* represents the map scale of 1:200'000. Each one consists of different thematic layers: each layer includes geo-referenced objects (points, lines or surfaces/areas), attributes and topology (relationship to neighboring objects)[3].

---

[1] As *linking* includes the matching process, the terms *matching* and *linking* are used synonymously in this thesis.

[2] The French Mapping Agency IGN distributes with BD TOPO® (scale 1:10'000 -1:30'000) and BD CARTO® (scale 1:75'000 - 1:150'000) two different data sets.

[3] A more detailed description of VECTOR25 and VECTOR200 can be found on the web site of the Federal Office of Topography: http://www.swisstopo.ch .

For a first prototype of the framework the number of layers has been reduced and regrouped into three *feature classes*[4] included in both data models:

- **Building**: including all types of buildings;
- **Road network**: with all drivable roads;
- **River network**: including all kinds of rivers.

A description of the terms used in this thesis can be found in the appendix A.1.

4.1.1 Digital Landscape Model VECTOR25

VECTOR25 is the digital landscape model of Switzerland whose content and geometry is based on the National Map 1:25'000. The product consists of eight thematic layers and includes a total of about 140 different kinds of objects. These eight layers include: road network, railway network, other transportation, hydrographic network, individual objects, hedges and trees, primary surfaces and functional surfaces.

Selected Feature Classes and Feature Elements

The different layers encompass several kinds of objects (e.g. for the road network more than 20 are available). Out of all these elements the most important ones have been selected and re-classified to restrict their number. The newly created layers are stored as *feature elements* and summarized in LOD_{25}. Table 4.1 illustrates all the data shared in the three main feature classes defined previously.

Feature Class	Feature Element	Selected kinds of objects (VECTOR25)
Road network	Highways	Autobahn
	Highway access	Ein_Ausf
	Main trunk road	Autostr
	1st - 6th class road	1_ - 6_Klass
	Neighborhood road	Q_Klass
Building	Building	Gebäude
River network	Stream	Bach
	Underground course of stream	Bach_U
	River	Fluss
	Underground course of river	Fluss_U
	Lake	See

Table 4.1: The selected kinds of object of VECTOR25 renamed and classified in the feature classes *road network*, *building* and *river network*.

---

[4] In this thesis the term *layer* (defined by the Swiss Federal Office of Topography) is replaced by the term *feature class*. The term *kind of object* replaced by *feature element*. A detailed description can be found in Appendix A.1.

4.1.2 Digital Landscape Model VECTOR200

VECTOR200 is a small-scale landscape model based on the 1:200'000 National Map in content and geometry. More than 400'000 feature objects are described with position, attributes and topology. Six thematic layers are defined comprising around 80 different types. These six layers are: transportation network, hydrographic network, primary surfaces, boundaries, buildings, and individual objects.

Selected Feature Classes and Feature Elements

Similarly to VECTOR25, the layers of VECTOR200 contain several kinds of objects called data layers. Table 4.2 lists the data layers selected from VECTOR200 renamed as feature classes. According to the selected feature classes in LOD_{25} the corresponding classes have been selected in LOD_{200}.

Feature Class	Feature Element	Selected kinds of objects (VECTOR200)
Road network	Highways	Autobahn
	Main trunk road	Autostr
	Main road 4m and 6m	DurchgStr4 and 6
	Main road as connection road 4m and 6m	VerbindStr4 and 6
	Secondary road 3m and 4m	Nebenstr3 and 4
	Road open to traffic	Fahrstraes
Building	Building	Z_Gebaeude
River network	River	Fluss
	Underground course of river	Fluss_U
	Lake shore	See

Table 4.2: The selected kinds of object of the data model VECTOR200 renamed and classified into feature classes *road network*, *building* and *river network*.

As it can be seen, both tables (4.1) and (4.2) illustrate that the classification of the feature elements in the feature class *road network* for VECTOR25 and VECTOR200, respectively do not agree with each other. Hence, the data models do not correspond, which implies that the same object does not belong to the same feature element in both data models. For example the feature element '1^{st} Class Road' of LOD_{25} has no corresponding feature element in LOD_{200}. This complicates the construction of the multi-scale database, where a linking between different representations of objects is of prime importance. In order to create these connections an integration and matching process must be carried out. This process will be described in more detail in section 4.3.

4.2 Data Enrichment

The process of **data enrichment**, sometimes also called 'data enhancement'[5], can be defined as follows:

> *Adding different kinds of information to an existing base data set to make it more valuable for a properly defined application.*

Integrating specific auxiliary information in an existing data set leads to a simplified processing of the data at a later point. This advantage is achieved by pre-computing values which need a lot of computational power to calculate and that do not detrimentally influence the data for the subsequent processes.

In most cases the added content is computed by analysis of the source data set for a specific use and prior to the mapping process. Such data analysis is often executed within a geographic information system (GIS) since these offer a set of functionality for exploring spatial data sets. This enrichment can assume different forms and is an integral part of the preprocessing stage which takes place off-line (Ruas 1999).

Data enrichment allows the storage of information or values from a preliminary generalization process which is scale-independent and does not constrain the later real-time processing. Not only pre-calculated values but also the addition of methods, which perform simple analysis concerning each object, augment the fitness of an enriched database (Abraham (1989), Jones *et al.* (1996)).

As pattern recognition becomes more and more important in the research field of cartographic generalization (Mackaness and Edwards 2002) it cannot be disregarded. Data enrichment can be used to treat pattern or structure of objects instead of dealing with individual objects (e.g., alignment of building objects).

4.2.1 Reasons for Data Enrichment

For on-demand web mapping, the additional information can help solve several problems concerning on-thy-fly generalization. The better the information included is tuned to the requirements defined in advance, the more the required generalization process will be enhanced. Three reasons can be distinguished as to why the enrichment of a database could be so helpful:

Accelerate Inclusion of selected information that has been computed in advance and does not have to be processed at run-time saves computational time and allows thus a faster response time.

Substitute As no corresponding algorithms exist for certain generalization operators or where they do exist but the computational time is too long, possibilities to replace them must be sought. A facility to replace such algorithms is to add relevant pre-calculated information to the database.

Describe Another kind of data enrichment is added information about the semantic and geometric description of objects (Jones *et al.* 1996) or groups of objects (pattern, structures). Examples can include **values** of 'importance' or values about the 'sequence of process' or even **methods**.

The third point 'Describe' includes all possible *sets of measures* used to qualify the characteristics of the geometry of an object. Measures can be used to detect and to evaluate particular characteristics of a feature in order to optimize the selection of an algorithm. Further details on measures in the context of cartographic generalization can be found in AGENT

---

[5] For this work the terms *data enrichment* and *data enhancement* are used synonymously.

Consortium (1999) and Peter (2001). These three types of added information can be illustrated with examples concerning cartographic generalization operators. The following section 4.2.2 describes a variety of possibilities for data enrichment.

4.2.2 Possibilities of Data Enrichment

The possibilities of data enrichment presented in this section are organized on the basis of the generalization operators listed in Figure 2.5. Options are shown to enrich the base data set in conjunction with on-demand web mapping.

Operator *Selection*

In the cartographic generalization process the operator *selection* is applied first of all. The choice of whether an object will be represented or not depends on the target scale, purpose of the map and the user requirements. For *selection* two possibilities for data enrichment are presented: calculation of the area value and a Horton/Strahler ordering of river networks.

- **Area**: *Compute area value for polygons*
  This added value is used in conjunction with polygon objects. An example is the feature element *lake*, that is not stored as a polygon feature but as a polyline representing the lake shore. To simplify the selection criteria a pre-calculated area value allows a faster selection/elimination of the object to be displayed. This value can also be used for other operators or build the basis of other, more complex geometric measures such as shape measures.

- **Horton/Strahler order**: *Ordering a hydrologic network*
  The Horton and Strahler order are stream orders developed by the physical geographers and hydrologists Horton (1945) and Strahler (1952). River objects in a network are identified and ranked in order of importance; for a reduced level of detail only a certain set of elements – the more important ones – will be used. By thresholding the data using one of these orders, the network can be simplified to the desired degree. Figure 4.1 illustrates the difference between these two kinds of ranking of a hydrologic network.

Figure 4.1: The Horton (left) and Strahler ordering (right) of hydrologic networks. Data: VECTOR200 © swisstopo (BA034957).

For the Horton order, the main (i.e. the longest) stream is selected and assigned the highest order throughout its length. From the remaining tributaries, the longest stream is selected in each and assigned the maximum order which occurs on its length. To define the main stream the criteria straightness and length are used (Richardson 1993). The Strahler ordering method starts at 1 at each terminal segment, and proceeds towards the network root. Each time a furcation node is encountered, if both daughter branches have the same order, then the order is increased by one, otherwise the largest order is used. The order is the function defined on the edges of the tree given by the following rules:

(i) edges connected to leaves have order 1,

(ii) if two edges of orders i and j concur at a node, the edge downstream has order $i \vee j$ if $i \neq j$; or $j + 1$ if $i = j$.

For a hydrologic network, it has been shown by Rusak Mazur and Castner (1990) that the Horton stream ordering method most closely approximates the generalization decisions made by a human cartographer.

Operator *Typification*

The typification operator can be defined as follows: a group of objects is represented by a new reduced set of objects showing similar characteristics concerning density and orientation.

- **Orientation**: *Orientation of buildings*
  In combination with the feature class *building* a pre-calculated orientation allows the detection of structures (e.g. building objects aligned along roads). In a further step buildings located near to each other and displaying the same orientation can be collected into a common pattern (*pattern recognition*). With the help of these groups the generalization process can be applied to them and thus accelerate the full map generation process. Hence the characteristics of the pattern can be preserved while reducing the number of objects. As this orientation value is calculated in advance no time restrictions must be regarded. Figure 4.2 illustrates a dissemination of buildings along a road path.

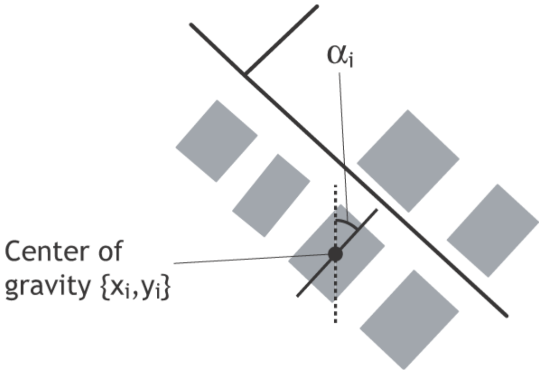


Figure 4.2: Defining patterns of buildings respecting the orientation of buildings.

The value 'orientation' is defined as angle α_i between the longest axis of a building and the x- or y-coordinate (see Figure 4.2). The location of a building is determined as the center of gravity $\{x_i, y_i\}$ of the object. These two values do not suffice to describe an alignment of buildings, but can avail for clustering objects forming a special arrangement (see also next bullet). More research on building typification and detection of building alignment is described in Christophe and Ruas (2002).

- **Grouping**: *Creating groups of buildings*
  As seen in the previous bullet grouping of buildings representing a special pattern helps to simplify the generalization process accomplished in real-time. Different kinds of building classifications are conceivable. Beside the aggregation of aligned buildings along roads all possible characteristics of clustering can be used. For example, a partitioning of the map by the road network leads to spatial arrangement. Another option is to calculate a ratio density D defined as the summarized area of the building objects A_{Bi} inside a specified area A:

  $$\mathbf{D} = \frac{\sum_{i=1}^{n} A_{Bi}}{A} \begin{cases} D \to 1 & \textit{dense} \\ D \to 0 & \textit{sparse} \end{cases}$$

  As the case may be, very dense or very sparse areas can be separated merging all the appropriate buildings to one group obtaining the same identification.

Operator *Simplification*

Simplification in cartographic generalization eliminates unimportant details either preserving a subset of the original coordinates (weeding) or creating new points to simplify the object (unrestricted simplification). For the feature class *building* two possibilities of enrichment – depending on the target scale of the map – can be distinguished. If the target scale is just slightly smaller a *description value* characterizes the outline of a building. If the transformation is bigger, a pre-calculated mathematically correct 'Minimum Bounding Rectangle' (MBR) described by four components [*width*, *length*, *center of gravity* and *orientation*] specifies the object. For the simplification of line objects (e.g. rivers) the enrichment of the data includes parameters used by the algorithm employed.

- **Description**: *Description of the shape of a building*
  For the feature class *building* the outline of objects can be described and compared with templates defined beforehand. These templates represent simplified patterns of building objects. Rainsford and Mackaness (2002) describe such pattern matching algorithms to select an outline out of a set of templates that best characterizes the more detailed form. In order to generalize the outline of an object only the corresponding template must be taken. Hence, an identification of the object outline must be accomplished first (*shape recognition*). In a second step a matching between building objects and building templates takes place. Both steps are carried out in advance off-line and are not limited by any constraints. Therefore the simplification step which must be accomplished at run-time can be simplified by replacing the building object by the corresponding template. Attention should be paid to the definition of these building templates. Figure 4.3 illustrates five different simplified buildings outlines T_j (cf. lower row). Each template T_j features an ascertained value which enriches the original building data set. The number and the detailing of the template shapes is not determined but it should be adjusted to the scale range and the purpose of the created map.

- **MBR**: *Minimum bounding rectangle*
  A minimum bounding rectangle (MBR) is defined as the smallest rectangle completely enclosing a number of objects. Figure 4.4 illustrates an example for a building object. The edges of this MBR – also described as the mathematically correct MBR – are not

Figure 4.3: Describing outlines of building objects by using alphabetic characters or templates defined beforehand.

parallel to the x- and y-coordinate but best fitted on the selected object, in this case a building. As the figure shows the MBR represents a simplified design of the original object. The outlines of building objects must be adjusted to the target scale of the requested map. In the map series of the Federal Office of Topography starting from the scale 1:50'000 the contours of the buildings are less detailed (cf. Figure 3.5) and at a scale of 1:100'000 most buildings are represented by a rectangle. To carry out this operation of simplification at run-time needs a lot of computation and thus time resources. Thus it would be ideal to calculate these 'rectangles' in advance and to include the information in the multi-scale database. Enriching the data set with a MBR of each building object beforehand allows the map creation process to be simplified especially for on-demand web mapping. The MBR can be pre-computed by a GIS for each building object. During the on-the-fly mapping process the pre-calculated 'rectangle' need only be adjusted to the requested map scale.

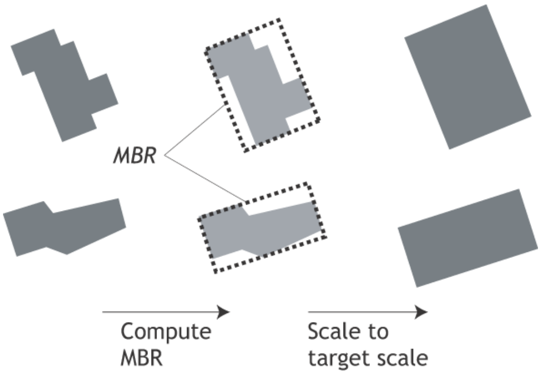


Figure 4.4: Computation of a minimum bounding rectangle for data enrichment.

- **Tolerances**: *Pre-calculated scale-dependent tolerances*
  Often the granularity of a polyline is not adapted for an application, that is, the vertices on the polylines representing the object boundaries are too fine-grained for the requested scale. For the sake of efficiency it does not make sense to display a polyline by all these vertices. There are several algorithms to reduce the number of vertices of a polyline

while retaining those that are essential (e.g. the Douglas-Peucker algorithm (Douglas and Peucker 1973) or the algorithm Visvalingam and Whyatt (1993)). To reduce vertices to produce a simplified polyline that approximates the original line within a specified *tolerance* one of these algorithms can be applied. Both algorithms can be pre-computed and for each vertex, the associated tolerance at which the vertex is eliminated can be stored in the database. While the algorithm by Visvalingam and Whyatt (1993) is directly based on pre-computation, methods for pre-computation using the Douglas-Peucker algorithm for the on-the-fly generalization have been presented in the literature (Oosterom van and Schenkelaars 1995).

Operator *Collapse*

The generalization operator *collapse* is essential for the manipulation of areal and linear features and describes a reduction of the dimensionality: from area features to line (e.g. a small lake to a river feature) or point features (e.g. a highway access) and from line to point features (e.g. a railroad yard). It is very difficult to decide how and how much to collapse an object to obtain the best representation of it (Bobzien and Morgenstern 2002, Schürer 2001). Complex operations always imply a lengthy computation time that is not desiderable for on-demand web mapping. The accomplishment of this operation in advance may accelerate the generalization and thus the mapping process.

- **Exchange shape**: *Definition of medial axis for lakes*
  As previously discussed, a pre-calculation for the collapse operator may be very useful in regard to the transformation from an areal to a line feature. A good example is the feature element *lake*. Figure 4.5 illustrates such an example where a lake object is collapsed to a line object that replaces it. This coincidence can be computed in advance by a skeleton method (Chithambaram *et al.* (1991), Bader and Weibel (1997)) enriching the original data set. Depending on the requested map scale one of these representations must be displayed. For example, if the target scale is so small that the extent of the lake object is smaller than the minimum perceivable area, the line object is shown. Conversely, if the extent complies with the minimal area, the area depiction is selected. Figure 4.5 shows the two possibilities. On the left hand side the areal feature is shown and on the right hand side the skeletonized lake feature, termed the *lake axis*.

4.3 Data Matching Process

The main problem when integrating two data sets is the different semantic and geometric representations of spatial objects. Even multiple acquisitions of data at the same scale may lead to different data sets because of different discretization of coordinates and interpretation of the landscape. Beside this, additional problems arise since a lot of data sets focus on other aspects of the world and/or may have data quality characteristics. In order to use all the available information the various data sets must be linked explicitly (Sester *et al.* 1998). The linking of different representations can be equated with the matching process and defined as follows:

> *The data matching process consists of the computation of correspondence relations between sets of geographical entities which represent the same phenomenon of the real world as two different representations.*

The process of data matching describes two main aspects to make the available data sets more useful: on the one hand it characterizes the correspondence between individual objects

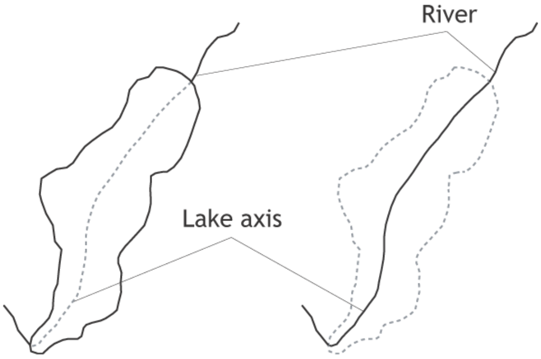


Figure 4.5: Representing a lake by its medial axis in a smaller map scale.

in different representations (e.g. different map scales). This not only involves the linking of single objects but also of groups of objects and can be used to define one common data set including several representations. By means of this, data sets with different provenance related by scale, thematic focus, and information content can be combined into one set in a *multi-scale database*. On the other hand, data matching can include information used for the derivation of new representations out of the existing ones. This generation of new data sets is carried out by a generalization process F which defines the transition from one level of detail to another (*database generalization*):

$$LOD_{A'} = F(LOD_A, Scale).$$

Linking can take place in various degrees and adopt different forms. On the one hand, on the individual level where the representations of every object are linked, and on the other hand on the 'group' level, where objects are summarized to a group (e.g. objects forming a cluster). Beside matching at the object level also an abstract matching on the feature element or the feature class level is conceivable.

Why is the matching process important for on-demand web mapping? The added information from the linking process allows the generalization procedure to be accelerated in two different ways. First, the two representations given in the data model demonstrate two states of an object: a start and an end state. These two states are necessary for applying the morphing approach outlined in section 3.5 and explained in more detail in 5.4.2. Second, knowing if there is a corresponding representation in the other LOD or not can simplify the selection/elimination part of the generalization process.

The difficulty of the matching process lies in detecting the same object or group of objects in all representations: a representation can alter its shape as well as topological dimension across scales. This can include a conversion from polygon to polyline (display of a small lake in a small-scale map) or from polylines to a point (e.g. roundabout). At the semantic (or schema) level three cases (types) of matching must be distinguished (compare the example '1^{st} Class Road' below):

Case 1 The representations of an object are classified into the same feature class and have the same description in the feature element level in both LODs;

Case 2 The representations of an object in the two LODs are in the same feature class but

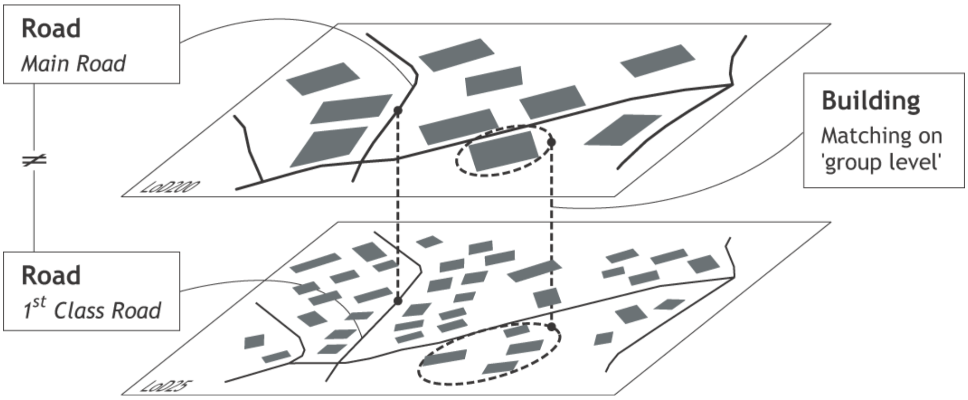


Figure 4.6: The same object in reality (in this case a road) do not necessarily have to be represented by the same feature element in LOD_{25} and LOD_{200}. For the feature class *building* a matching process on the 'group level' takes place. Data: VECTOR25/200 © swisstopo (BA034957).

named differently (see Figure 4.6);

Case 3 There is no representation in another LOD and thus no corresponding feature element.

	Data Model	Feature Class	Feature Element
Case 1	1:200k	Road Network	1st Class Road
	1:25k	Road Network	1st Class Road
Case 2	1:200k	Road Network	Main Road
	1:25k	Road Network	1st Class Road
Case 3	1:200k	Road Network	-
	1:25k	Road Network	1st Class Road

Sester *et al.* (1998) presented three approaches for the linking process of objects in spatial data sets. The first defines linking as a geometric matching problem, the second and third focus more on the semantic derivation of a representation. The model generalization is based on sematic links between the different data sets used (ATKIS and GDF[6]). Walter and Fritsch (1999) extended this work and based their approach on statistical investigations of manually matched data sets. The result of applying this approach to a road network shows that a high percentage of correct matches can be found automatically even in difficult areas. The work of Gabay and Doytsher (1994) considers more how to match a map which is slightly different regarding geometric properties. Saalfeld (1985) and later Filin and Doytsher (2000) made a further step in detecting node objects rather than linear features as candidate counterparts for the matching process. Filin and Doytsher propagated a related approach for the detection of corresponding linear objects in different data sets of nearly the same scale. Their approach is based on using geometric and topological criteria to detect counterpart nodes. They applied their algorithm to a number of data sets with low and medium data intensity and the result

---

[6] ATKIS is the German topographic cartographic spatial database, GDF is a European standard for the modelling, acquisition and exchange of road network data primarily used in car navigation.

was a correlation of nearly 100 percent. In dense areas more than 90 percent of correct matches were achieved which provides evidence of the robustness of the approach. Another solution for matching linear elements is proposed by Devogele (2002). The focus of his work was on identifying homologous points for the geometry of homologous objects. This process matches common points automatically, based on measures derived from the Frèchet distance. The drawback of the work is that these homologous objects must be found beforehand.

4.3.1 Relations and Cardinality

The matching process is based on the study of the relations of objects represented in different LODs. So, it is important to understand all possible associations or types of associations between objects, feature elements or feature classes. Mapping these connections can be done with the help of *cardinality constraints* which define the number of possible relationships. The cardinalities declare which relation an object has to another object or group of objects and characterize the level of complexity of such a relation. To distinguish are

$$[1:1], \quad [1:n] \quad and \quad [n:m] \qquad \forall n, m \neq 1$$

The matching process for merging spatial data sets in a multi-scale database is complex. The definition of relations and cardinalities between individual objects or group of objects is strongly dependent on the selected feature classes and feature elements. For the simplest situation – a one-to-one correspondence $[1:1]$ – only a few cases are conceivable. In almost all other cases more complex relations are usual. Figure 4.7 illustrates the possibilities for the selected feature classes *road network*, *building* and *river network*.

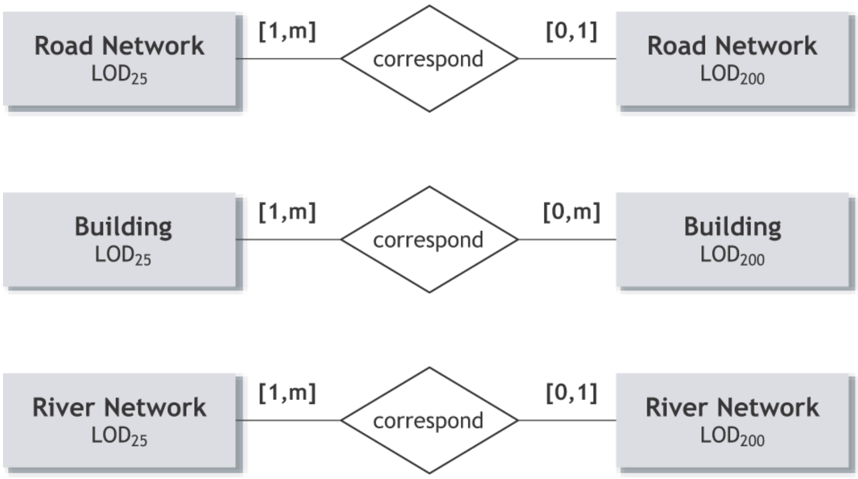


Figure 4.7: Possible relations for the selected feature classes and the corresponding cardinality ratios.

For the feature class *road network* the cardinality constraints are represented by $[0,1]$ and $[1,m]$. $[0,1]$ means that each element of LOD_{25} corresponds to no or one element of LOD_{200} (zero-to-one relation). On the other hand $[1,m]$ describes that each road element of LOD_{200} has a minimum of one counterpart in the other data set (one-to-many relation). The same case can be ascertained for the feature class *river network*. Buildings act differently to the other two classes. Their constraints are $[0, m]$ and $[1, m]$. The feature class *building* plays a special role in maps of the Federal Office of Topography. In most maps of other countries individual buildings are displayed as a built-up area (using areal tints) starting from a scale of 1:50'000. On Swiss maps this happens only around 1:200'000 so that in this research work the buildings are always represented as individual objects (both at LOD_{25} and LOD_{200}).

Cases of a direct linkage between individual objects ($[1:1]$) are very rare when matching spatial data sets with different scales. In most cases the objects in the smaller LOD act as a collection of similar elements in the LOD with a larger scale. Beside this difficulty, two further problems complicate the matching process. On the one hand the differences of the data models (schemata) of VECTOR25 and VECTOR200 (especially for the feature class *road network*) and on the other hand the cardinality constraints for the feature class *building* which is mapped by $[n:m]$.

4.3.2 Approaches to Matching

For the realization of the matching process four main approaches must be distinguished:

- Matching based on **semantic** criteria;
- Matching based on **geometric** criteria;
- Matching based on **topological** criteria;
- **Combination** of two or all of the above.

Depending on the feature class to be matched, one of the first three approaches must be chosen. For the best result and the most accurate matching a combination of them must be selected.

Semantic matching Semantic matching is, in most cases, the easiest way to connect corresponding objects. This can be done by the object name (e.g. road name) or the object identifier OID if they are equal in both representations. In general this information is not or only partly available. For some feature classes such an approach is not accomplishable because a one-to-one relation $[1:1]$ is not given in any case (e.g. building objects).

Geometric matching Geometric matching is based on the detection of corresponding objects in two representations by means of purely geometric characteristics. The realization of such a geometric combination can be obtained by comparing the geometric location of an object or a group of objects. Other measures like area, length and values for the description of surfaces (e.g. shape index) can be included to find corresponding elements. This approach has been used for matching groups of building objects (see section 4.3.4).

Topological matching Topological matching uses topological relationships between different objects. Especially for networks like the road or the river network topological matching can enormously help to find the best counterpart.

Combination of the approaches Depending on the available information in the data set a combination of the above approaches can lead to the most accurate result for selected feature classes. Particularly for the river network, where some larger rivers are enhanced by their name and smaller ones are not, the application of all approaches makes sense.

For solving problems of matching various methods are conceivable. The main idea in this work concerning the matching process is not to create a new landscape model for both data sets (VECTOR25 and VECTOR200) but to merge them into one common schema. This model therefore includes all created links between corresponding objects in LOD_{25} and LOD_{200} and is defined as a multi-scale database. The result of the matching process is an identification number called **group_ID**, which is included in all objects of both LODs as attribute value. Corresponding objects or groups of objects are characterized by the same value. This identification value group_ID is not to be confused with the OID (Object Identification Number) which is unique for every object and has nothing to do with matching or linking.

In the following sections the problem of linking is discussed separately for the three selected feature classes *road network*, *building* and *river network*. The focus is placed on deriving the group_ID value in the matching process. Based on the data sets VECTOR25 and VECTOR200 an integration of the two landscape models into one will be reached.

4.3.3 Matching Process for the Feature Class *Road Network*

For the matching process of the road network, the road centerlines captured in the different landscape models were considered. Because of differences in the classification of the feature elements of the two data sets LOD_{25} and LOD_{200} the linking process is more complex and more expensive than for other feature classes. Tables 4.1 and 4.2 illustrate the differences within the feature class. This shows that the matching process cannot be based solely on the semantic approach but a combination with geometric matching must be considered.

To simplify the sematic step (schema integration) a prior study of the possibilities of correspondence between the feature elements in both data sets must be accomplished. Based on the selected test area an evaluation of this identification was carried out and resulted in the following list:

Autobahn = {Autobahn}
Autostr = {Autostr}
DurchgStr6 = {1st class road}
VerbindStr6 = {1st class road, (2nd class road)}
VerbindStr4 = {1st class road}
Nebenstr3 = {2nd class road, (1st class road), (Q_Klass)}
Fahrstraes = {3rd class road, 4th class road, Q_Klass}

The left hand side represents the feature element in LOD_{200}, the right hand side describes the sets of possible corresponding feature elements in LOD_{25}. For instance, all representations of objects in feature element VerbindStr6 (LOD_{200}) can pertain to the feature elements 1st class road or less probably to 2nd class road in LOD_{25} (values in parentheses are less probable).

Process Model

Since it is not always possible to match the corresponding feature elements on the basis of the semantic approach, a combination will be described. The main idea is to match the road segments indirectly by matching the nodes of the road network. Thereby the matching process can be divided into the following two steps:

(i) In the first stage the *nodes* of the road network of both LODs are matched;

(ii) In a second step the matched nodes are connected with the corresponding road segments.

Figure 4.8 gives more details of the workflow of the road matching process. The starting point of the process is the selection of nodes of a feature element (e.g DurchgStr6 of LOD_{200}). As explained these nodes are matched with the ones of the corresponding feature elements in LOD_{25}. In this case the objects of 1st class road are concerned.

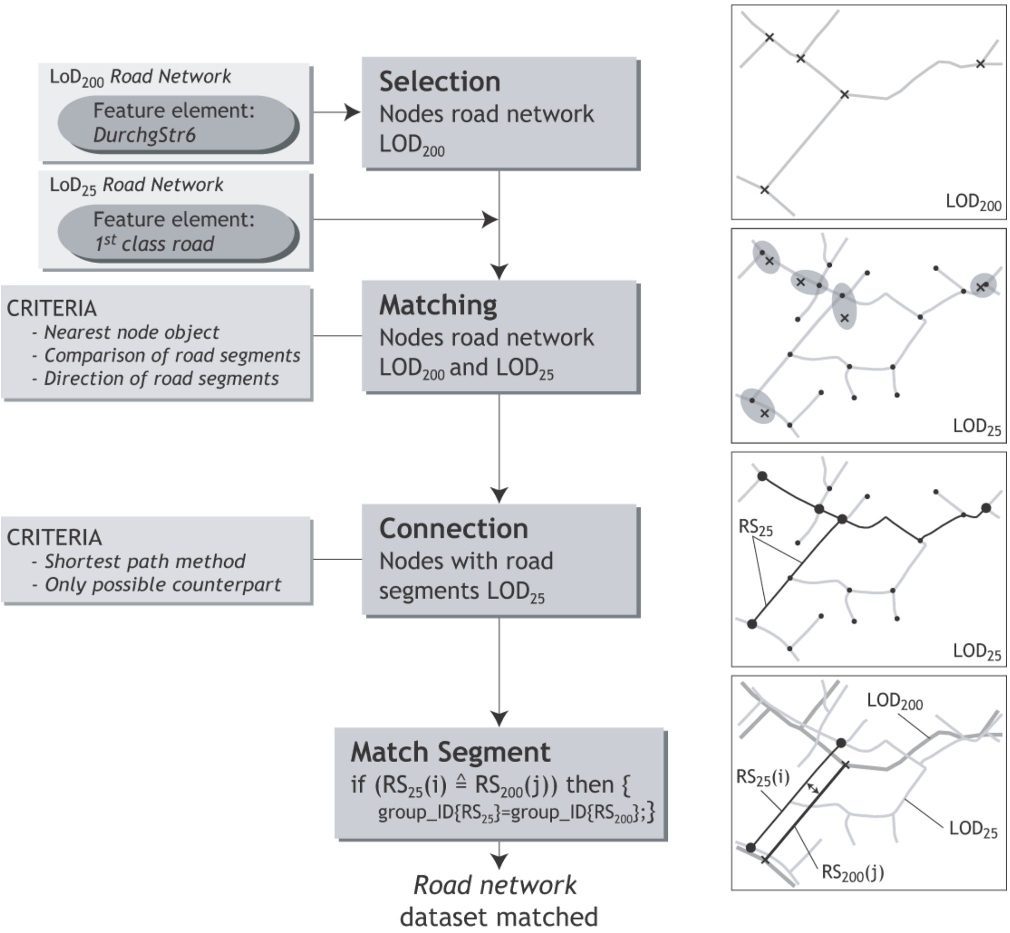


Figure 4.8: The workflow of matching the road network of two LODs, discussed on the example of DurchgStr6 (LOD$_{25}$) and 1st class road (LOD$_{200}$).

The first step in locating possible counterpart nodes is the *selection* of all nodes of DurchgStr6 (LOD$_{200}$) and the corresponding feature element in LOD$_{25}$, which is 1st class road. The search of the corresponding node is based on three criteria: i) nearest node object; ii) comparison of adjacent road segments; and iii) direction of adjacent road segments. Through the elimination of impossible candidates (classification of the adjacent road segments does not correspond) the geometric matching can be helped considerably. The road segments RS between the matched nodes can then be linked. According to the possible feature elements two reasonable criteria must be considered: i) the specification of the shortest path (see Skiena (1997)); and ii) the affiliation to a possible counterpart (concerning the feature element). The matching process ends with distributing the same value for the group_ID to the corresponding road segments RS.

Discussion of the Process

In considering this approach of road network matching some restrictions and assumptions have been made. Concerning the nodes only [1 : 1] relations are considered, which implies that each node has at most one counterpart. In reality each junction can be represented by a different number of nodes depending on the scale (Figure 4.9, left). At larger scales in most cases a [1 : 1] relation is usual. At smaller scales this is not always the case since, for reasons of lack of space, junctions which are right next to each other are collapsed into one. Another difficulty for the matching process is given by roundabouts (Figure 4.9 (right)), where at smaller scales they are represented by one node and at larger ones by the number of incoming roads ([1 : n]-relation).

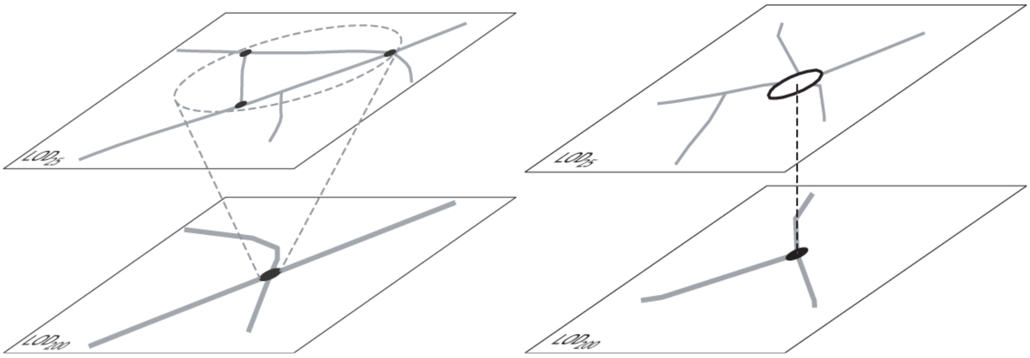


Figure 4.9: The three junctions in LOD_{25} are represented by one in LOD_{200} (left). Roundabouts are represented in smaller scales (LOD_{200}) as node and in larger scale (LOD_{25}) as linear feature (right).

4.3.4 Matching Process for the Feature Class *Building*

Matching the feature class *building* poses various difficulties. One of the most important one is that a [1 : 1] correspondence between the objects in LOD_{25} and LOD_{200} is an exceptional case. In the majority of cases relations of type [n : 1] or [m : 0] dominate (whereby $n \neq 1$ and $m \geq 1$). Thus it does not make any sense to search for *a* counterpart candidate in the smaller-scale LOD. Hence, the idea is based on matching groups[7] of aggregated building objects. As no semantic information is included in the data sets of the Federal Office of Topography for the feature class *building* semantic matching is practically impossible. In this case only geometric matching of the objects can be accomplished. Two main steps must be distinguished:

- Arranging building objects to groups in LOD_{25};
- Searching counterparts of this group in LOD_{200}.

Aggregating buildings into groups or clusters is a widespread research topic in cartographic generalization (Regnauld 1996, Boffet and Rocca Serra 2001). The proposed approach is based on a four-step procedure illustrated by Figure 4.10.

---

[7] The more neutral term *group* illustrates better that collections of building objects are formed in this case than the term *cluster*.

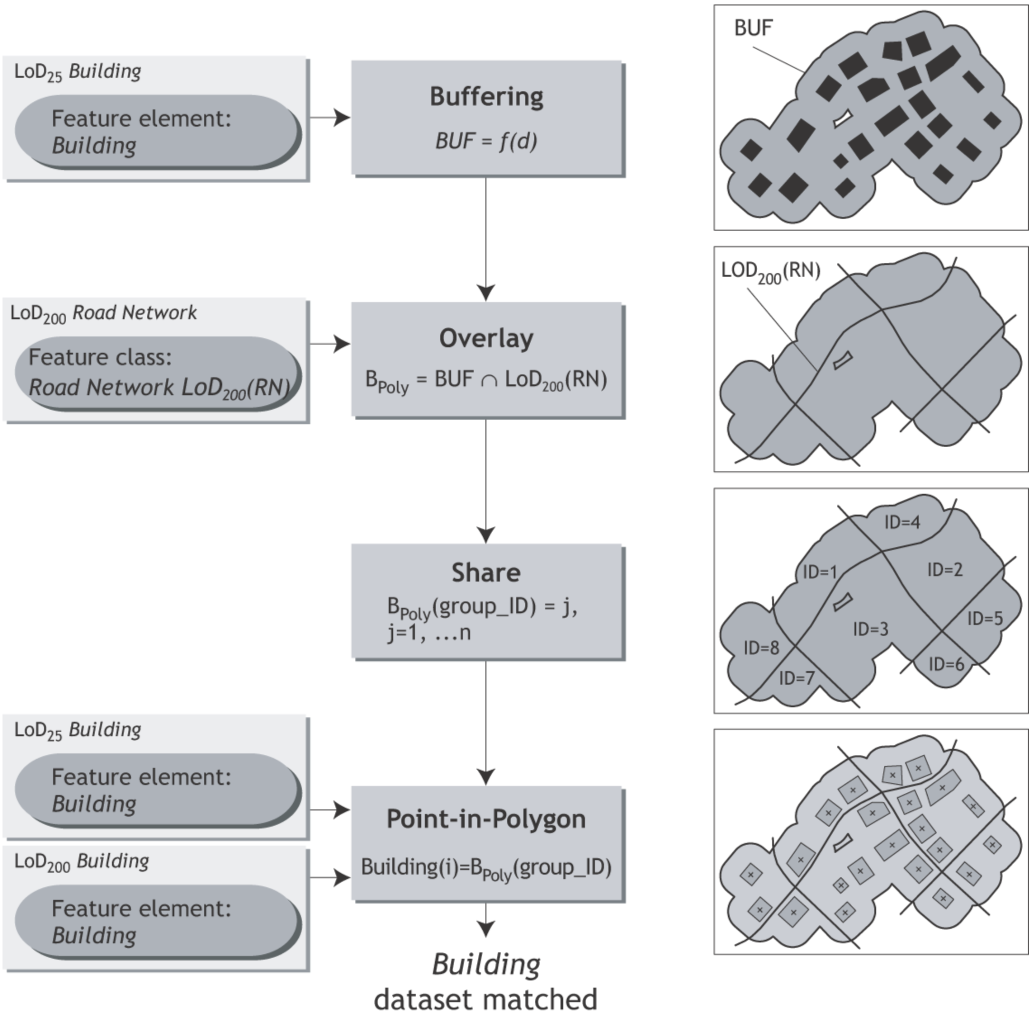


Figure 4.10: Workflow of the matching process for the feature class *building*.

Starting with the feature elements of LOD_{25} a buffering operation with distance d is carried out. Each original object is assigned a buffered area around it. Overlapping areas are merged into a common polygon (which is equivalent to the boolean operation OR). The generated result (BUF) represents several polygons of various expansion, depending on the number of objects included. To subdivide the 'big' polygons of these groups into smaller ones an overlay with the road network of LOD_{200} is carried out next. This method of *area partitioning* (Brazile 2001) allows a better handling of the building objects in connection with the subsequent generalization process. At this moment each created polygon (B_{Poly}) is assigned a consecutive number which represents the corresponding objects in both LODs (group_ID). For the last step of the matching process a point-in-polygon-test (Saalfeld 1987) including all objects of LOD_{25} and LOD_{200} must be accomplished. To enable the execution of the test (building objects are polygons) a point of the building must be taken. In this case the most suitable point representing the individual building is the 'center of gravity'. Using this test with the center of gravity (or centroid) each object can be assigned to one group. Since the buffer polygons B_{Poly} are based on the data set of LOD_{25} no problems appear with this layer. On testing objects of LOD_{200} in some cases unassigned points can be found. This means that the corresponding center of gravity does not pertain to one polygon. In such a circumstance the point will be assigned to the nearest polygon and obtained its group_ID. As soon as all points are allocated and thus all buildings belong to a group the matching process has finished.

Discussion of the Process

The advantage of not trying to match individual buildings but groups of buildings allows a simplified generalization process at a later time. By the use of area partitioning self-contained groups are defined which can act more or less autonomously from other groups or building objects. The combination with a buffering approach permits to subdivide the space more building-specifically than only with the road network. As the grouping process depends on the buffer distance d different solutions are possible. Depending on the value of d the number of building objects pertaining to one group can vary considerably (see the experiments in section 6.2.2).

4.3.5 Matching Process for the Feature Class *River Network*

Although rivers are linear elements[8] like the road network (cf. 4.3.3) a two step procedure must be defined regarding the data set of the Federal Office of Topography used:

(i) Matching on the basis of the name of the river (semantic criterion);

(ii) Perform the same procedure as for the road network (geometric criterion).

The data set used includes for the feature class *river network* an attribute with the name of the river in both LODs. In the case of LOD_{200} all objects are named (in contrast to LOD_{25}). But since the objects that appear in both LODs are named the searching of a counterpart can be alleviated by comparing the name of the rivers and matching them accordingly. If, however, rivers are *not* named a similar approach as for the road network can be performed.

Figure 4.11 illustrates the process for the river network in the case of the data set of the Federal Office of Topography. The matching process is carried out using the attribute name. Starting with the selection of an object in LOD_{200} the counterpart in LOD_{25} with correspondence of the attribute name is looked for. By detection of the same value both objects get the same value for the attribute group_ID.

Compared to the road network the river network can additionally be classified by a stream order, like the Horton-ordering (Horton 1945), as described in section 4.2.2. By means of this value the number of candidate corresponding objects can be reduced.

Discussion of the Process

As a semantic matching based on the name of the rivers can be accomplished the process itself becomes relatively simple. Lakes, which can interrupt rivers, must be paid special attention. Particularly smaller lakes which disappear in small scale maps must be replaced by a river segment. A practical solution is to not only store the outlines of a lake but also their medial axis. This lake axis will be displayed if the minimal dimensions of the polygonal object cannot be retained (cf. section 4.2.2).

---

[8] The objects of the hydrographic network (data set of the Federal Office of Topography) form a line network, whereby the lines are vectorized in the flow direction.

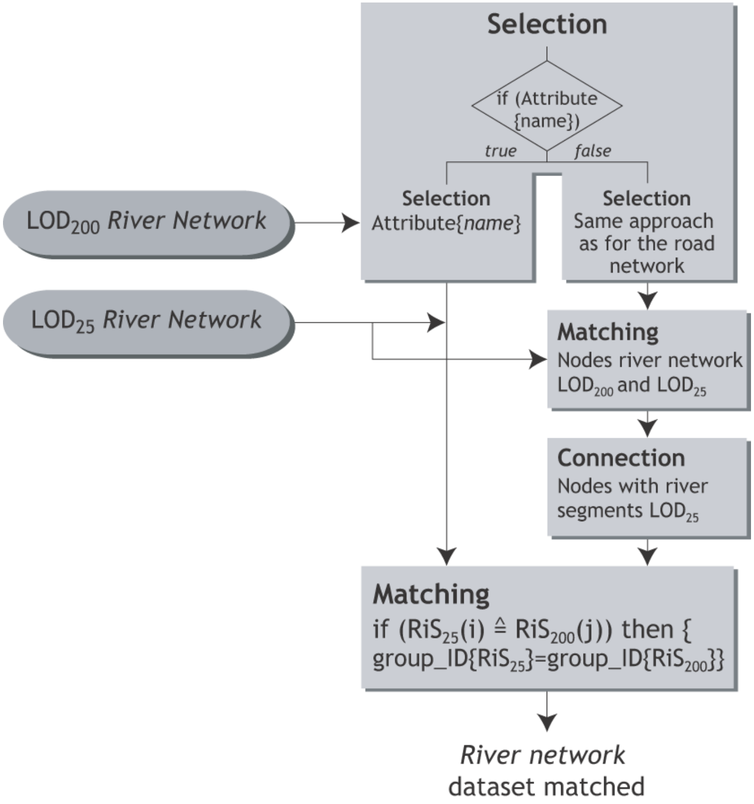


Figure 4.11: Workflow of the matching process for the feature class *river network*.

4.4 Data Model (Schema)

A **data model** (or schema) is a conceptual representation of the data structures that are required by a database. The data structures include the data objects, the associations between data objects, and the rules which govern operations on the objects. As the name implies, the data model focuses on what data are required and how they should be organized rather than what operations will be performed on the data. In summary a data model is

- a result of a conceptual design showing the generalized, user-defined view incorporating only the data relevant to an application;
- a formal method of describing the behavior of the real-world entities and defining their attributes and the relationships between these entities;
- independent of a computer system and its associated data structures.

The data model defined for this work is based on the system used for the implementation (which will be presented in section 6.1) LAMPS2 from Laser-Scan Ltd[9]. Before starting some

---

[9] Laser-Scan delivers commercial solutions for geographic information systems and is member of the YEOMAN Group plc (*www.laser-scan.com*).

general remarks about the model and the notation are needed. Each feature class consists of one main class and several subclasses inheriting from the main class. Subclasses only contain additional attribute values. The main class is built-up of a set of five information elements (C, G, O, A, M) whereby

- **C** describes the *name* of the feature class;
- **G** determines the simple *type of geometry* of the feature class;
- **O** defines the *object identifier*, which is needed for the instantiation of a class;
- **A** lists a set of *attributes* describing an instantiated object;
- **M** contains a set of specific *methods* included in the feature class.

In the following sections 4.4.1 - 4.4.3 the data model for the feature classes *road network*, *building* and *river network* are presented. Other possible feature classes (e.g. railroad, forest) will not be discussed in this work. The data model described here is based on the data set utilized and designed for the purpose of the subsequent generalization process for on-demand web mapping.

4.4.1 Feature Class *Road Network*

The feature class *road network* is composed of four different classes: the main class Road and three subclasses Road25k, Road200k and RoadRes which all inherit from the main class. Figure 4.12 illustrates the structure for this feature class.

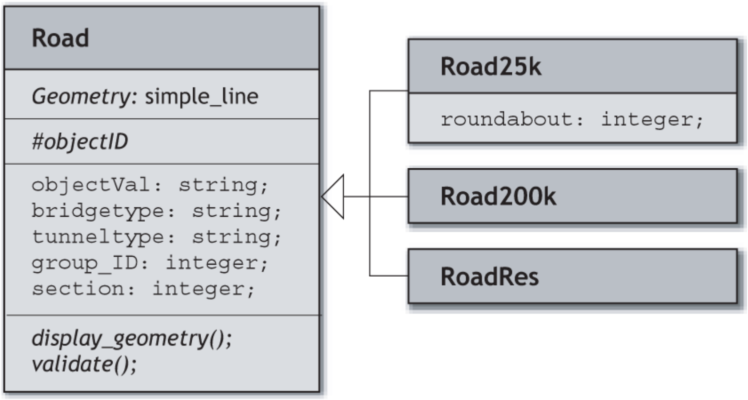


Figure 4.12: Structure of the feature class *road network*. The classes Road25k, Road200k and RoadRes inherit from the superclass Road.

As explained before the main class contains five sets of elements of information. The road network is described by linear objects (road centerlines) at all scales. The attribute list is not complete but represents the most important ones for the feature class. `ObjectVal` (atomic type: *string*) describes the classification of the road type (e.g. 1st class road). `Bridgetype` and `Tunneltype` (at: *string*) give more information on the construction type of the segment. Two values enriching the original data set are `group_ID` and `section`. The value of `group_ID` (at:*integer*) is the result of the matching process presented in section 4.3. For the subsequent generalization process the attribute of `section` (at: *integer*) is introduced. This value is a consecutive value for numbering the sections of a road between two junctions. The methods mentioned (*display_geometry()* and *validate()*) only represent a small selection of all existing ones. These methods will not be dwelt on because they are not directly relevant for the subsequent generalization process.

In addition to this main class three subclasses are defined: Road25k and Road200k represent the objects of the corresponding LOD. Additionally the subclass Road25k owns an attribute `roundabout` (at: *integer*) which is used for road segments describing a segment of a roundabout. This added value is important because it can be linked with the node which represents the roundabout on a smaller scale. The third subclass RoadRes contains the generalized result after a user request has been served.

4.4.2 Feature Class *Building*

Similar to the previous feature class, the *building* class consists of one main class Building and three subclasses Build25k, Build200k and BuildRes. Figure 4.13 illustrates the structure of this class.

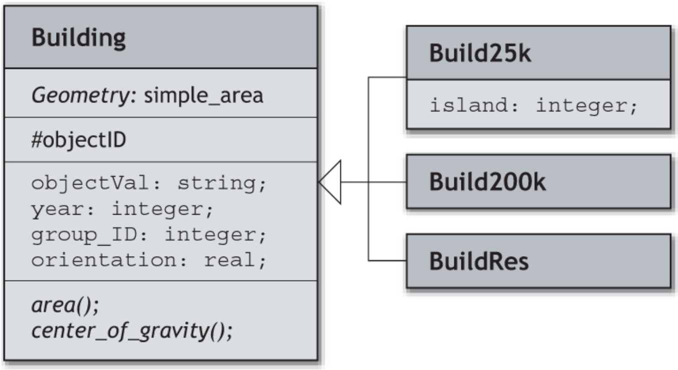


Figure 4.13: Structure of the feature class *building* with the main class Building and the subclasses Build25k, Build200k and BuildRes.

All five sets of information are also included in this main class. Given the feature object, the *name* of the class is **Building**. Attention must be paid to the selection of the *geometry type*. In principle, both linear (polyline) and areal types are possible for the representation of building objects. In this case the areal type has been chosen for reasons of the following generalization process. The values for the attributes `objectVal` (at: *string*) and `year` (at: *integer*) are not so important because no distinction is made for both of them. In the original datasets[10] all objects hold the same value 'Z_Gebaeude' for the attribute `objectVal` and the value '1993' for the attribute `year`. Changing the value of the attribute `objectVal` to indicate important buildings a special identifier could make the selection part of the generalization process even simpler (i.e. a further possibility of data enrichment). As for data enrichment, the next two attributes `group_ID` (at: *integer*) and `orientation` (at: *real*) can be considered. A description of them can be found in section 4.2.2. The two methods *area()* and *center_of_gravity()* are relevant for the generalization process, especially for the *typification* operator (see more in section 5.3).

---

[10] That is, VECTOR25 and VECTOR200 from the Federal Office of Topography.

Area() computes the area of the building object while the latter method calculates the x- and y-coordinate of the center of gravity of each object.

In addition to the subclasses Build25k and Build200k which represent the corresponding LOD_{25} and LOD_{200} another subclass BuildRes inherits from the main class. The latter contains the generalized building objects of the created map. As the courtyards of buildings are included in LOD_{25}, a separate attribute has been added to meet the requirements of building generalization. Further to all this added information more could be specified to enhance the database for a specific purpose.

4.4.3 Feature Class *River Network*

Figure 4.14 shows the feature class *river network* which consists of one main class River and the inheriting subclasses Riv25k, Riv200k and the 'result' subclass RivMorph.

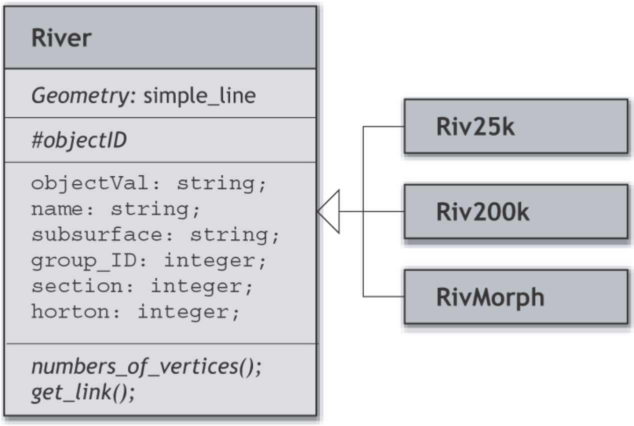


Figure 4.14: Structure of the feature class *river network* with the main class River. The three subclasses Riv25k, Riv200k and RivMorph inherit from the main class.

The main class River is also composed of the five sets. Although in large scale maps river objects are displayed as areal objects, a linear representation has been chosen for the *geometry type* for both LODs. One of the most important attributes is given by `objectVal` (at: *string*) where the type of the river (e.g small river, stream) is defined. Additionally this attribute describes if the object stands for a lake, as a lake axis, in case the size of the lake is too small for the requested scale. Since it is used for the matching process in section 4.3.2 the attribute `name` (at: *string*) has already been introduced. If a river object is not visible on the earth's surface but located subterraneously, it is specified with the attribute `subsurface` (at: *string*). Further to the existing characteristics of the original data set, three more attributes have been added to fit the data set to the defined purpose: `group_ID`, `section` and `horton`. `group_ID` (at: *integer*) is the result of the matching process discussed in the previous section, while `section` (at: *integer*) is similar to the same attribute of the feature class *road network*. The numbering is individual for each river starting at the source and ending at the outlet. Both values as well as the attribute `horton` (at: *integer*) (described in section 4.2.2) are needed for the subsequent generalization process. In addition to the listed methods other methods as with the previous feature classes exist which will, however, not be elaborated on. The method *numbers_of_vertices()* computes the actual number of vertices of each object, while *get_link()*

is used to define the direction of the river object.

The result of the generalization process is included in the subclass RivMorph where the notation indicates the method used. Riv25k and Riv200k represent the original data sets of LOD_{25} and LOD_{200}, respectively.

4.5 Discussion of the Design of the MSDB

Database issues in multiple representations are concerned with how to accommodate the different sources of information in a multi-scale database. The focus of this chapter was on the integration of two data sets (VECTOR25 and VECTOR200) representing different map scales as LOD_{25} and LOD_{200} in a common multi-scale database. Three main points have been discussed: First a description of enriching the data sets for making them more valuable in the real-time generalization process, second a matching and linking process of the LODs and third the design of an adapted data model for the integration of the individual feature classes. The remainder of this chapter is devoted to a discussion of the individual approaches.

4.5.1 The Data Enrichment Process

The data enrichment process plays a very important role for the real-time generalization and thus the on-the-fly mapping process. A lot of information used for creating maps can be pre-computed and therefore reduce the high computation costs caused by the time consuming generalization process. However, adding information of whatever type must be well considered so as not to congest the database with useless information. By means of a well-defined purpose and later utilization of the system specific useful intelligence in the form of attributes and methods should be added to the original data. The type of information discussed in section 4.2 provides possible examples for data enrichment but could be extended further.

One major point in data enrichment should be to try to describe or define pattern or structure in a map. Examples can be an alignment of buildings along a road or a road network arranged in a grid. If such structures, which are not based on the individual object level but at a more abstract level, can be recognized and treated as a 'super object', this will facilitate the whole generalization process and thus maintain the typical characteristics of the content of a map. Approaches in this direction have been made by Christophe and Ruas (2002) for the detection of building alignments, Boffet and Rocca Serra (2001) for the identification of spatial structures within urban blocks and Mustière and Moulin (2002) for describing the spatial context in cartographic generalization.

4.5.2 The Matching and Linking Process

Linking of corresponding objects or groups of objects is crucial for optimal usage of a multi-scale database in the context of cartographic generalization. Different approaches and solutions are known for specific data sets which, however, are not necessarily applicable to other data sets. Examples of such solutions were presented by Sester *et al.* (1998) and Filin and Doytsher (2000). More work on this research topic must follow to amplify the advantage of multi-scale databases. The difficulty of matching corresponding objects in different data sets lies in fact that a $[1:1]$ relation is more an exception than the rule. For example, one building object on a smaller scale always represents a number of objects of a larger scale as a placeholder. So, linking is not always possible (relation $[0:n]$) or not always unique (relation $[n:m]$). As discussed in the previous section linking groups of objects (representing a specific pattern or structure) instead of individual ones can improve the generalization process and help to retain the characteristics of a map. Pattern could then be treated as one (group) object and thus linked as one object. The matching process proposed in section 4.3.4 for the building objects follows this approach of linking groups of objects.

On the other hand the whole matching and linking process is strongly dependent on the data set used, which influences the procedure most significantly. As shown for the feature class *river network* the matching process can be simplified enormously if the names of the rivers are given.

4.5.3 The Proposed Data Model

The proposed data model for the defined feature classes has been designed to be simple, including only the most critical and decisive aspects. It has been attempted to keep the structure of the main class and the subclasses as similar as possible defining a set of five information elements which can be found in all feature classes. The basic principle was to design a data model that is both as logical as possible and as specific as possible.

For a full implementation the proposed data model can be used as a foundation and augmented by all conceivable kinds of attributes and methods. As this system only considers two LODs (LOD_{25} and LOD_{200}) the possibility of adding more LODs is taken into account by the defined structure. With minor changes (adding for example a new class RivX) a further LOD can be stored easily in the multi-scale database.

Chapter 5

Generalization Process

In the previous chapter the focus was placed on the design of a multi-scale database for on-demand web mapping. The advantage of using such a type of database is that missing generalization algorithms or algorithms with high computational costs can be substituted. Additional information enhances the MSDB for a defined environment and purpose. All these positive effects should be used to define a generalization process meeting the requirements of on-demand web mapping.

This chapter presents a generalization process for the selected feature classes: *road network* in section 5.2, *building* in section 5.3 and *river network* in section 5.4. The section 'General Remarks' (5.1) at the beginning of this chapter introduces the context of the generalization process while a short discussion (section 5.6) ends this chapter. A special part (section 5.5) is dedicated to the generalization operator 'displacement' which in practice is not suitable in the chosen context of web mapping.

Besides the main generalization processes for the various feature classes, two generalization techniques will be presented in more detail for

- Linear features: **Morphing transformation technique**. From a start and an end state, also known as 'keyframes' (Foley *et al.* 1996), an intermediate state is generated;
- Buildings: **Mesh simplification technique**. This algorithm offers a possibility to implement the typification operator.

Both techniques will be discussed for the designated feature classes; advantages and disadvantages that arise are pointed out.

5.1 General Remarks

The generalization process proposed in this section is designed for on-demand web mapping, especially for scenario 2 described in section 3.1.2. Note that the MSDB and the proposed approaches rely on data from the Swiss topographic maps[1]. Hence, the characteristics of the Swiss national map series apply. A description of the individual generalization operators can be found in section 2.1.6. Each feature class is introduced by illustrating an excerpt of LOD_{25} and LOD_{200} of the same area, whereby the latter is enlarged to the same dimensions as the former (but not to scale). Other feature classes such as railways or polygonal features such as forest are not discussed in this work.

---

[1] The data sets of the digital landscape models VECTOR25 and VECTOR200 are used.

5.2 Feature Class *Road Network*

The classification scheme of the feature class *road network* is defined inconsistently for the two LODs (cf. section 4.1). This renders the use of these data sets more complex and requires a re-classification. To illustrate the (geometric) differences between these two LODs with regard to the road network, excerpts of the maps representing LOD_{25} and LOD_{200} of the Federal Office of Topography are depicted in Figure 5.1.

Figure 5.1: Excerpt of the Swiss national maps of 1:25'000 (left) and 1:200'000 (right, magnified to 1:25'000) illustrating the different representations of the road network of the same area (not to scale). Data: © swisstopo (BA034957).

On the left a portion of the Swiss National map at a scale of 1:25'000 is shown and on the right the same stemming from the scale 1:200'000 (LOD_{200}) series but re-enlarged to 1:25'000 is presented. Attention should be paid to the different number of roads in the two samples. In the following, two main types of the feature class *road network* are distinguished: *highway* and *main and minor road* (where all feature elements presented in section 4.1 are included).

Highway In both data sets, LOD_{25} and LOD_{200}, all highway objects exist. Modifications in the transition from the large to the small scale are relatively small, mainly affecting interchanges and ramps (e.g. entries and exits of highways).

Main and Minor Road The type *main and minor road* is more interesting because not all elements are depicted in both LODs. At a scale of 1:25'000, all main and minor roads are usually represented because sufficient space is available. At smaller scales, only main roads and important minor roads are displayed. Special elements, such as 'roundabouts', collapse quickly into road sections or connecting nodes.

5.2.1 Outline of the Generalization Process

The proposed generalization process for the feature class *road network* is illustrated in Figure 5.2. This approach is based on the sequential processing of digital line data proposed by McMaster (1989). Shaded boxes in light gray denote the use of generalization operators or other operations while dark gray boxes stand for the data sets used.

Road selection is carried out in a two-step sequence. The first step is based on the comparison of both LODs. Each road segment that exists in both data sets must be represented in the requested target map. The second step decides, based on the road classification and predefined constraints (e.g., attribute: `objectVal`), which road section is shown at which scale range. For the feature element *highway* this step is simple since all objects except ramps are represented in both LODs and thus have to be selected.

Next, it has to be decided whether a 'morphing transformation' takes place or a weeding operator is applied directly. The difference between the two options depends on the data contained in the database. If the same road section is contained in LOD_{25} as well as in LOD_{200} a morphing process can be initiated; otherwise, if a road section is only represented in LOD_{25} the weeding operator is used. 'Morphing' does not belong to the generalization operators in the narrow sense (McMaster and Shea (1992), Bader *et al.* (1999)), because it needs a minimum of two representations to transform the selected object. As a MSDB with two different representations is used in this work, this technique can be utilized for the generalization process. The morphing transformation technique will be discussed in more detail in section 5.4 in combination with the feature class *river network*.

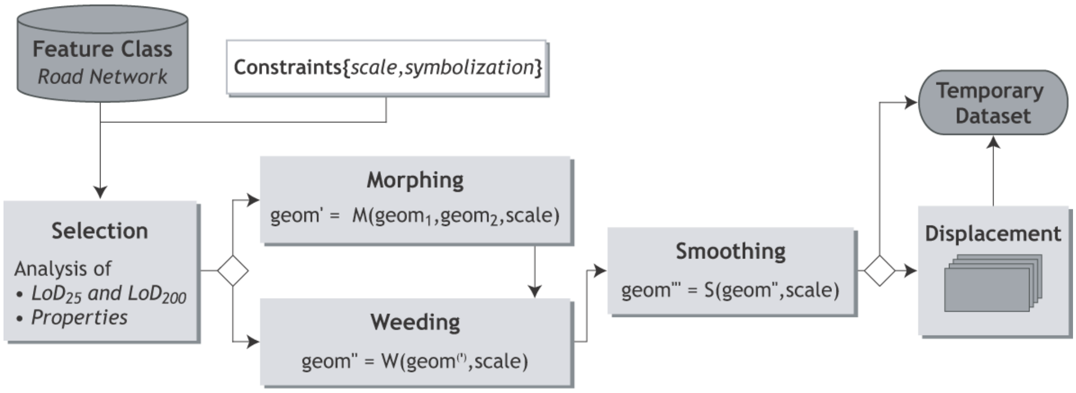


Figure 5.2: Generalization process for the feature class *road network* for the feature elements 'highway' and 'main and minor road'.

For further refinements and to attain a natural impression of the objects in the final map, a smoothing process is applied. The use of this operator adjusts the object's geometry to improve its visual appearance.

The displacement step is optional and carried out only if the user requests a high-quality map and hence is willing to tolerate longer computing and wait times. In the case of highways it will be needed only in specific situations, as highways are of key importance and hence will resist displacement. For the operator 'displacement', all feature classes must be considered. The symbolization (i.e. line width, color type) must be defined either by the user (in advance by defining the preferences) or by default values adapted to the user's environment and the target map scale.

Operator	Description	Algorithm
Selection	The roads are selected by comparison of LOD_{25} and LOD_{200} or with the aid of the attribute `objVal`.	Approach presented by Thompson and Richardson (1995).
Morphing	LOD_{25} and LOD_{200} form the 'key frames'. The intermediate scale (requested scale) is generated from these data sets by a morphing algorithm.	Morphing algorithm based on Sederberg and Greenwood (1992) and Monmonier (1989).
Weeding	If no morphing transformation is possible a weeding process is carried out.	Douglas-Peucker algorithm (Douglas and Peucker 1973) or Visvalingam and Whyatt (1993).
Smoothing	For maintaining the natural morphology of line objects a smoothing follows the weeding operation.	Gaussian smoothing algorithm (Badaud *et al.* 1986).
Displacement	If road objects overlap or if they are too close to each other a displacement takes place.	Displacement algorithm using elastic beams (Bader 2001).

Table 5.1: Generalization operators for the feature class *road network* with corresponding algorithms.

Table 5.1 summarizes the generalization operators and possible corresponding algorithms that are best suited for the map creation process in the context of on-demand web mapping, where the displacement operation is optional depending on the requirements of the user.

5.3 Feature Class *Building*

Buildings are displayed in different shapes depending upon the scale. At scales of 1:25'000 and larger most objects are represented as polygon elements, while at scales smaller than 1:100'000 a representation as rectangle is more common, with the exception of large buildings (e.g. factory buildings). As Figure 5.3 shows, the smaller the map scale the simpler the outlines of buildings are. In densely populated areas (e.g. town centers) they may simply be represented as city blocks. At small scales the representation changes: the built-up area is now shown as tinted polygons instead of individual buildings. For Swiss national maps this change takes place at the transition to 1:500'000. In VECTOR25 and VECTOR200, both scales use individual buildings. Since no additional semantic attributes are available for building objects (e.g. describing the type of building) no further feature elements can be distinguished in attribute `objectVal`[2]. More information, however, would be helpful for the generalization process to estimate the importance of each object.

After reducing the scale of a representation of spatial objects, many objects can no longer be clearly perceived as they may have become too small. This is particularly the case for building objects. Therefore, in order to make the small scale representations compatible with the requested map scale the generalization process must cover at least the following two basic

---

[2] Neither VECTOR25 nor VECTOR200 encode more detailed information for individual buildings.

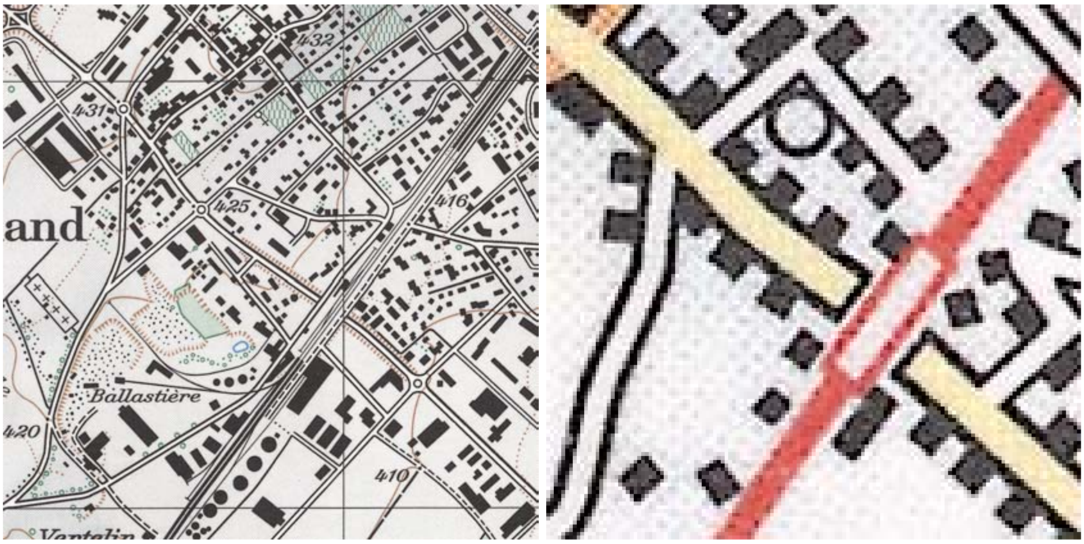


Figure 5.3: Excerpt of the map of 1:25'000 (left) and 1:200'000 (right, magnified to 1:25'000) illustrating the different number of building objects representing the same number in reality (not to scale). Data: © swisstopo (BA034957).

types of operators:

- Enhancement: Adapting the size of buildings to the target scale (primarily by enlargement);
- Simplification: Eliminating details of the building outlines.

Enhanced building objects invariably will lead to a larger footprint. Hence, overlaps and conflicts among buildings and with other feature classes can potentially arise. Thus additional operators – adapted to on-demand web mapping – must be included.

Three further generalization operators are eligible:

- Selection: Extraction of purpose and scale adapted objects or groups of objects, based on database attributes;
- Typification: Transformation of an initial set of objects into a subset, while maintaining the distribution characteristics and pattern of the original set;
- Displacement: Moving objects to eliminate overlap conflicts and when distances between objects are too small.

From those three additional operations, all are not equally feasible for on-demand web mapping (see section 3.5). *Selection* as an important operation at the beginning of every generalization process (cf. 2.1.6) is well suited and needed to reduce the amount of data for web mapping. *Displacement* is strongly time-consuming and thus not really usable in a real-time environment. For the operation of *typification*, an adapted method must be developed to be used in this context.

5.3.1 Outline of the Generalization Process

The main task of settlement generalization for smaller scales using individual buildings is to maintain the overall pattern and arrangement of the large scale representation. Hence, particularly in dense areas (like town centers), typical arrangements of buildings (cluster alignments) must be identified and defined in the MSDB. These groups of buildings must be linked across the different LODs as presented in section 4.3.4. The generalization process for the feature class *building* is summarized in Figure 5.4.

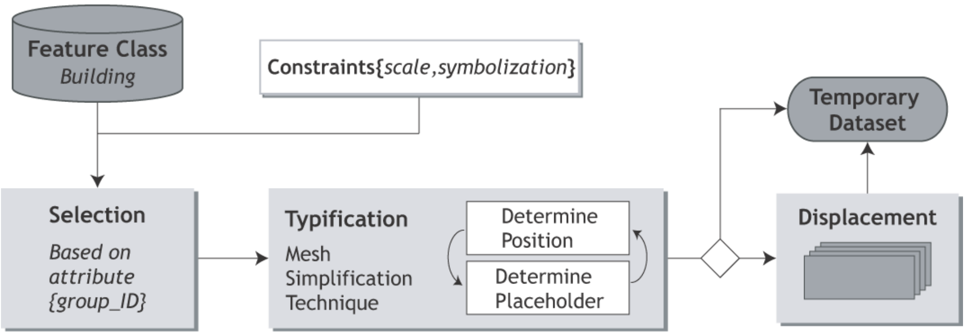


Figure 5.4: Generalization process for the feature class *building*.

As explained above, the selection step can be alleviated by using previously defined cluster elements. In this case not individual buildings but all objects of a group with a corresponding `group_ID` are selected. The *typification* process creates a new, reduced set of objects, which shows similar characteristics concerning density and orientation. As presented in the AGENT-report (Bader *et al.* 1999) the task of this operator is described as a combination of several basic algorithms (selection, aggregation, simplification) into one algorithm. Based on this subdivision of the complex operator *typification* into operation which are more easily performed, a technique called **mesh simplification** is proposed. A more detailed discussion of this technique is presented in section 5.3.2. The detection of overlapping building objects triggers displacement. As mentioned above, however, this operator is not really suited for the real-time context and hence only executed if the user requests a high-quality cartographic output and is willing to tolerate longer wait time.

Special buildings are objects that are represented by individual symbols on large and small scale maps. Examples are churches, castles, power plants etc. In these cases, the generalization process can be executed easily: in many cases the relevant elements can be selected (comparison with 1:200’000) and replaced by a previously defined symbol. Overlap conflicts with other map objects can be solved by either removal or, alternatively, by displacement in the case of very important buildings. Since the semantic information about the building type is not included in the original data set, however, the feature class *building* has not been further separated into feature elements. Table 5.2 gives an overview of the proposed generalization operators with corresponding algorithms. For the (optional) displacement operation objects of all feature classes must be considered.

5.3.2 Mesh Simplification for Building Typification

The mesh simplification technique is related to *mesh optimization techniques* well known in the research area of computer vision which hold great potential for many applications. These kinds

Operator	Description	Algorithm
Selection	Buildings are selected by the attribute `group_ID` in LOD_{25}. Objects with same value are processed by the following typification process.	Subset retrieval based on attributes in the database.
Typification	Generates a reduced set of buildings that exhibits the typical arrangement as displayed at a smaller scale.	Mesh simplification technique.
Displacement	Can be executed when individual objects overlap.	Displacement algorithm using elastic beams (Bader 2001).

Table 5.2: Generalization operators for the feature class *building* with corresponding algorithms.

of techniques have been mainly used for surface reconstruction from sampled data, occurring in many scientific and engineering domains.

The main reason for using a mesh optimization technique is to reduce the amount of data. Thereby three goals or purposes can be defined: i) faster rendering, ii) reduced storage volume, and iii) simpler manipulation. A lot of research work has taken place in this topic. Mesh optimization, as considered by Turk (1992) and Schroeder *et al.* (1992), refers to the problem of reducing the number of faces in dense meshes (usually made up of triangles). The contribution of this early work was the development of a method for smoothly interpolating between models representing the same object at different levels of detail. Two main disadvantages, however, impede using this approach in our work. First, the method is best suited for models that represent curved surfaces. Secondly, the required algorithm is very complex and time-consuming and thus not adapted for the context of on-demand web mapping.

The research work of Hoppe *et al.* (1993) has shown that mesh optimization can be put into use effectively in at least two applications: surface reconstruction from unorganized points, and mesh simplification (the reduction of the number of vertices in an initially dense mesh of triangles). Their principal idea is to describe mesh simplification as an optimization problem, defining an energy function E that directly measures deviation of the final mesh from the original. To solve the optimization problem they minimize the energy function E that captures the competing objectives of a tight geometric fit and a compact representation. One of the main disadvantages concerning this approach is the very time-consuming computation of the energy minimizing function E. Based on the method of Turk (1992) and Hoppe *et al.* (1993) for mesh optimization, an adapted, less time-consuming energy minimizing function has been developed for on-demand web mapping.

The problem considered in this work can be stated as follows:

> *Given a collection of data points* X *in* $\mathbf{R}^2$ *and an initial triangulation mesh* M_0*, a mesh* M_f *with a smaller number of vertices is sought that fits the original data well.*

Figure 5.5 illustrates the main idea of reducing the amount of vertices of a mesh (*mesh simplification*) for multiple representations preserving the original characteristic vertex distribution. On the left hand side of the Figure, a dense mesh made up of a large number of vertices describes the front of a 'face'. In the middle part, the amount of vertices is reduced maintaining the typical outline and characteristic of the original form. On the right hand side a strongly simplified version, with a fraction of the original vertices of the starting data set, is depicted.

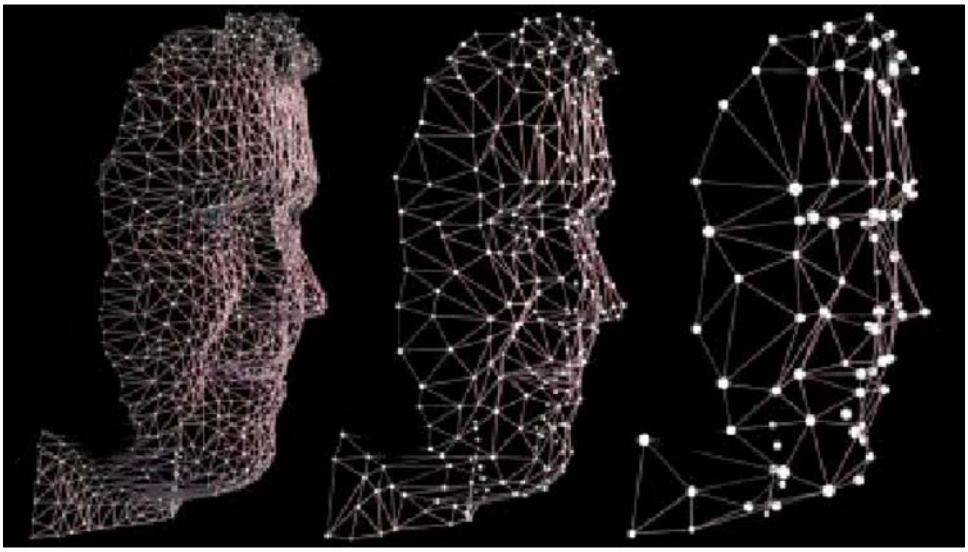


Figure 5.5: Example of reducing the number of vertices of a mesh (from Dudziak (2000)).

Each state best represents the original form. Thus the selection of points plays a crucial role and must be optimized to preserve the original arrangement. Regions with a dense number of vertices should be maintained in each state or level as dense areas and vice versa with sparse parts. This is well illustrated in Figure 5.5. Transferring this idea to cartography and particularly for building representation, the various states of the 'face' in Figure 5.5 can be looked upon as different scales or LODs of a map. The vertices do thereby not display the buildings themselves but, for example, the centers of gravity of building objects.

Starting from a mesh M_0 with n vertices (representing the centers of gravity of the individual buildings) a new mesh M_j with $(n-1)$ vertices is sought. This new mesh should best represent the original one with minor changes. Hoppe *et al.* (1993) define a mesh M as a pair (K, V), where: K is a simplicial complex representing the connectivity of the vertices, edges and faces; $V = \{v_1, ..., v_m\}$, $v_i \in \mathbf{R}^3$ is a set of vertex positions defining the shape of the mesh. To obtain a mesh that provides a good fit to the original point set X an energy function $E(K, V)$ is defined where:

$$E(K,V) = E_{dist}(K,V) + E_{rep}(K) + E_{spring}(K,V)$$

By varying number, position and connectivity of the vertices a minimization of this value is looked for. The distance E_{dist} is equal to the sum of squared distances from the point set X of the mesh. The value E_{rep} is proportional to the number of vertices. E_{spring} is looked upon as a regularizing term and describes the sum of the edge lengths. Using this energy function E for a mesh optimization in principle provides the possibility of observing several constraints which restrict the generalization process (such as the distance between the objects, building alignment, etc.). But since, on the one hand, the minimization of the energy function $E(K, V)$ is computationally very time-consuming (Puppo and Scopigno 1997) and on the other, only *one* constraint (distance between the objects) has to be considered in this work, a simpler function, more adapted for on-demand web mapping has been derived.

Before explaining the various steps of the proposed mesh simplification technique in more detail, the full approach of typification should be discussed. The typification procedure, as shown in Figure 5.4, is composed of two steps which are not independent and interact with each other:

- **Position**: Determining the *number* and the *position* of the new objects with respect to the requested scale;
- **Representation**: Creation of a new building objects at the determined position.

The typification process (mesh simplification) is an iterative process where the termination criterion is dependent on the requested map scale RS_U. By means of the original number of objects m of LOD_{25} and the scale value RS_U, an adapted number of building objects $t(m, RS_U)$ is computed for terminating the iteration. The basic idea is, that (concerning the feature class *building*) two objects which lie next to each other can be replaced by a new one between them, called a representative (placeholder). This simple operation is known as **edge collapsing** (ecol) (Hoppe 1996) and is sufficient for effectively simplifying meshes. As shown in Figure 5.6, an edge collapse transformation $v_n = ecol\{v_s, v_t\}$ unifies two adjacent vertices v_s and v_t into a new single vertex v_n.

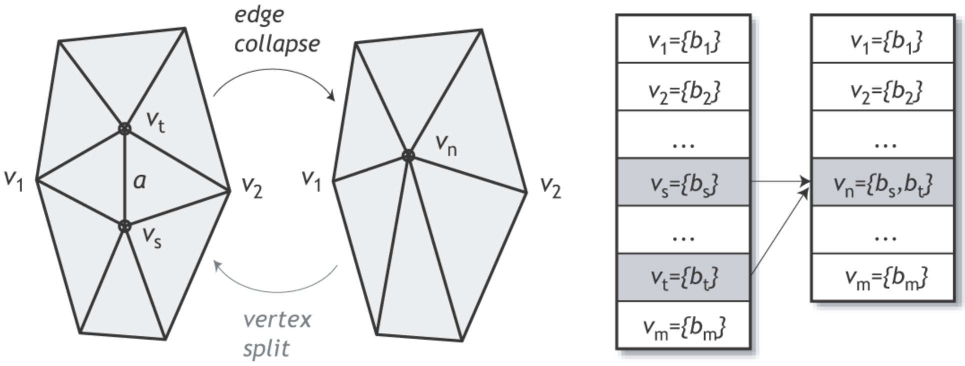


Figure 5.6: The method of edge collapsing for mesh simplification (left). On the right a sequence of edge collapses is shown. v_i describes the center of gravity of building b_i.

The inverse transformation *vertex split* adds a new vertex v_t and thus two new faces to the original mesh. Since typification must reduce the number of objects, the inverse case is not relevant here. Transferring the idea of edge collapsing to the feature class *building*, where each vertex represents a building object, helps to solve the first step of the typification process (position). From the number of objects in LOD_{25} and LOD_{200}, the adapted number nb of objects with individual position $\{x, y\}$ for the target map scale can be computed (Figure 5.7, left).

For the second step of the typification process (*determining the placeholders*), each remaining vertex v of the final mesh M_f must know which building or buildings it represents. For example, in Figure 5.6 the placeholder v_n represents the vertices v_s and v_t and thus the building objects b_s and b_t. From the geometric information of the represented objects (e.g. area A, orientation α) a new best fit building object (i.e. a placeholder) must be created for the requested map scale RS_U (Figure 5.7, right).

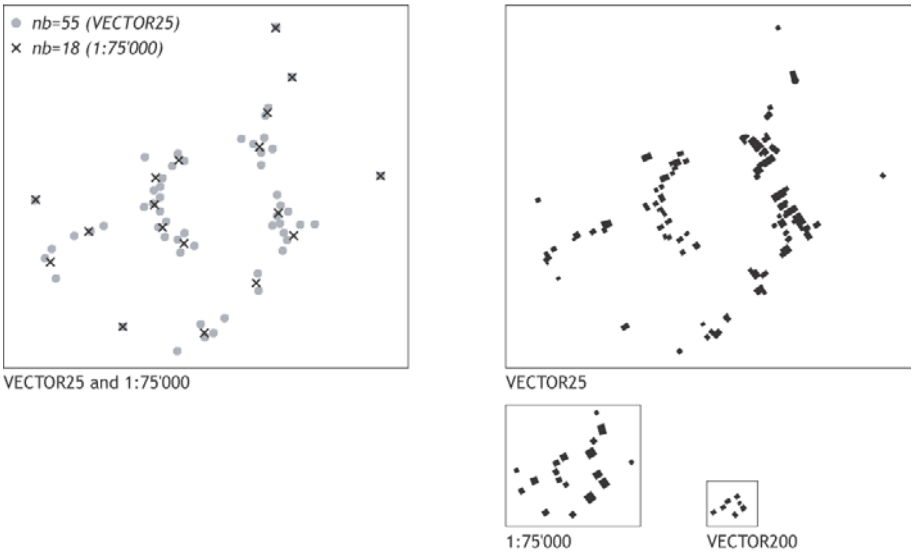


Figure 5.7: Left: The positions of the placeholders for scale 1:75’000 (not to scale). Right: The dimensions of the placeholder computed out for scale 1:75’000.

5.4 Feature Class *River Network*

The hydrologic network normally consists of the feature classes *river network* and *lake*. In this work only the linear feature class *river* has been considered to propose a generalization process. Figure 5.8 illustrates the differences between LOD_{25} and LOD_{200}. On the left an excerpt at a scale of 1:25’000 is shown; the same area at 1:200’000 is represented on the right. A river network at a large scale depicts many elements of the network – from the largest river to the smallest tributary. At a small scale, the large number of tributaries complicates the map to the point of uselessness and the network is thus pruned considerably. Generalization then provides a more effective presentation of the characteristics of the network.

In general a river can be modelled as an area (polygon) or a line (polyline) element. River objects are represented as linear elements at smaller map scales – in contrast to larger map scales. At larger scales both forms of representation can be found depending on the symbology defined by the map producer. Both in VECTOR25 and VECTOR200, however, the elements are represented by the medial axis of the feature object (i.e. as lines). This fact simplifies the generalization process considerably because no collapse operation for areal features (by computation of a medial axis) needs be used. Another advantage of the representation of this feature class in VECTOR25 and VECTOR200 is that the line objects are oriented by the stream direction and thus a hierarchical structure can be defined. Just as with the road network, the rivers are classified into two main elements (*Bach* (=’stream’) and *Fluss* (=’river’)). This classification is not too important for the proposed process. Of more relevance is the attribute `name`, which helps to easily identify and thus compare the same objects in both LODs.

An aside about the feature class *lake*: Lakes will be retained as long as possible through the scales. In most cases they are represented as a polygon. If the area of a lake falls below the minimum size limit it collapses to a linear element and is treated as a river object.

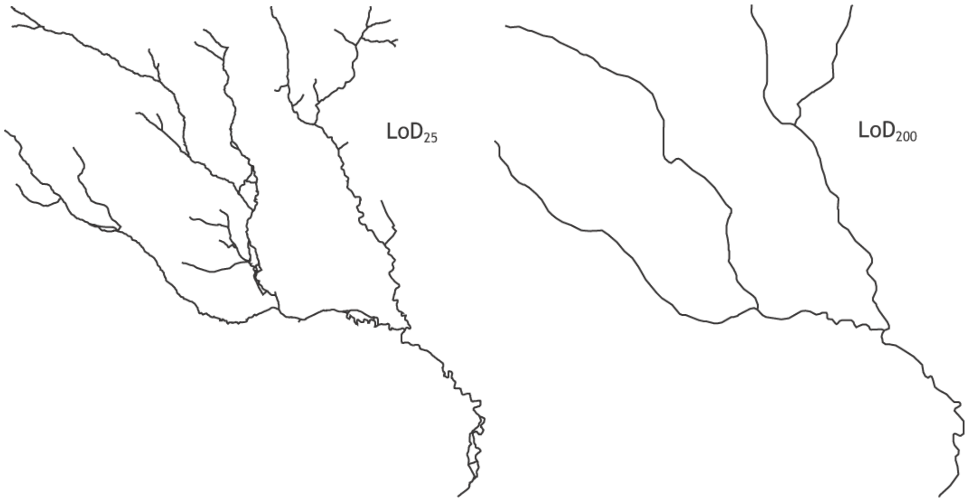


Figure 5.8: Excerpt of the hydrologic network of 1:25'000 (left) and 1:200'000 (right) illustrating the complexity of the river network (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

5.4.1 Outline of the Generalization Process

For the generalization of the river network a similar process can be accomplished as for the road network or other linear objects. In contrast to other linear features, however, this feature class possesses two main advantages which can simplify the generalization process:

- Hierarchical structure of the network, branching structure (which can be expressed by the Strahler order);
- Definition of a stream order (expressed by Horton order).

Both of these ordering schemes facilitate the selection (by tree pruning) as well as the simplification operations, which dominate the generalization process. Based on this information, it is possible to simplify the hydrologic network preserving the main pattern characteristics. Figure 5.9 describes the generalization process defined for the feature class *river network*.

For 'selection', a two-step procedure is defined:

1. The first step involves a selection by comparison of both LODs. If a representation of an object exists in LOD$_{25}$ as well as in LOD$_{200}$ a computation of a representation is needed for the requested map scale.
2. In the second step, more rivers can be selected (or eliminated) by extracting the stream order. This stream order (as introduced in section 4.2.2 as element of data enrichment) is relatively simple but effective in ordering stream segments and network pruning. For the hydrologic network, the Horton stream ordering method (Horton 1945) most closely approximates the generalization decisions made by a human cartographer. By means of this order, a simple selection of additional objects is possible.

The next operation of the generalization process is affected by the existing information in the LODs. If an object is represented in both LODs a 'morphing transformation' is carried

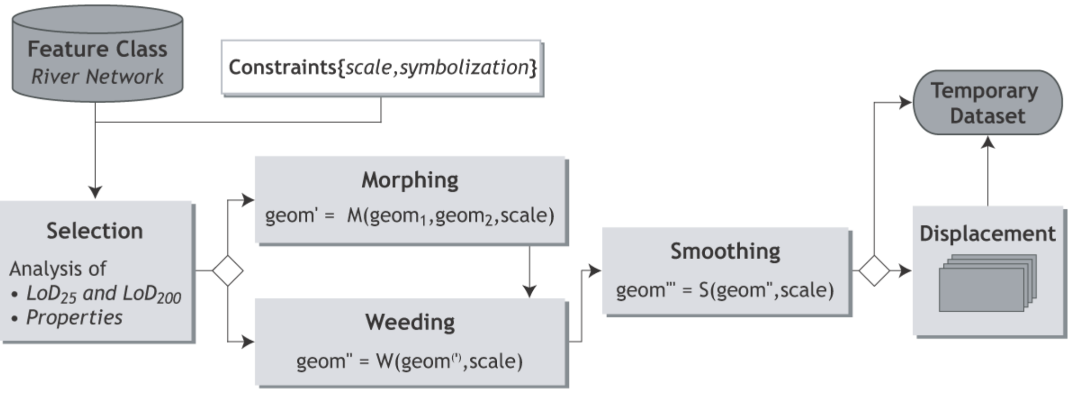


Figure 5.9: Generalization process for the feature class *river network*.

out, otherwise a 'weeding' process is performed. Morphing (described in more detail in section 5.4.2) is well suited for being used in context with MSDBs to create an intermediate state out of two 'keyframes' (i.e. a start and an end state). Each LOD of the database stands for an individual state or keyframe. In this case the start state is described by LOD_{25} while the end state is defined in LOD_{200}. By means of morphing it is possible to create an intermediate shape for the requested map scale in the range defined by the extreme scales. This approach of morphing, also known as (image) warping or in-betweening, is widespread in several disciplines, mainly in the area of computer vision (Shapira and Rappoport 1995).

If no corresponding object is available in the small-scale representation (i.e. if no end state can be established) a weeding process based on the Douglas-Peucker algorithm (Douglas and Peucker 1973) is carried out. The same process is performed after the morphing to adapt the granularity[3] to the requested map scale.

Next, a smoothing operation is applied in order to refine the shape of the generated polylines and give them a more natural, smooth appearance. In case of overlap or congestion problems a following displacement operation can be enforced if scenario 3 is used.

Table 5.3 summarizes the operations needed for the generalization process of the feature class *river network*. The last column describes algorithms that may help to implement the corresponding generalization operators.

5.4.2 Morphing

The advantage of using a morphing transformation for the simplification operation is that the process can be controlled by the LODs contained in the MSDB. Because the shapes are represented in the keyframes (i.e. the two extreme LODs) the simplified shapes of the intermediate representations are constrained. Hence, the quality of the data set used to generate the LODs in the MSDB can bear influence on the generalization process.

The term 'morph' (derived from the Greek word 'morphe', meaning 'shape')[4] has, in recent years, acquired the meaning of 'shape changing'. This technique of continuously transforming an object into another object has become increasingly popular in computer graphics and

---

[3] *Granularity* is the relative size, scale, level of detail, or depth of penetration that characterizes an object or activity. It can refer to the level of a hierarchy of objects or actions, to the fineness of detail in a map.

[4] Out of the 'The American Heritage® Dictionary of the English Language' (*http://www.bartleby.com/*).

Operator	Description	Algorithm
Selection	The rivers are selected by comparison of LOD_{25} and LOD_{200} and the attributes `objVal` and `Name`. Additionally a selection by a predefined ordering scheme can be accomplished.	Tree pruning by Horton order.
Morphing	LOD_{25} and LOD_{200} form the 'key frames'. The intermediate scale (requested scale) is generated from these data sets by a morphing algorithm.	Morphing algorithm based on Sederberg and Greenwood (1992).
Weeding	If no morphing transformation is possible a weeding process is carried out.	Algorithms by (Douglas and Peucker 1973) or Visvalingam and Whyatt (1993).
Smoothing	For maintaining the natural morphology of the line object a smoothing follows the weeding operation.	Gaussian smoothing algorithm (Badaud *et al.* 1986).
Displacement	If river objects are too close to each other or to other objects a displacement takes place.	Displacement algorithm using elastic beams (Bader 2001).

Table 5.3: Generalization operators for the feature class *river network* with corresponding algorithms.

computer vision. Typically, in 2D morphs, control points are placed in two images at locations that correlate. A 2D morphing algorithm then calculates intermediate images, that, when viewed in succession, smoothly change the first image into the second. Most approaches for morphing techniques are available for raster data (image warping) and reviewed in detail in Wolberg (1990).

In the context of vector data, this problem is also known as shape transformation or shape blending (Sederberg and Greenwood (1992), Shapira and Rappoport (1995)). The shape blending approach of Sederberg and Greenwood (1992) presents an algorithm for smoothly blending between two 2D polygonal shapes. Based on a physical model where one of the shapes is considered to be constructed of wire, a solution is found if the first shape can be bent/streched into the second shape within a minimum amount of work. The disadvantage of this approach is that the number of vertices of the two polygons must be equal and the shape must describe a *closed* polygon. Another weak point is that several parameters must be set at the beginning of the procedure.

Morphing between two different 'keyframes' consists of a two-step procedure: 1) finding the correspondence between the two shapes and 2) an interpolation process. The first part of finding corresponding elements is covered by the MSDB, where this linking information is stored. Known as *data matching* the step is carried out in advance and does not affect the time needed for the map creation process. A detailed discussion of data matching and linking of corresponding objects was given in section 4.3. For the second part – the shape interpolation – an approach based on the idea of Monmonier (1989) adapted to on-demand web mapping has been developed.

Figure 5.10 should illustrate the basic principle of a morphing transformation with respect to linear features such as a river network. The main condition is that for the same object different representations exist. In relation to cartography, the feature objects must be described in a minimum of two frames at two different scales.

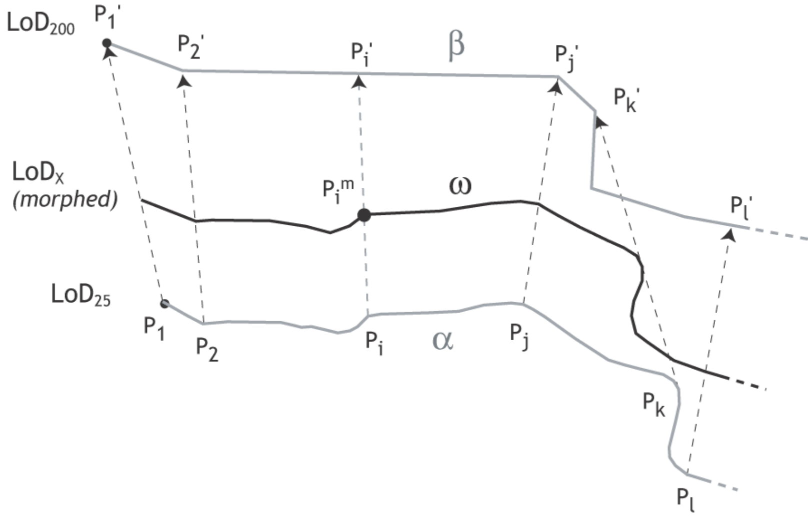


Figure 5.10: The principle of a morphing transformation is to create an intermediate state out of two 'keyframes' represented in LOD_{25} and LOD_{200}.

In this work a data set with two LODs is given – LOD_{25} and LOD_{200} – where the representations of the border frames are defined. Using this information a representation for an intermediate state can be computed. Starting with the representation α, from LOD_{25}, described by the vertices $P = [P_1, ..., P_n]$ and β from LOD_{200} with vertices $P' = [P'_1, ..., P'_m]$, a representation ω for the requested scale $X(= RS_U)$ is sought. For calculating the morphed vertices P^m, two methods can be selected: i) allocation of corresponding vertices $P_i \rightarrow P'_i$ (illustrated by Figure 5.10); ii) triangulation over all points P and P'. Both techniques are presented and discussed more detailed in section 6.4.2.

5.4.3 Smoothing Operator

Smoothing is a process by which data points are averaged with their neighbors in a series. This has the effect of planing the sharp edges of the original data set. Many methods for accomplishing a smoothing process exist (a classification can be found in McMaster (1989)). In this work the Gaussian smoothing method (Badaud *et al.* 1986) is implemented and discussed. The smoothing operator will be used for the feature class *river network* in two different ways:

- Preprocessing: To prepare the data for the MSDB and for the subsequent generalization processing, a smoothing operation is applied. This helps to eliminate digitization inaccuracies and plane away small perturbations, which interfere the morphing transformation.
- In the generalization process: After morphing a (weeding and) smoothing process follow (Figure 5.9). In this case, the granularity of the line segments should be reduced and adapted to the requested map scale.

Both applications are needed to get a suitable cartographic result. Depending on the quality of the original data set the first smoothing process can be omitted. The second one is mainly responsible for the natural appearance of the river network.

5.5 Displacement

Feature displacement is used if two or more feature objects are in conflict, for example, if the distance between them falls below a tolerance value or an overlap situation occurs. To resolve or minimize these problems two approaches are available: a) a re-selection/elimination considering the conflict location and the related objects; and b) a spatial displacement (shifting) of the location of the objects. In the context of on-demand (web) mapping the displacement step is strongly dependent on the selected scenario (as defined in section 3.1.2). For scenario 1, no or only marginal corrections are carried out. Compared to scenario 1 scenario 3 allows the use of the most sophisticated procedures, with a preference for spatial displacement. The second scenario describes a special case, where the time component plays such an important role that no major displacement efforts can be undertaken. Instead, the approach a) above is used and an appropriate elimination of the objects concerned is executed. For example, if a building causes a conflict the elimination of the corresponding object can be the best suited solution. The alternative approach is to prevent or minimize conflicts instead of solving them. Using a MSDB the potential for possible interference can be reduced. The generalization of a LOD to a scale that is near to the target map scale potentially induces fewer conflicts than generalizing from a more detailed data set. With these methods it should be possible to solve the problem of overlaps or congestion to a certain degree. More research work concerning feature displacement can be found in the literature (Barrault and Bader (2002), Højholt (2000)). In this work the displacement problem will not be dwelt upon further.

5.6 Discussion

5.6.1 General Remarks

Using a multi-scale database with several levels of detail for the generation of a specific scale demands adapted methods and techniques to properly exploit the available information. Especially in the context of on-demand web mapping, the advantages of working with a multi-scale database must be utilized. The proposed map generalization processes for the selected feature classes answer this purpose. Two of the generalization techniques proposed above benefit directly and particularly from the use of a MSDB and will thus be elaborated on in more detail in chapter 6:

- **Mesh Simplification**: using several LODs the mesh simplification process can be accomplished to generate an appropriate, typified subset of buildings;

- **Morphing Transformation**: out of two 'keyframes' an intermediate state can be created.

A point not to be underestimated is the quality of the scale representations (LODs) used. It is not possible to produce a quality map with incorrect or improper data set represented in the MSDB. The quality of the original data sets, in each case, plays a **critical** role and thus it is decisive for the following map creation process.

5.6.2 Mesh Simplification

The mesh simplification method as described above is best suited for the feature class *building*. The vertices of the mesh are determined using the centers of gravity of each individual building object. By applying a process on this mesh an adjusted representation for a requested scale can be computed. This happens using an iterative process in which new objects are defined to represent the original ones. The number of new objects decreases by one in each loop of the iteration. Thus a stopping criterion must be defined taking into account the requested scale and the bounding LODs (LOD_{25} and LOD_{200} in this case). Knowing the information of the maximum (LOD_{25}) and the minimum (LOD_{200}) number of building objects permits a scale-adjusted number to be computed (see more in next chapter). This iterative operation of mesh simplification is computationally complex and depends on the number of vertices (objects). With a large number of objects, the computation time increases disproportionately since the shortest distance a must be found between all vertices. By means of the groups defined beforehand in the MSDB (for the matching process) the number of objects involved, however, can be kept small. Hence, groups with a small number of objects in a range of $20 \leq m_{25} \leq 80$ are optimal and typical for this technique. Also, since only simple calculations are carried out the proposed procedure is stable and robust. All this makes the method well qualified for use with the feature class *building* but not well-suited for linear objects, such as *rivers* or the *road networks*.

5.6.3 Morphing Transformation

In all existing classifications of generalization operators in the literature, the method of *morphing* is never listed. The main reason is that the classical operators always start from one detailed data set and trys to generate a representation for a certain scale. By contrast, the morphing transformation technique needs two different states, a start and an end state. With the help of these two keyframes a representation defining an intermediate form can be generated. The advantage of using a multi-scale database with this method is that each LOD can act as keyframe.

In the context of cartographic generalization, the method works correctly particularly for linear features (river and road network). The reason is that, for this approach, a cardinality of [1:1] must be given. For each feature object in LOD_{25} *one* counterpart in LOD_{200} must exist. From this constraint it follows that morphing is not really practical for feature classes with a [n:m]-relation ($n, m \in \mathbb{N}_0$, $\setminus$[1:1]). Thus it has not been implemented for the feature class *building*. Another drawback concerning buildings is that the morphing method does not accentuate small characteristic parts of building objects (e.g. bay window) but smoothes them away (which is not always desired). A disadvantage for linear features is the fine-grained resolution of vertices created in the morphing process. This can be compensated for by carrying out a smoothing or a simplification procedure on the morphed lines afterwards.

Chapter 6

Implementation of the Concept

In the previous chapter the generalization process for the selected subset of feature classes selected was presented. This chapter sets the focus on the implementation of the proposed methods, both the matching process as well as the generalization step. A discussion and an evaluation of the mesh simplification and morphing transformation technique conclude this chapter.

6.1 Platform Characteristics

For the implementation of parts of the conceptual framework, in particular the *mesh simplification* and the *morphing transformation technique*, the Gothic LAMPS2 system of the company Laser-Scan Ltd. has been chosen. Due to its strength in generalization developed as a consequence of the AGENT project (Barrault *et al.* 2001), this commercial system for map production had the priority over other software systems. Gothic LAMPS2 is a production system for the creation and updating of maps, charts and geographic data products, centered on an object-oriented database. Handling with objects means that each class holds the opportunity to include operations and attributes to make the data set more valuable. For the development of operations or other applications and extensions in Gothic a programming language named LULL[1] is provided. LULL itself is a simple language, which has similarities with the languages *C* and *Pascal* and is thus not object-oriented but procedural. In combination with the object-oriented database, the overall LAMPS2 system, however, nevertheless assumes an object-oriented behavior. The power of the LULL language lies in functions that are executed when they are called. Knowing LULL means knowing the libraries in which the functions are organized.

By the definition of a schema each data set can be imported to the system. For the insertion of the selected data set (concerning the data models of VECTOR25 and VECTOR200 in our case) into the object-oriented database, which is acting as MSDB, a schema based on the data model presented in section 4.4 has been designed.

6.2 Implementation of the MSDB

This first part of the chapter is dedicated to the implementation of the multi-scale database by means of the proposed data model. The main focus is set on the one hand on the realization of the matching process mentioned above and on the other hand on describing the kind of information added to the original data set within the data enrichment process.

---

[1] *Laser-Scan User Language.*

6.2.1 Test Area

For the case study the data set described in section 4.1 is used. It includes LOD_{25} and LOD_{200}, both derived from the data models of VECTOR25 and VECTOR200[2]. The selected test area is located in the western part of Switzerland around the region of Nyon. All examples included in this thesis are within the map sheet Nyon (sheet number 1261). The overall region covered by the map sheet amounts to 210 km^2 (12 km x 17.5 km) and describes a partly sparsely populated area bordering on the lake of Geneva with two small towns and several larger and smaller villages.

For the implementation of the multi-scale database the three feature classes mentioned above, *road network*, *building* and *river network* have been chosen.

6.2.2 Matching Process of the Feature Class *Building*

An overview of the workflow of the matching process for the feature class *building* was described in section 4.3.4. With buffering a merging or separation of clusters of objects belonging together (for example lying next to each other) is obtained. These groups of objects can be used for the later typification step of the generalization process, accomplished by the mesh simplification technique. Figure 6.1 summarizes the approach of matching building objects.

Starting from the building objects in LOD_{25} a buffer with distance d is computed around each house. As can be seen in Figure 6.1 a) one widely ramified polygon is the result. To subdivide this large polygon a partitioning is needed where the characteristics of the settlement area are best maintained. This can be realized by subdividing the whole area by the road network of LOD_{200}. The two main reasons for this partitioning (with LOD_{200}) can be described as follows: 1) the road network consists only of the most important roads and characterizes thus a more or less evenly spread mesh over the whole area; 2) the roads in LOD_{200} describe at the minimum the ones appearing in the final map. Hence, it is more appropriate to use the road network of LOD_{200} than the one of LOD_{25} for area partitioning. Each newly created partition is assigned a value (group_ID) used for the matching process. Those partitions (as illustrated in Figure 6.1 b) determine the region to which a building pertains; for LOD_{25} as well as for LOD_{200}. On the basis of the center of gravity each individual building of LOD_{200} is assigned to one partition cell getting for the attribute group_ID the corresponding value (Figure 6.1 c). The advantage of creating these groups of objects is that the mesh simplification process need not be applied on all buildings but only on the objects of one group. Since the amount of objects in a single group is usually small the computation process is less time-consuming than if it was applied over all building objects within the study area. Table 6.1 illustrates the number of groups depending on the buffer distance d. The number of buildings within the test area amounts to 11'547.

With distance $d = 0$ each individual object defines a group[3]. If d increases the number of groups BUF diminishes dramatically at the beginning and nears to a limiting value. For the values $d = \{80, 100, 120\}$ the amount B_{Poly} after the road partitioning process is showed in the third row, whereby the ratio $\frac{B_{Poly}}{BUF}$ is between 1.37 for $d = 80$ and 1.75 for $d = 120$. Regarding table 6.1 it can be pointed out that the biggest changes concerning the number of groups happen for smaller distances ($d < 140$). To avoid groups with too few objects the value of d should not be too small ($d > 60$). The reason is the trade-off between:

- computing the mesh simplification process for too many groups with a small number of objects (for the process a minimal number of object within a group is required); and
- computing it for few groups with many of objects (the computation time arise stronger than the number of objects).

---

[2] The data sets have been provided by the Swiss National Mapping Agency (swisstopo).

[3] The distance d defines an area around each building object starting from the outline.

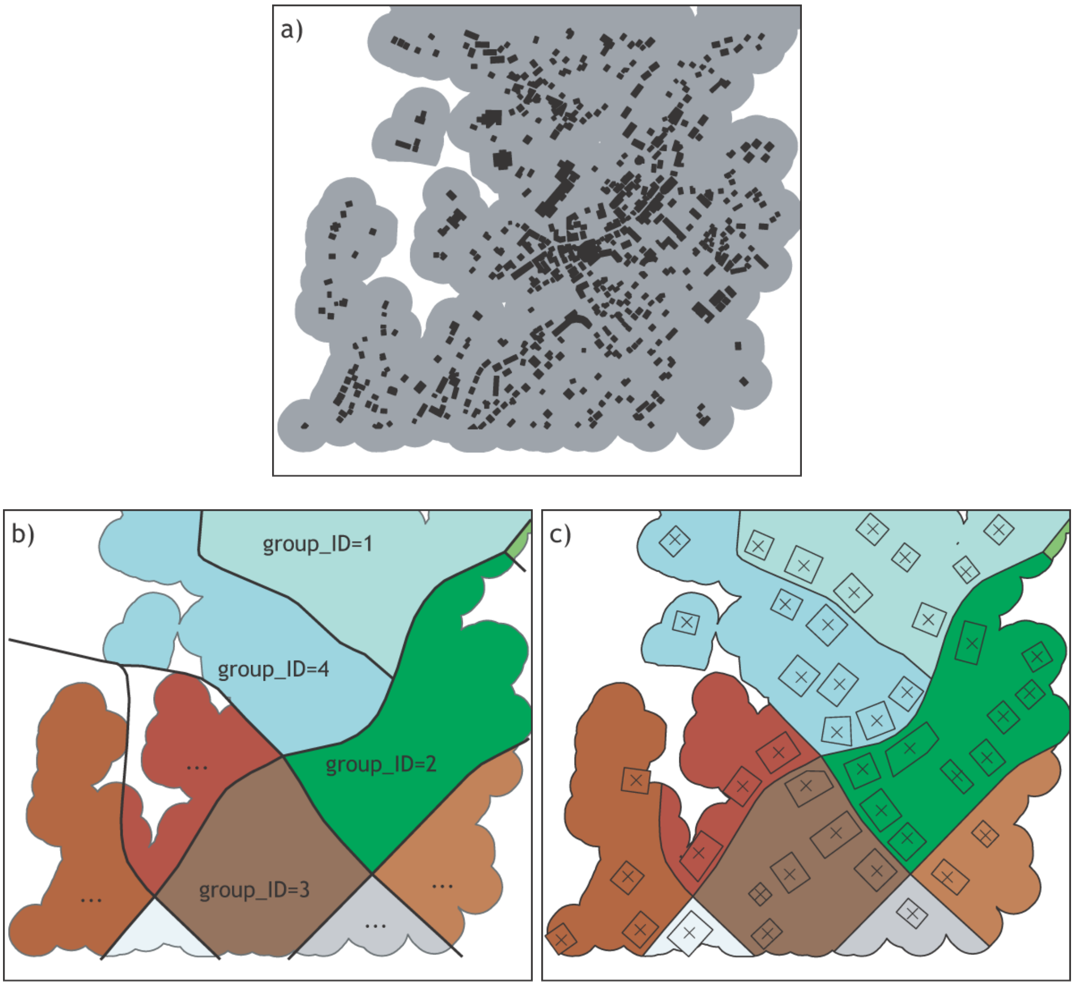


Figure 6.1: Process of the matching process for the feature class *building*. a) The representation of a section of building objects (LOD_{25}) with the computed buffer polygon, b) partition of this polygon by the road network of LOD_{200}, c) assigning each building object of LOD_{200} to one partition. Data: VECTOR25/200 © swisstopo (BA034957).

The best range for the distance d can in this case (relating to our test area) be defined as follows: $60 < d < 140$. By means of Figure 6.2 the evolution of the number of objects with respect to the number of groups is illustrated for the distance $d = \{80, 120\}$.

Both curves (for $d = 80$ and $d = 120$) demonstrate that many groups consist of only a few number of objects. For $d = 80$ the number of groups composed of only one object is 129 ($d = 120$: 69), composed of two building objects it amounts to 64 (30). On the other hand only a few groups include a large number of objects. For $d = 80$ the number of groups with more than 200 objects is 8 (10). The curves illustrate that the assumption to define d between 60 and 140 is reasonable. For optimizing the mesh simplification process a large number of groups with a small amount of objects is needed, for reasons of performance. For the given study area and scale, choosing the distance d between 60 and 140, where the best

Distance d (m)	0	60	70	80	90	100	110	120	130	140	150
No of groups BUF	11547	512	421	368	305	262	233	193	167	149	141
No of groups B_{Poly} (after partitioning)				504		419		336			
Groups with more than 3 objects				265		244		212			

Table 6.1: Number of groups depending on the buffer distance d.

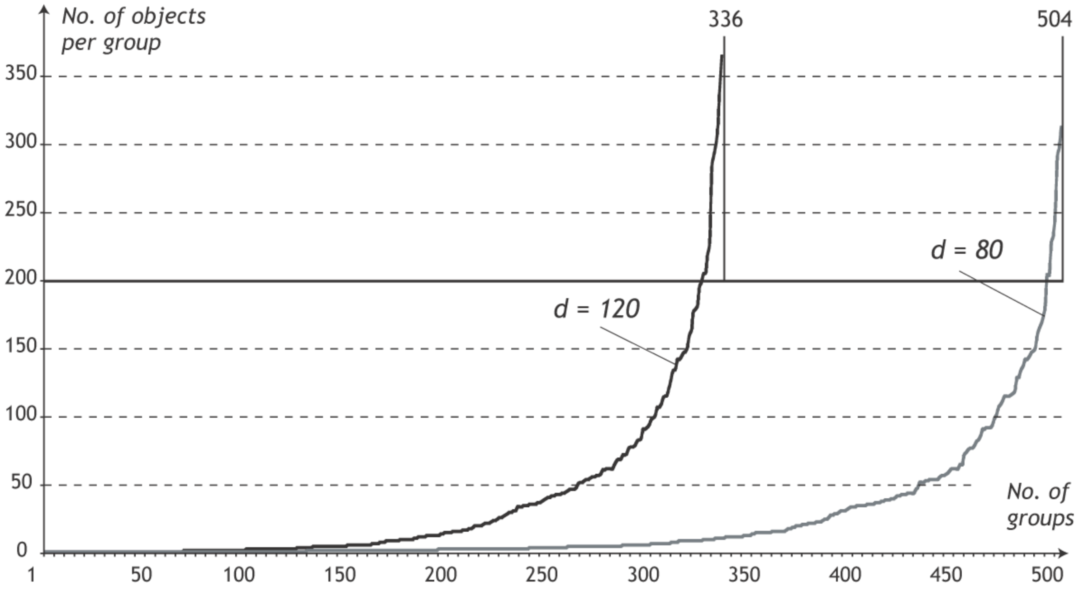


Figure 6.2: The two curves for $d = \{80, 120\}$ illustrate the spread of the amount of building objects in each group with respect to the number of groups formed.

fitted distance is between [100,120], means to obtain the best results for the mesh simplification afterwards. In this project a distance $d = 120$ has been selected to create suitably sized groups for computing the buffer and subsequently the matching process. As shown in table 6.1 the number of groups B_{Poly} amounts to 193 without area partitioning and to 336 after the segmentation through the road network. Interesting is the fact that the amount of groups containing more than 3 building objects is slightly decreasing if the distance d is increased. This shows that the number of groups with more than 3 objects is nearly constant for $60 < d < 140$.

As discussed above the suitable value of d for the selected test area and LOD is within $[60, 140]$. Concerning other areas and LODs (representing other scales) the value d must be determined empirically following the same procedure as explained above. It cannot be defined by a function. The number of objects of the created groups B_{Poly} is determined by the distance d (see table 6.1). Starting from the suitable number of objects in each group for the mesh simplification technique the range for d can be computed. Thereby three requirements must be considered:

- The number of B_{Poly} must be as small as possible;
- The ratio between number of groups with 3 and more objects and B_{Poly} must be as near to 1 as possible;
- The number of objects n in each group should be smaller than 200.

By means of these requirements it is possible to define a suitable range and thus a value for d pertaining to test area and available LOD.

After the definition of d and after partitioning of the area by the road network the matching process for the buildings of LOD_{200} can be accomplished. The cardinality is now [1:n], $\forall n$: each object in LOD_{200} can be assigned to exactly one group B_{Poly} or, in other words, each group can be linked to n objects of LOD_{200} ($n \in \mathbb{N}_0$). In the test area all buildings can be linked to one group, whereby the number of groups concerned amounts to 166. For all other groups no counterpart is available in the other LOD_{200}. The number of objects of a group is between 1 and 29 buildings, whereby around 44 groups are made up of only 1 (similar to the distribution of the objects in B_{Poly}). In conclusion it can be said that the proposed method is robust and suitable for the purposes of matching the feature class *building*.

6.2.3 Enrichment of the Database

As mentioned in the sections above the enrichment of the database with generalization-specific information helps reduce the computation time and thus the 'costs' of the map generation process. In section 4.2 various possibilities of enhancing the database have been proposed. In this section the ones included in the original data set are presented for the corresponding feature class and for the relevant generalization operator. All values have been computed in advance by GIS (ArcGIS[4], LAMPS2) and included in the data set before importing it in the database.

Feature Class *Road Network*

For the feature class *road network* three attributes have been added to the original data set: `group_ID`, `section` and `roundabout`. The relevant generalization operator is mentioned in parentheses.

- `group_ID` (selection): The attribute `group_ID` can be found in all feature classes. It represents the information of the matching/linking process. Corresponding road objects in different LOD get the same value.
- `section` (selection): For a finer-grained subdivision of the road network, each road is cut into road sections (at maximum between two junctions). The sections are numbered consecutively starting form 1 to n (n = number of segments). Corresponding roads (with the same `group_ID`) have the same classification.

---

[4] ArcGIS is a family of software products that form a complete GIS built on industry standards developed by Environmental Systems Research Institute, Inc. (ESRI).

- `roundabout` (selection): Especially for the feature class *road network* the attribute `roundabout` is needed. Since roundabouts are represented in different ways depending on the scale (as linear feature or as vertex) a corresponding attribute must be defined.

Feature Class *Building*

In contrast to the feature class *road network* where the added attribute were more in combination with the matching process (between the different LODs) the class *building* is enriched with more information concerning the individual object, namely: `group_ID`, `island`, `orientation` and `center of gravity`.

- `group_ID` (selection): For this feature class the value of the attribute `group_ID` does not describe the same object in both LODs, but the group it belongs to in LOD_{25} and LOD_{200}.

- `island` (mesh[5]): Building objects with a court yard (especially in the representation of LOD_{25}) are assigned a special value for this attribute.

- `orientation` (mesh): The orientation of a building describes the angle between the vertical axis and the longest side of the building (see section 4.2.2).

- `center_of_gravity` (mesh): The center of gravity is defined by a x- and a y-coordinate and is used for describing the location of a building object by a vertex.

Feature Class *River Network*

Similarly to the *road network* the feature class *river network* is enhanced by the line specific attribute (`section`). Additionally, a value defining the hierarchical structure of the network is given (`horton`). In contrast to the information added before as attribute values this one describes a method[6].

- `group_ID` (selection): As in all other classes the `group_ID` defines corresponding rivers in different LOD. If the name of the river is given at both levels the name and the `group_ID` can be used equally.

- `section` (selection, morphing[7]): This attribute numbers the sections of a river continuously from the source to the pour point.

- `horton` (selection): The horton ordering indicates the 'importance' of a branch of the river network as illustrated in section 4.2.2.

- `number_of_vertices` (morphing): The method `number_of_vertices` computes the amount of vertices of a river section. Since the number can change it must permanently be updated during the generalization process.

---

[5] Mesh simplification technique.

[6] The object-oriented approach allows to define methods and to add them to the object class.

[7] Morphing transformation technique.

6.3 Generalization Process for the Feature Class *Building*

This section consists of two parts: First a discussion of the implementation of the mesh simplification technique is made and second an evaluation takes place.

As noticed in section 3.4.1 LOD can be used within a certain scale range defined by the limits of applicability LA_i. Displaying a LOD without executing a generalization process reduces this range notably whereby the limits are more or less accurately definable (by the values $S_{i,l}$ and $S_{i,h}$). Within this region only a scaling process (with the ScaleChanging Module) is carried out. The range of this scale band $[S_{i,l},S_{i,h}]$ depends on the scale of the LOD and the purpose of the map (scenario). On the basis of this assumption the mesh simplification process is only applied if the requested scale RS_U is determined between the scales $S_{i,h}$ and $S_{(i+1),l}$ $(= LA_{(i+1)})$.

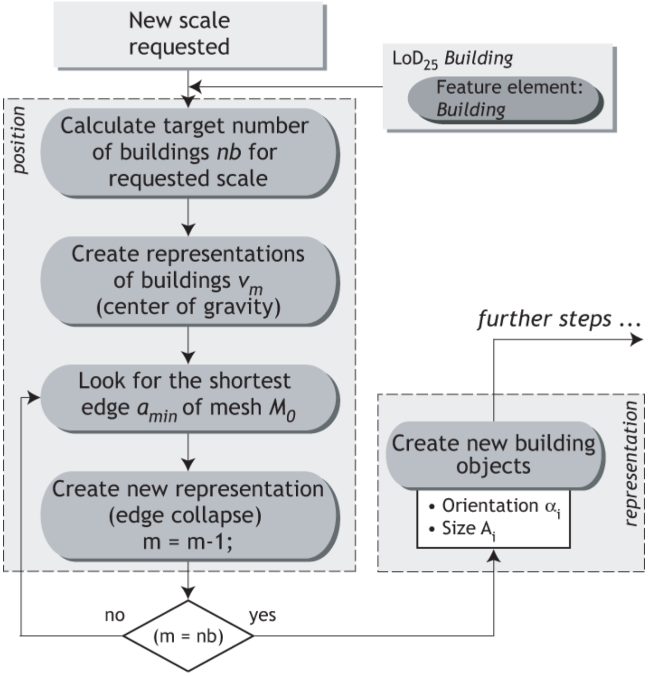


Figure 6.3: Flowchart for the typification operator of the generalization process for the feature class *building*, carried out by the mesh simplification technique. The highlighted part on the left computes the position of the new placeholders, while the right part defines the representation.

Before discussing the mesh simplification technique in more detail a flowchart, distinguishing the two steps discussed in section 5.3.2 *position* and *representation*, is illustrated in Figure 6.3. On the left side of the figure the flowchart of the *position* and on the right side the *representation* step is shown. The figure gives an overview of the sequence of the algorithm and is discussed in the next section.

6.3.1 Mesh Simplification Technique

The main phases of the mesh simplification technique for the feature class *building* are illustrated in Figure 6.4. It explains the first step of the typification process (position).

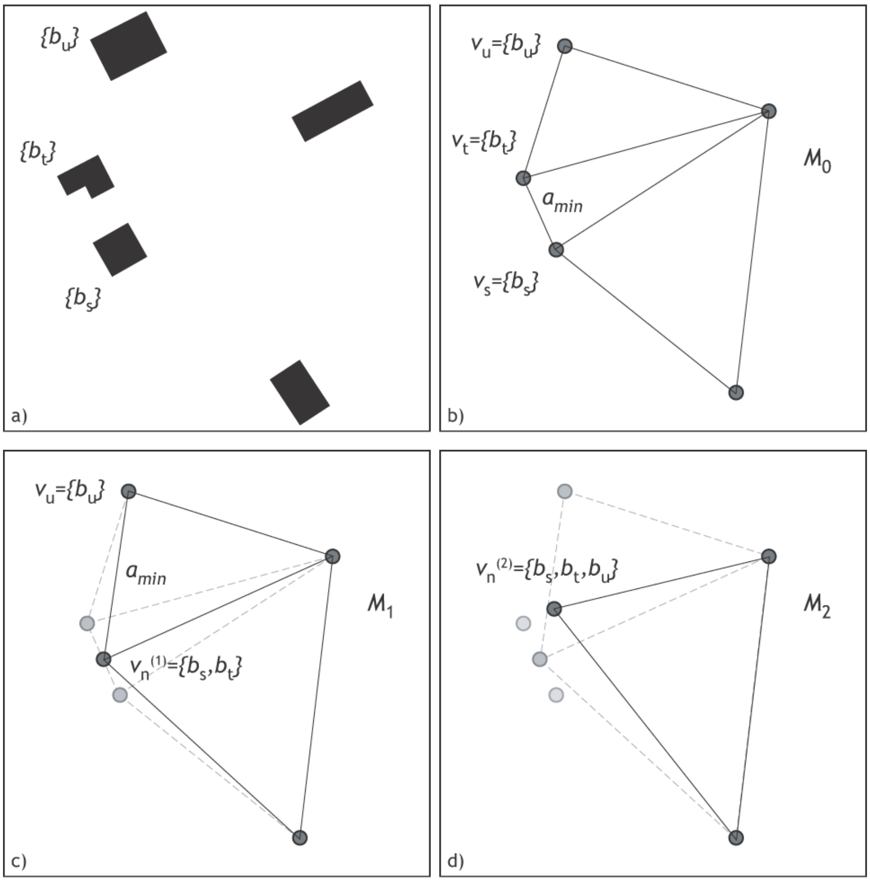


Figure 6.4: Phases of the mesh simplification technique for the step *position* (for a detailed description see text).

Let X be a set of points (with m_{25} vertices $\{v_1,...,v_m\}$) representing the centers of gravity of the building objects $\{b_1,...,b_m\}$ in LOD$_{25}$ with the same group_ID and M_0 a mesh over this point set. In a first iteration loop a new mesh M_1 is sought, where two vertices v_s and v_t are replaced by a new one $v_n^{(1)}$ as illustrated in Figure 6.4 c). The criterion for the selection of these two vertices is that they describe the shortest edge a_{min} of the mesh M_0. The new vertex $v_n^{(1)}$ is defined as center of v_s and v_t. By searching the shortest edge and replacing v_s and v_t through $v_n^{(1)}$ the modifications take place locally and thus the main characteristics of the mesh will not be disturbed. For the following iteration step the number of vertices of X decreases by one as illustrated in Figure 6.4 c). The vertex $v_n^{(1)}$ in this cycle represents the vertices v_s and v_t and thus the building objects b_s and b_t. The same edge collapsing step is carried out with the remaining vertices ($m_{cur} = m_{25} - 1$). The iteration is continued until the current number of vertices m_{cur} is equal to the value $nb(m_{25}, RS_U)$.

The value $nb(m_{25}, RS_U)$ is the number of building objects appearing in the final map with scale RS_U. It depends on the original number of vertices (buildings) m_{25} in LOD_{25} (with the same value for group_ID) and the requested map scale RS_U. To compute $nb(m_{25}, RS_U)$ various terms must be explained and calculated first. In our case we define $Sc(LOD)_0 = \frac{25'000}{1000}$ and $Sc(LOD)_2 = \frac{200'000}{1000}$ as described in Figure 3.4. We set

$$A_{25} = \sum_{j=1}^{m_{25}} ABu_j$$

where A_{25} is the sum of the areas of the building objects ABu_j in LOD_{25}. Out of this area A_{25} a temporary (not adjusted to the final map scale) target number of buildings

$$nb_{tmp} = \frac{A_{25}}{\frac{A_{25}}{m_{25}} * \frac{RS_U}{Sc(LOD)_0}} = \frac{Sc(LOD)_0 * m_{25}}{RS_U}$$

is computed. The factor $\frac{RS_U}{Sc(LOD)_0}$ gives the ratio between the scale selected by the user and the scale of the largest LOD (whereby the scales are always divided by 1000, for example for the requested scale 1:50'000 $RS_U = 50$). If the user asks a map with scale 25'000 ($RS_U = 25$) a factor of 1 results. Tests with nb_{tmp} as target number of buildings have shown that in most cases the number of building objects in the final map scale is too high regarding the available space. To adjust this amount an additional ratio $r_{200/25}$ must be computed to obtain better results. Since we are using a MSDB with two LODs a comparison of the amount of objects m_{25} in LOD_{25} and m_{200} in LOD_{200} (with same value for group_ID) can be done.

$$r_{200/25} = \frac{m_{200} * \frac{Sc(LOD)_2}{Sc(LOD)_0}}{m_{25}}.$$

By means of $r_{200/25}$ several threshold values can be computed to best fit the amount of objects to target scale requested by the user. For each group such a ratio $r_{200/25}$ can be computed. A chart where $r_{200/25}$ is arranged according to its value is shown in Figure 6.5 ($0.364 \leq r_{200/25} \leq 1.037$).

The optimal value for $r_{200/25}$ would be 1.0, which means that the number of objects in LOD_{25} and the number in LOD_{200} extrapolated to scale 1:25'000 is equivalent. Within a range σ (cf. Figure 6.5) no correction factor must be applied (case 1). As shown in Figure 6.5 in most cases the ratio is below this range $[1.0\text{-}\sigma, 1.0{+}\sigma]$ which means that an adapted correction must be applied. For different ranges of $r_{200/25}$ an adequate factor f_{cor} must be determined, where

$$f_{cor} = 1.0 - \psi * \frac{RS_U - Sc(LOD)_0}{Sc(LOD)_2 - Sc(LOD)_0}.$$

Analyzing several results of tests and experiments in the test area three further ranges for $r_{200/25}$ with corresponding term ψ have been defined and listed in Table 6.2.

Finally, for the adjusted number of buildings nb at the requested target scale RS_U two cases must be distinguished:

Case 1: if $(r_{200/25} \geq 0.75)$ $nb = nb_{tmp};$

Case 2: if $(r_{200/25} < 0.75)$ $nb = nb_{tmp} * f_{cor}$.

If the number of building objects to appear in the final map is known the iteration process of edge collapsing (replacement of vertices) can proceed until that number is reached

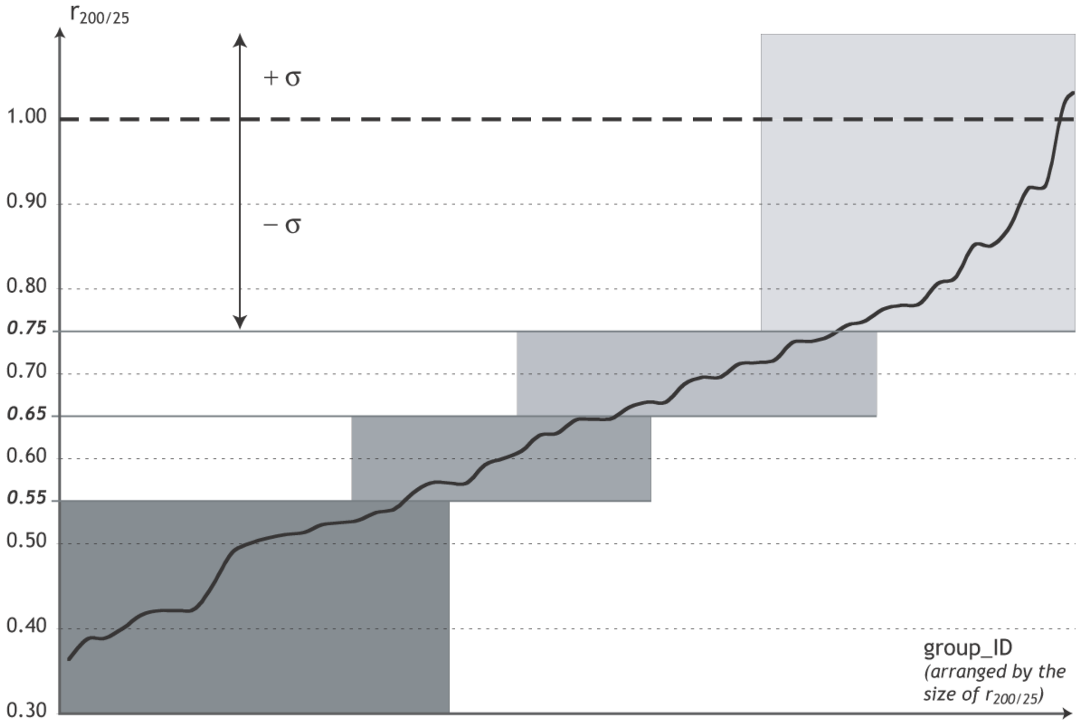


Figure 6.5: Chart of the ratio $r_{200/25}$ arranged by its size. The gray boxes indicate the ranges of $r_{200/25}$ listed in table 6.2.

$r_{200/25} \leq 0.55$	$\psi = 0.7$
$0.55 < r_{200/25} \leq 0.65$	$\psi = 0.6$
$0.65 < r_{200/25} \leq 0.75$	$\psi = 0.5$
$0.75 < r_{200/25}$	no correction value

Table 6.2: Ranges of $r_{200/25}$ acquired from analyzing several tests and results.

$(m_{curr} = nb)$. Examples and results of the approach are illustrated and discussed in section 6.3.2.

Since our approach is based on looking for the vertices describing the shortest distance (shortest triangle edge), areas with high building density would be thinned out more strongly than sparsely populated regions. On the one hand this could be desirable to avoid too many objects in urban areas, but on the other hand it would distort the characteristics of an area. To meet this requirement a correction factor f_a is included in the calculation of the distance. This factor f_a allows to elongate the real edge length a between the vertices and thus to decrease the thinning process in densely populated areas. It depends on the number of objects (n_{rep})

that a vertex represents[8] as well as the term s_u which must be set by the user in advance meeting the following conditions:

$$0 \leq s_u \leq 1 \quad \longrightarrow \quad \begin{array}{rl} s_u = 0.0: & \text{no correction is done;} \\ s_u = 1.0: & \text{full correction by means of} \\ & \text{the original number of} \\ & \text{represented objects.} \end{array}$$

From these two values the factor f_a can be defined as follows

$$f_a = s_u * (n_{rep} - 1) + 1.$$

As described above the shortest edge a_{min} of all edges a_i ($\forall i$) of the mesh M_0 is looked for:

$$a_{min} = min[a_i], \qquad \text{where} \qquad a_i = f_a * \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2}.$$

Figure 6.4 shows the concept of the technique. The shortest edge a_{min} is between vertex v_s and v_t. By collapsing this edge a new vertex $v_n^{(1)}$ is created, which represents the two vertices v_s and v_t of the edge (these vertices describe the center of gravity of the building objects b_s and b_t). In next iteration again the shortest edge a_{min} of the new mesh M_1 is looked for.

As displayed in Figure 6.4 c) the vertices $v_n^{(1)}$ and v_u determine a_{min}. For the calculation of the position of the new vertex $v_n^{(2)}$ all affected vertices (and thus buildings) must be considered: v_s, v_t and v_u. Hence, the position of $v_n^{(2)}$ is fixed by the center of gravity of the vertices which are replaced (show in Figure 6.4 d)). This iteration cycle proceeds until the number of remaining vertices is smaller than the target number of buildings nb. After termination this iteration process each remaining vertex represents one or more of the original buildings.

The next step of the typification process (*representation*) creates for each vertex a building object from the ones it represents. For example, for the vertex $v_n^{(2)}$ the vertices v_s, v_t and v_u determine the shape of the new building object $b_n^{(2)}$.

Thereby the two attributes

- `area` A[`width`,`length`], and
- `orientation` α (as defined in section 4.2.2)

of each represented object are considered. The selected approach is based on the idea that the newly created building object $b_n^{(2)}$ should on the one hand best represent the largest object of the group and on the other hand also attempt to maintain the characteristics of the whole group.

`Area` A

Comparing the map series of the Swiss National Mapping Agency it can be stated that most buildings are strongly simplified at a scale of 1:100’000 and smaller and thus less detailed. The shapes are usually represented as rectangles to meet the minimum separability distance as defined in Spiess (1990). In the context of on-demand web mapping where the minimum separability distances are more severe (owing to the coarse display resolution) the depiction of buildings should be kept very simple. The concept here is also to define each new object as rectangle, whereby the area A_n is computed out of the average of the pertaining buildings A_k

---

[8] For example, in Figure 6.4, the value for $v_n^{(2)}$ is $n_{rep} = 3$.

(in LOD_{25}), extrapolated to the requested scale RS_U:

$$A_n = \frac{\sum_{k=1}^{nb} A_k}{nb} * \left(\frac{RS_U}{Sc(LOD)_0}\right)^2.$$

Out of this value A_n the `width` and `length` of the new object can be calculated. The ratio $\frac{width}{length}$ of the new building must be the same as for the largest represented object (in LOD_{25}).

As mentioned in section 6.3.2 the computed area A_n for the created object is in most cases too large for the requested scale because it is based on the largest object of the group. To avoid this effect and adjust the objects to a reasonable size a scale-dependent adjustment factor f_{area} must be used. This value which has been defined merely empirically can be defined as follows[9]:

$$f_{area} = 1.0 - \phi * \frac{RS_U - Sc(LOD)_0}{Sc(LOD)_2 - Sc(LOD)_0}.$$

However, more research work must be carried out to define possibilities to obtain optimally sized objects considering the corresponding minimum separability distances defined by Spiess (1990).

`Orientation` α

For the orientation α of the new building only the value of the largest object of the group is considered. The reason for this approach is that the orientation of the largest building can be assumed to be most representative for the represented objects and influence or even dominate the characteristics of its environment. Hence, the orientation α of the new and the largest object are equal.

6.3.2 Evaluation

In order to test the *mesh simplification technique* different groups of buildings in the test area have been selected. The selection was based on the two criteria, number of objects and spatial arrangement of the objects within the group, representing different cases:

group_ID	**77**	**125**	**145**	**279**	**405**	**497**
m_{25}	41	55	94	76	79	57
m_{200}	4	7	6	7	5	2

The first row defines the identification number of the group (group_ID) while m_{25} and m_{200} indicate the number of objects in the original representation of LOD_{25} and LOD_{200}. Note that the ratio $\frac{m_{25}}{m_{200}}$ varies between 4 and 30 (concerning all groups within the test area). It can be stated that there is no dependence between the number of objects in a group and the ratio $\frac{m_{25}}{m_{200}}$.

The following evaluation consists of two main parts: 1) a discussion of the computed number of objects nb depending on the selected map scale RS_U for the mentioned groups; 2) the illustration and discussion of the results of the created building objects.

Position of New Buildings

As explained in section 6.3.1 the factor s_u influences the distances a of the edges (between the vertices). For the mentioned groups with $ID = \{77, 124, 145, 279, 405, 497\}$ Table 6.3

---

[9] In our case the value $\phi = 0.4$ has been calibrated empirically for the used map series.

illustrates the number nb of created objects for different scales. The values for the scales m_{25} and m_{200} are the original ones of LOD_{25} and LOD_{200} while the other ones are computed with the mesh simplification technique to achieve the transition between the two extreme scales.

	m_{25}	*50k*	*75k*	*100k*	*150k*	*200k*	m_{200}
77	41	*20*	*13*	*10*	*6*	*5*	4
125	55	*27*	*18*	*13*	*9*	*6*	7
145	94	*42*	*25*	*16*	*7*	*3*	6
279	76	*35*	*21*	*14*	*8*	*4*	7
405	79	*35*	*21*	*13*	*6*	*3*	5
497	57	*25*	*15*	*9*	*4*	*2*	2

Table 6.3: Number of objects in each group computed for the scales 1:50'000, 1:75'000, 1:100'000, 1:150'000, 1:2000'000 with the mesh simplification technique.

Table 6.3 illustrates that the computed number of objects for each group depends on both values m_{25} and m_{200}. For example, groups 125 and 497, respectively can be compared. Both start with nearly the same value for m_{25} (55 vs. 57) but end with a different number of objects for m_{200} (7 vs. 2). The computed values intermediate are in relation with the extreme values and thus not equal for the same scale (100k: 13 and 9, respectively). For the second group the reduction is stronger than for the first one as expected in comparison to the extreme values. Another comparison can be done between the computed and the original values of scale 1:200'000. Comparing the computed $200k$ and the original m_{200} values a difference can be detected. For groups with a small amount of objects the difference is slight (group_ID=77: (5 vs. 4)). In contrast for large groups the difference may be up to 50% (group_ID=145). However, as the majority of groups amounts to less than 100 objects (see therefore Figure 6.2: 302 of 336 groups contain less then 100 buildings) the method uses a good approximation concerning the number of objects for a specific scale. Evaluating all the values of this table two things must kept in mind. On the one hand the number of objects of a group in LOD_{200} is extremely small compared to the number in LOD_{25}. Hence, the relative error between computed and original value can vary enormously, although the absolute changes may be small. On the other hand LOD_{25} represents most buildings with a cardinality of [1:1] related to reality (almost every building exists in reality). LOD_{200} is a strongly generalized version of the reality, where the amount of buildings represented by one object is strongly dependent of the producer's/cartographer's view. Thus, the significant divergence of 50% between 200k (computed) and m_{200} (for example group_ID=145) have to be read with some caution.

Besides the amount the positions $[x, y]$ of the created objects in cartesian coordinates play an important role. The new placeholders must best portray the original nature of each group. In the following several figures illustrate the results computed with the mesh simplification technique. Due to lack of space not all groups are discussed but examples are shown to demonstrate the quality of the proposed method. First the original and created placeholders of group_ID=77 should illustrate the obtained results.

Figure 6.6 a) shows the building objects of LOD_{25} (VECTOR25) with $nb = 41$ elements. In Figure 6.6 b) only the centers of gravity representing the original data set are displayed. These points compose the vertices of the mesh M_0 as discussed in the previous section and define the starting point of the mesh simplification technique. With the iterative process of *edge collapsing* a number of vertices are removed or replaced by new ones. Figure 6.6 c) shows the result for the requested scale of 1:50'000 displayed as black crosses whereby the

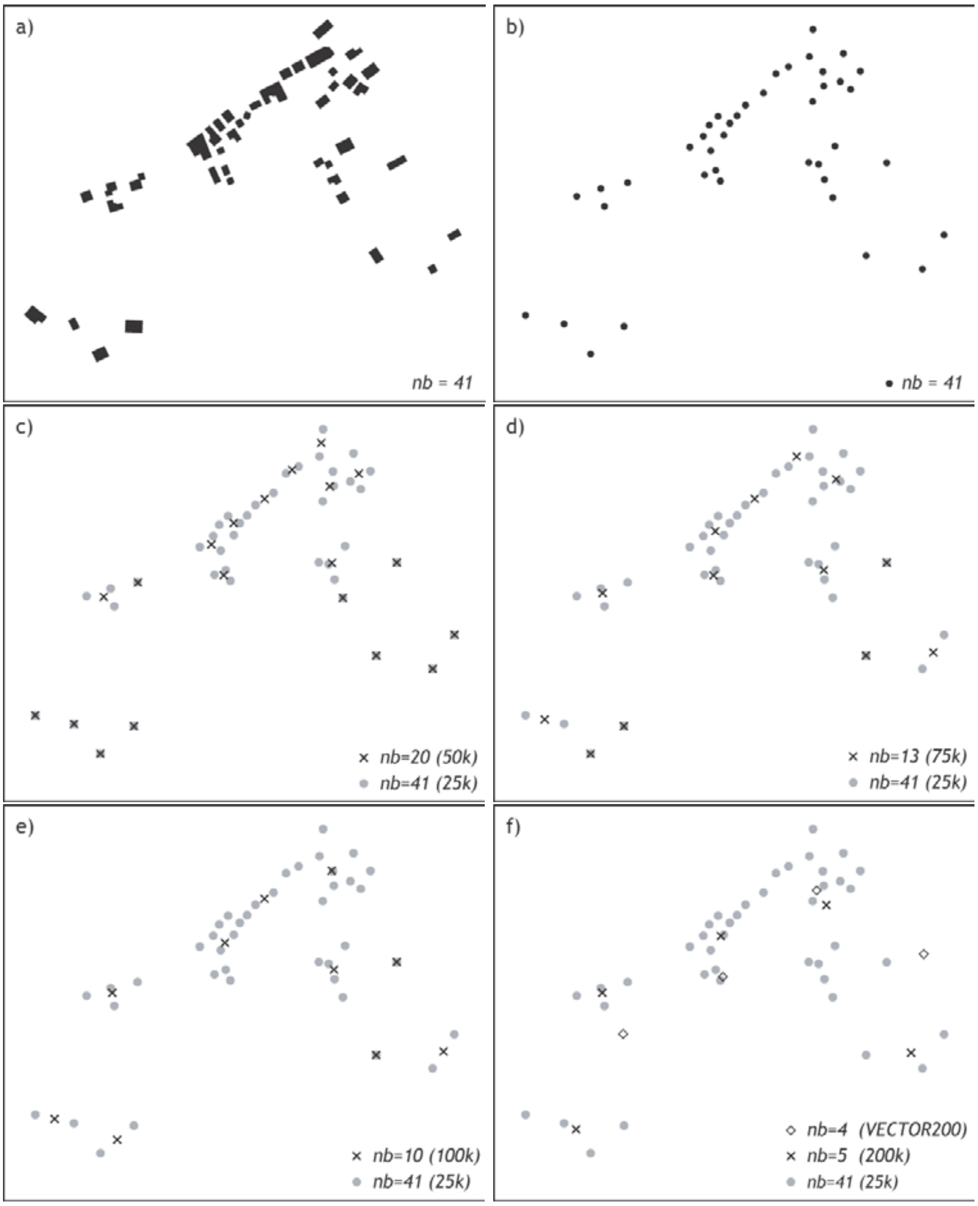


Figure 6.6: The positions of the placeholders for the following scales: a) buildings in LOD_{25}, b) 1:25'000 (original data set), c) 1:50'000, d) 1:75'000, e) 1:100'000 and f) 1:200'000 with the objects' center of gravity in LOD_{200} (all not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

amount is decreased to $nb = 20$. In the background the vertices depicted in gray describe the vertices of M_0. The placeholders for scale 1:75'000 are displayed in Figure 6.6 d) where the number decreases to $nb = 13$. In Figure 6.6 e) the mesh is composed of only $nb = 10$ vertices. Finally, in Figure 6.6 f) a comparison with the positions of the vertices of LOD_{200}

can be done. In contrast to LOD_{200} the computed solution results in 5 objects, whereby three of them are describing a similar position. Evaluating the created placeholders for each scale with the original data set of LOD_{25} it can be noticed that reasonable results have been achieved maintaining the main characteristics of the group through all scales. The same can be observed in the following two examples.

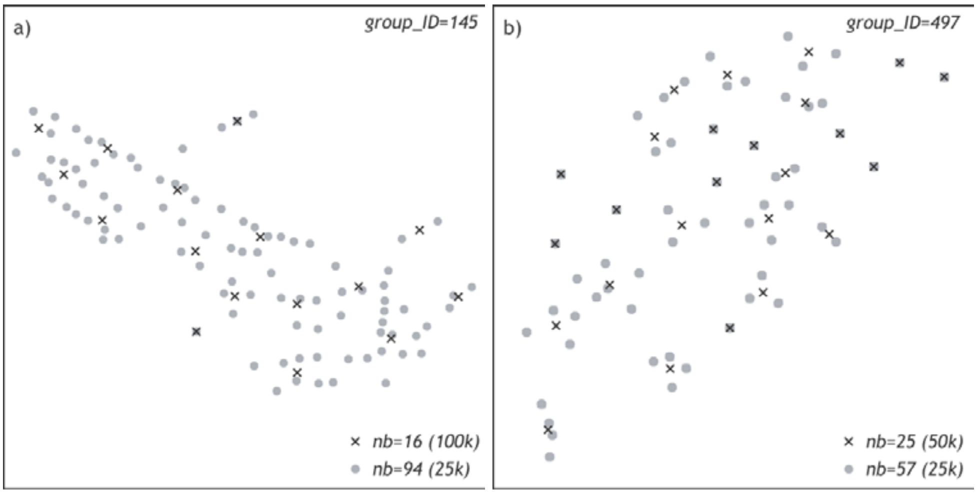


Figure 6.7: The positions of the placeholders for the groups a) group_ID=145 and b) group_ID=497 (both not to scale). Data: VECTOR25 © swisstopo (BA034957).

Figure 6.7 a) illustrates the result for group_ID=145 with a large amount of objects ($nb = 94$) for the scale of 1:100'000 ($nb = 16$). Similarly as for the previous example for this distribution well-located placeholders are found with respect to the original density variation. Another arrangement is illustrated in Figure 6.7 b) for group_ID=497. Out of $nb = 57$ vertices representing buildings in a densely populated area it results $nb = 25$ objects for the scale of 1:50'000.

All the examples discussed so far have been computed with the factor $s_u = 0.0$ (see section 6.3.1) which implies that no correction has been done concerning the distance between the vertices. To incorporate this quality the value s_u has been introduced to decrease the thinning process in densely populated areas[10]. The difference between computing a representation with $s_u = 0.0$ or $s_u = 1.0$, respectively is displayed in Figures 6.8 a) and b) for group_ID=279 and scale 1:75'000 (s_u may range between 0.0 (no correction) and 1.0 (maximum correction)).

For both results the number of objects to be displayed for the requested scale is equal ($nb = 21$). What changes are the positions of the centers of gravity of the placeholders. On the left side (Figure 6.8 a)) the highlighted area α is represented by only two buildings while on the right part of the figure (b)) four objects are displayed within the same area. Comparing these two areas reflects the effect of the factor s_u: computing the result with a high value for s_u ($s_u \longrightarrow 1.0$) takes the density of the building objects more into account than with low values ($s_u \longrightarrow 0.0$). The same effect can be found in the second highlighted example β where places with a high density of buildings are represented better in b) than in a). Working with low values for s_u ($s_u \longrightarrow 0.0$) prefer single seated objects as illustrated in the highlighted area γ.

---

[10] Densely populated areas are defined as regions where the density of building objects is higher-than-average.

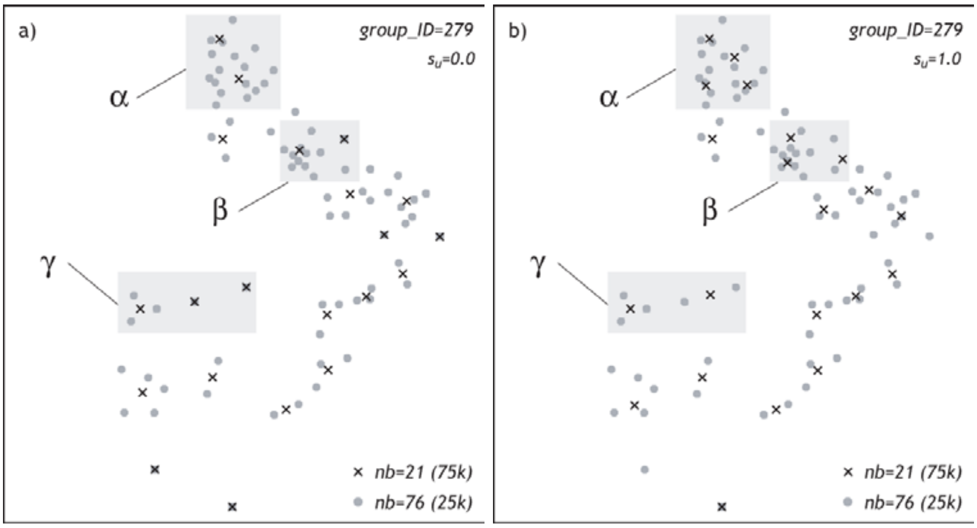


Figure 6.8: The positions of the placeholders for group_ID=279 with a) $s_u = 0.0$ and b) $s_u = 1.0$ (both not to scale). Data: VECTOR25 © swisstopo (BA034957).

Representation of New Buildings

As seen in the previous part of this section the positions of the placeholders are well-placed maintaining the characteristics of the groups of the original data set. Besides the position the size and dimension of the new buildings are of crucial importance. The methods for computing the two main attributes area and orientation of this new objects are explained in section 6.3.1. In this part some examples should be illustrated and discussed.

Figure 6.9 displays the result of group_ID=77 computed for different scales whereby the examples b) - f) have been scaled to the same size as a).

Figure 6.9 a) shows the original data set of LOD_{25} for group_ID=77 with $nb = 41$ objects. For scale 1:50’000 (Figure 6.9 b)) the amount of objects decreases to $nb = 20$. The dimension [`length`,`width`] of each object depends on the dimensions of the represented objects, while the orientation α is fixed by the same orientation as the largest building. In Figure 6.9 c) $nb = 13$ objects are represented for scale 1:75’000. As can be seen overlap problems arise in the lower part of the figure. As one possible solution the elimination of one object can be taken into consideration. Figure 6.9 d) shows $nb = 10$ objects for scale 1:100’000. The last two Figures 6.9 e) and f) display the buildings objects for scale 1:200’000: e) the computed objects and f) the data set of LOD_{200}. Comparing the size of the buildings at scale 1:200’000 a similarity can be assessed. Note that for all examples shown the constraints of minimal distance between individual objects has not been taken in account. All illustrations are depicted at 1:25’000. Figure 6.10 shows the same situation, but now at the corresponding target scale.

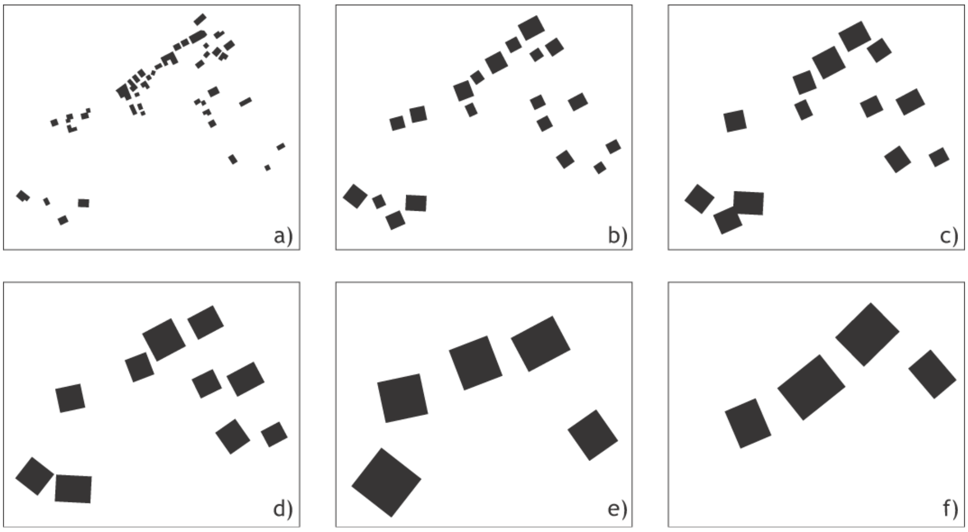


Figure 6.9: The positions and dimensions of the placeholders of group_ID=77 for different scales: a) 1:25'000 (representing LOD_{25}), b) 1:50'000, c) 1:75'000, d) 1:100'000, e) 1:200'000 and f) 1:200'000 (representing LOD_{200}) (all at scale 1:25'000). Data: VECTOR25/200 © swisstopo (BA034957).

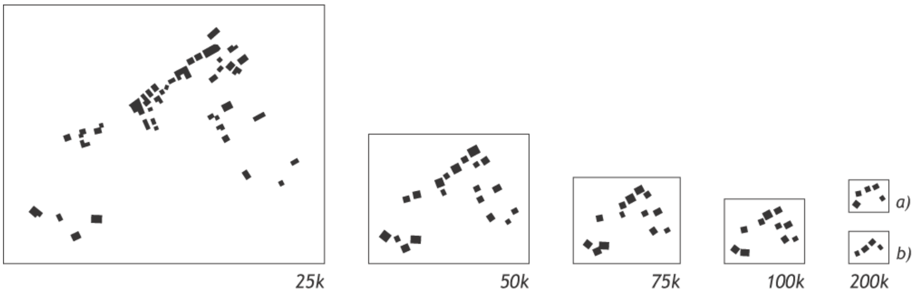


Figure 6.10: The buildings of group_ID=77 computed for different scales with correct proportions. For scale 1:200'000 the computed (a)) and the original (b)) representations are shown. Data: VECTOR25/200 © swisstopo (BA034957).

6.3.3 Limitations of the Methods

For both parts of the approach limitations can be found. They may be summarized in the following four points:

- In the present implementation the mesh simplification technique fails to maintain alignments of buildings. Such particular patterns could be preserved by introducing additional constraints to the method. Constraints could be generated by an off-line preprocessing step that detects buildings alignments (for instance, using the method proposed by

Christophe and Ruas (2002)) and observed in the algorithm by setting increased weights for buildings in alignment structures.

- The positions of the placeholders only depends of the center of gravity of the objects it represents. In some cases, especially when representing large or important buildings, a combined geometric and semantic-based edge collapsing could be of advantage to keep these special building objects. The selected approach of mesh simplification must be enhanced by the possibility of using additional parameters (e.g. importance of the building objects).
- The computed area of a placeholder depends on the areas of all represented buildings and is derived from the average of the area of all included objects. Even if a placeholder represents buildings with extremely different sizes the average size is displayed. This can influence the local nature of these few objects. A similar problem arises for example if one large-sized building and several small ones are used to compute the area of a placeholder. In this case the size of the new object will be too small compared to the large building. This situation leads to an unfortunate depiction of the placeholder. To better determine the size of a placeholder a statistical evaluation of the area of the represented building objects can be accomplished. Figure 6.11 shows the number of building objects in each class of the attribute area A. By means of this spreading the size of the placeholder can be determined.

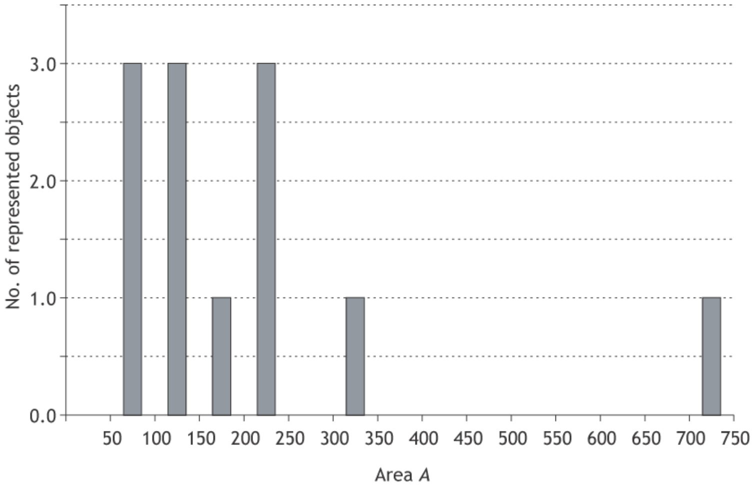


Figure 6.11: Histogram for a placeholder of an object of group_ID=77 illustrating the number of objects in each class (area A). In most cases the area is smaller than 250 m^2 (10 of 12 objects). By means of this spreading the size of the placeholder can be determined.

- The minimum separability distances are taken into account only indirectly by the two given states LOD_{25} and LOD_{200}. For the computed placeholders this constraint is not respected in the current implementation. A solution in context with on-demand web mapping could be to downsize the concerned building objects and thus gain space between them in order to avoid overlaps and congestion. An example of overlaps created by excessive building sizes is given in Figure 6.12 which illustrates the buildings of group_ID=497.

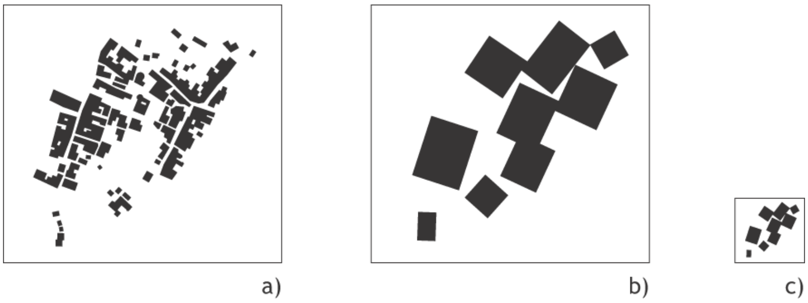


Figure 6.12: Example for computed objects that are disproportionate (group_ID=497) for the requested scale. Figure a) LOD_{25} with 57 objects, b) result computed for scale 1:100'000 (at scale 1:25'000), and c) computed result at scale 1:100'000 (9 objects). Data: VECTOR25 © swisstopo (BA034957).

> Figure 6.12 a) shows the original data set (LOD_{25}) of group_ID=497 for the feature class *building*. Figure 6.12 b) and c) display the computed representation with the mesh simplification technique for the scale 1:100'000. Besides the overlap problem the size of the objects is not appropriate for the requested scale. The problem is that very large buildings in LOD_{25} define the basis for computation of the building sizes for the placeholders at 1:100'000. For that reason some objects are about eight times the size required by the constraints defining the minimum perceptual limits (Spiess 1990). In this case the size of the objects can be reduced to meet the requirements mentioned. A comparison can be done with Figure 6.9 d), where the dimensions of the objects are well proportioned.

With the possibility of defining the correction factor s_u in the range of [0.0,1.0] the user can influence the kind of map he/she wants to have. The differences between the representations concerning the positions of the building objects for ($s_u \longrightarrow 0.0$) or ($s_u \longrightarrow 1.0$) are significant.

The advantage of using this method of mesh simplification is that any scale can be generated out of the two border data sets LOD_{25} and LOD_{200}.

6.4 Generalization Process for the Feature Class *River Network*

The implementation of parts of the generalization process for the feature class *river network* as proposed in section 5.4.1 represents the main portion of this section. Two topics are presented: a) an implementation of the morphing transformation technique, and b) an evaluation and discussion of this method. By means of some examples advantages as well as disadvantages of this approach will be illustrated.

As for all feature classes also for the *river network* the basis of the data set is composed by LOD_{25} and LOD_{200}. To best fit the original data for the given requirements of on-demand web mapping a preprocessing takes place. As it is especially tailored to the morphing process it will be presented in section 6.4.1.

As discussed in section 3.4.1 between the limits of scale $[S_{i,l},S_{i,h}]$ just a scaling and no generalization process is used, whereas for applications between the limits $[S_{i,h},LA_{(i+1)}]$ a generalization procedure is required (see Figure 3.4) to create a scale adapted map. This requires different steps, whereby the morphing transformation technique forms a crucial part. A flowchart (Figure 6.13) illustrates the several steps of the process.

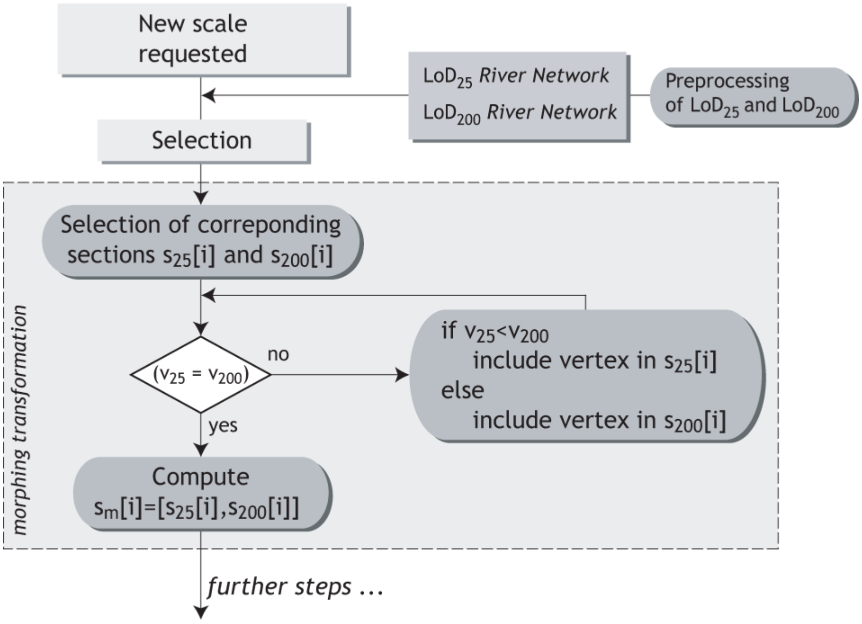


Figure 6.13: Flowchart of the morphing process for the feature class *river network*. The number of vertices of section $s_{25}[i]$ and $s_{200}[i]$ are defined by v_{25} and v_{200}.

This overview should illustrates the implementation of the morphing procedure described in the next section (6.4.2). The process itself is divided into three main parts: i) selection of homologous sections, ii) adding, if necessary further vertices, and iii) computing the morphing transformation.

The morphing transformation always requires the presence of corresponding line sections on both the detailed and the less detailed LOD used. If this prerequisite is not met, morphing cannot be carried out. An alternative is described by a weeding operation accomplished by the Douglas-Peucker algorithm (Douglas and Peucker 1973) which we will not discuss further, however.

6.4.1 Preprocessing the Data Set

As discussed in section 5.4.2 the morphing procedure only works if two different representations of the same object exist, representing the start and end state. This implies the existence of a MSDB with several LODs. Before applying such an approach a preprocessing step must be accomplished to prepare the data. Three kinds of enhancements must be distinguished:

- To improve the data set spurious shapes (small spikes) on the digitized line objects must be removed by a smoothing operation (accomplished by the Gaussian smoothing algorithm). As a side-effect this procedure results in a better (more even spaced) distribution of the vertices along the line.

- Computation of the Horton-ordering and a matching process (group_ID) are used to define homologous sections of the two lines.
- A preprocessing step especially used for morphing is the definition of corresponding sections (as parts of the river line) of the same object in different representations. Figure 6.14 illustrates this step.

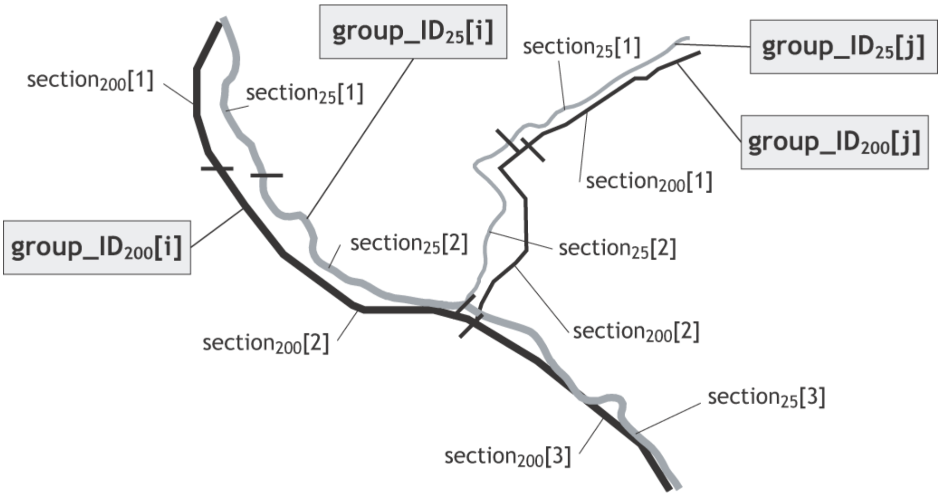


Figure 6.14: Partitioning of the river network into corresponding sections s for each river object. Data: VECTOR25/200 © swisstopo (BA034957).

Figure 6.14 shows two rivers with the identification group_ID=i (heavy lines in gray and black) and group_ID=j (for LOD_{25} and LOD_{200}). To obtain a good result in the morphing process it does not suffice to know the counterpart of the whole river object (determined by the group_ID). Since rivers usually describe very long line objects (such as streams) smaller *sections* must be defined to adequately execute the morphing process. In the illustrated example the river is subdivided into three sections which implies that all representations of this river are assigned the same number of sections. This partitioning of the river is in our case done manually, as step of the preprocessing step.

After completion of this preprocessing step the enriched data set is ready to be utilized in the morphing process.

6.4.2 Morphing Transformation

The principles of the morphing process have been discussed in section 5.4.2 and illustrated in Figure 5.10. In the following discussion of the implementation only the steps in the highlighted area of Figure 6.13 are presented.

As the morphing process is carried out for each section $s[i]$ of a river object R the starting point is given by the selection of the corresponding representations of the river section $s_{25}[i]$ (LOD_{25}) and $s_{200}[i]$ (LOD_{200}) as illustrated in Figure 6.15 a).

These two representations define the 'keyframes' of the river section $s[i]$. Depending on the scale RS_U requested by the user an intermediate frame $s_m[i]$ of the section has to be computed

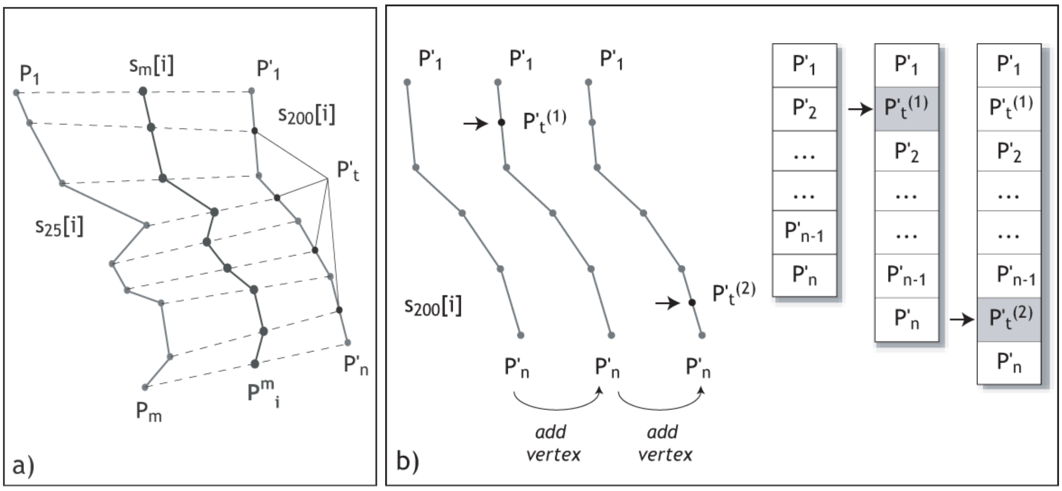


Figure 6.15: a) The principle of the morphing transformation technique. b) Adding vertices in order to be able to carry out the process.

('morphed'). The two keyframes, however, may consist of a different number of vertices P, m for LOD_{25} and n for LOD_{200}. Depending on the morphing method used the numbers of vertices m and n of the two keyframes have to be made equal by vertex insertion (Figure 6.15 b)) before the actual morphing procedure can be carried out. As discussed in section 5.4.2 two morphing methods are possible:

- Allocation of corresponding vertices and distance computation between the vertex sets P and P';
- Triangulation over all vertices P and P' of both representations.

In the following, we will present the implementation of both methods and evaluate their cartographic feasibility on the basis of experiments.

Morphing transformation based on distance computation between corresponding vertices

For the first method the representations of the section $s_{25}[i]$ and $s_{200}[i]$ must be made up of the same number of vertices. Usually the number of missing vertices h between the two representations differs considerably. Depending on the length of the section a factor of 8-12 is possible, whereby obviously $s_{200}[i]$ is made up of a lesser number. To meet the requirements of the morphing method additional vertices $h = m - n$ must be inserted. Therefore, since in almost all cases m is greater than n, h vertices must temporarily be included in $s_{200}[i]$. Figure 6.15 a) shows an example where $m = 9$ and $n = 5$ and thus the number of vertices to be inserted amounts to $h = 4$. The question is now *where* or *how* to insert these auxiliary vertices. In the literature several possibilities are known to solve this problem. For this work the following two rules have been defined to distribute the missing vertices along the line $s_{200}[i]$:

1. The first and the last vertex (P'_1 and P'_n) of the section are fixed and thus all insertions take place between these two vertices.

2. While $n > m$ the longest connection D_m of two consecutive vertices P'_i and P'_{i+1} must be split in half by inserting a new vertex P'_t. Thereby D_m is defined as follows

$$D_m = max|P'_i - P'_{i+1}| \qquad \text{for } i = 1, ..., n.$$

   Figure 6.15 b) illustrates the insertion of one vertex P'_t for each iteration cycle. On the left hand side the insertion on the line object is shown; the right hand side depicts the insertion of the new vertex at the correct position of the array data structure.

As soon as n is equal to m the iteration process will stop. Now both representations are composed of the same number of vertices and thus the requirements for a successful morphing transformation are fulfilled.

For computing a representation $s_m[i]$ for the requested map scale RS_U corresponding vertices in $s_{25}[i]$ and $s_{200}[i]$ are needed. To simplify this process vertices with the same index are connected (vertex matching). Since $n = m$ each vertex in P has a counterpart in P'. For example P_1 corresponds to P'_1 and $P_h \longrightarrow P'_h$. Consequently for each index j a tuple of two matched vertices is defined:

$$\{P_j[x_j|y_j], P'_j[x'_j|y'_j]\} \qquad \text{for } j = 1, ..., m.$$

To determine the position of vertex P^m_j of the morphed section $s_m[i]$ an iteration process must be activated starting from $j = 1$ to m. The position of P^m_j is calculated as follows:

$$P^m_j = P_j + \Delta P, \qquad \text{where } \Delta P = |P_j - P'_j| * s_f. \tag{6.1}$$

For this approach a scaling factor s_f can be selected which may vary between 0.0 and 100.0.

$$0.0 \leq s_f \leq 100.0 \qquad \longrightarrow \qquad \begin{array}{ll} s_f = 0.0: & \text{defines same representation} \\ & \text{as for } \mathsf{LOD}_{min}\ (\mathsf{LOD}_{25}); \\ s_f = 100.0: & \text{defines same representation} \\ & \text{as for } \mathsf{LOD}_{max}\ (\mathsf{LOD}_{200}); \end{array}$$

A value for s_f between 0.0 and 100.0 describes a representation for a scale between 1:25'000 and 1:200'000. Determining the correct scale out of a value for s_f in [0.0,100.0] is not as trivial as one might think. A definition of this scale-dependent value s_f has been established by empirical analysis. For our test area and the given representations LOD_{25} and LOD_{200} the following equation for s_f has been derived:

$$s_f = \left(\frac{RS_u - Sc(LOD)_0}{Sc(LOD)_2 - Sc(LOD)_0} \right)^{0.35} * 100. \tag{6.2}$$

The value *0.35* for the exponent has to be calibrated empirically for each map series. By means of the requested scale RS_U the corresponding value for s_f can be listed:

RS_U:	1:25k	1:50k	1:75k	1:100k	1:150k	1:200k
s_f:	0.0	50.6	64.5	74.3	88.9	100.0

For well-define the correlation between requested scale RS_U and s_f a more systematic analysis must be accomplished.

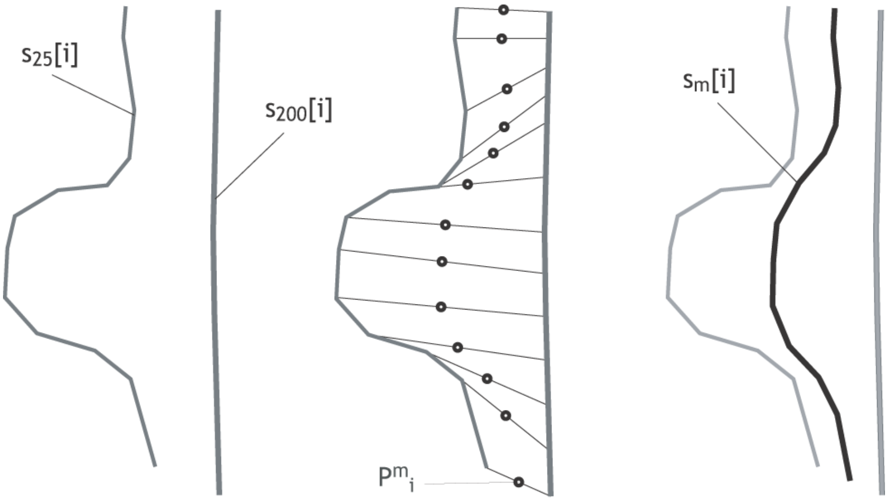


Figure 6.16: Example for a morphing transformation with the method of corresponding vertices ($s_f = 50.0$).

Figure 6.16 illustrates an example for a river section with $s_f = 50.0$. The left part of the figure depicts the two keyframe representations of a hypothetical river section as discussed above. With the presented method a morphed section $s_m[i]$ out of the two 'keyframes' is derived. In the middle the vertices P_i^m are shown for each tuple of corresponding vertices. Connecting the consecutive vertices a new representation $s_m[i]$ of the section for a requested intermediate LOD results (right).

Triangulation-based Morphing Transformation

The second method for the creation of a new representation is to define a triangulation (Delaunay triangulation) with all vertices of the representations $s_{25}[i]$ and $s_{200}[i]$. Based on this triangulation the same process as described above using equation (6.1) can be carried out. The advantage of this approach is that the number of vertices m and n of $s_{25}[i]$ and $s_{200}[i]$, respectively may be different and that no missing vertices need be added. Figure 6.17 illustrates the result of a morphing transformation accomplished with this method.

To compare the result with the outcome of the previous method the same section has been selected. The left hand side of the figure shows the two representations of the selected section $s[i]$. Defining a triangulation over all vertices of P_i and P_i' and computing the morphed vertices P_i^m the result shown in the middle is obtained. Problems, however, arise when the vertices P_i^m are connected. At the two ends of the section the intermediate representation seems to be adequate. In the central part, however, sequential chaining of the generated vertices yields spurious 'stumps' and the interpretation of the shape of the intermediate representation becomes unclear. Obviously the computed vertices P_i^m do not all pertain to the final shape of the intermediate representation. This is a major drawback of the triangulation-based method. Even though this approach is simpler because no vertices must be added and no indirect matching is necessary it is insufficient as a basis for the morphing transformation. Finding a solution for the problematic zones, however, does not seem to be trivial. Due to these shortcomings it was decided not to pursue this approach further. In the following evaluation

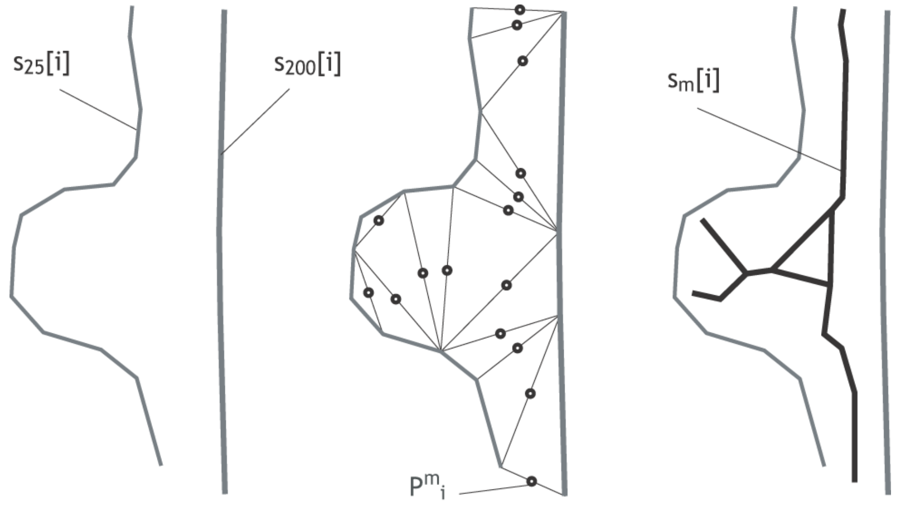


Figure 6.17: Example for a morphing transformation with the triangulation-based method.

only examples computed through the first method will be discussed.

6.4.3 Evaluation

For the evaluation and the discussion of the results of the morphing transformation technique five examples within the test area are presented (Figures 6.18 - 6.21). The first example shows at which positions the missing vertices P'_t are added and displays representations for several scale factors s_f. The second and third examples are used to evaluate the cartographic quality of the approach for more complex sections. At the end two examples of morphed river objects are discussed that partially failed to maintain the characteristics of the original data set. Table 6.4 gives an impression of the number of vertices involved.

	Example 1	**Example 2**	**Example 3**	**Example 4**	**Example 5**
m ($s_{25}[i]$)	39	127	178	112	99
n ($s_{200}[i]$)	9	15	26	16	11
h	30	112	152	96	88
$\frac{n}{m}$	0.231	0.118	0.146	0.143	0.111

Table 6.4: Number of vertices for s_{25} and s_{200} as well as the number h of missing vertices. The last row shows the ratio between n and m.

Table 6.4 lists the number of vertices for the five examples. The first row shows the number m for the representation of section $s_{25}[i]$, the second row the number n for $s_{200}[i]$. The difference h of missing vertices is listed in the third row. The ratio $\frac{n}{m}$ for the selected examples is shown in the last row. The *length* of the sections (and thus the number of vertices of a section) is defined in the preprocessing part of the morphing process. The length of a section (for the representation s_{25}) is crucial for the morphing transformation because the morphing method is applied only on the section and not on the full length of the river object. Hence, river objects are subdivided into shorter sections during preprocessing in order to improve the cartographic quality of morphing.

The first example is illustrated in Figure 6.18. Several intermediate representations (dashed lines) of a section have been computed by the morphing process. In this example $s_{25}[i]$ counts $m = 39$ and $s_{200}[i]$ $n = 9$ (displayed in gray). The first step – temporarily adding the missing number of vertices $h = 30$ to compute the morphing process – is illustrated by the black points P'_t on the representation $s_{200}[i]$. The inserted vertices on $s_{200}[i]$ exhibit an even distribution. This is due to the Gaussian smoothing procedure that was applied to $s_{25}[i]$ in the preprocessing step, achieving a consistent alignment of the vertices.

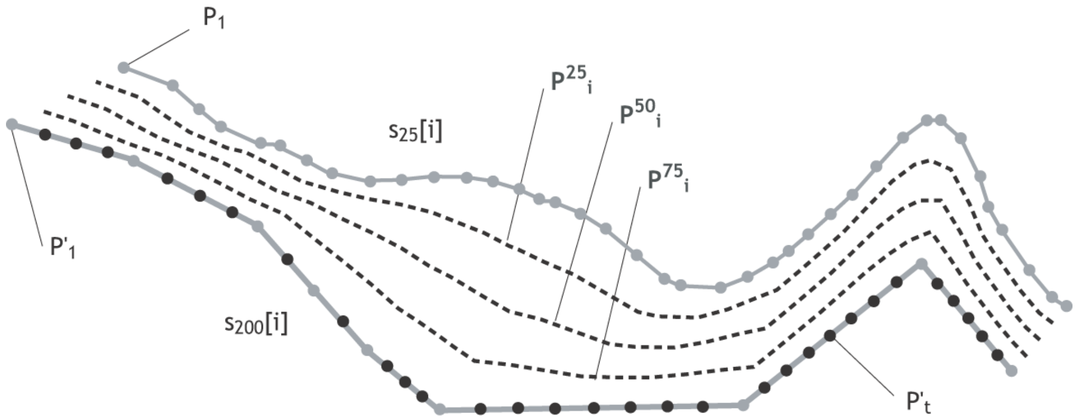


Figure 6.18: Example 1 of the morphing transformation technique applied to a section of a river object (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

The best results are obtained in the morphing transformation if the section is short but defined by a large number of vertices. Consequently, the distances between the individual vertices P'_i and P'_{i+1} are smaller and thus a better reference between corresponding vertices can be achieved. If

$$m \longrightarrow \infty \qquad \Longrightarrow D_m \longrightarrow 0$$

the best possible result for the vertex matching is accomplished, implying the best result for the morphing process. In contrast, if the section consists only of few vertices the possibility of matching the wrong vertices increases and a suboptimal result may be reached. For the section shown here three different intermediate scales are illustrated. The scale factors are $s_f = 25.0$, $s_f = 50.0$ and $s_f = 75.0$ (roughly corresponding to the scales ~1:28'000, ~1:50'000 and ~1:100'000 computed by equation (6.2). Computing a representation with $s_f = 0.0$ or $s_f = 100.0$ would be equivalent to $s_{25}[i]$ or $s_{200}[i]$, respectively.

The second example illustrated in Figure 6.19 defines a longer section with many more vertices. The number of vertices of representation $s_{25}[i]$ is $m = 127$ and of $s_{200}[i]$ is $n = 15$. That means that $h = 112$ vertices must be inserted over the full length of the section. Since the vertex insertion process is an iterative procedure it is time-consuming. Shorter sections would therefore be better for on-demand web mapping due to reasons of performance.

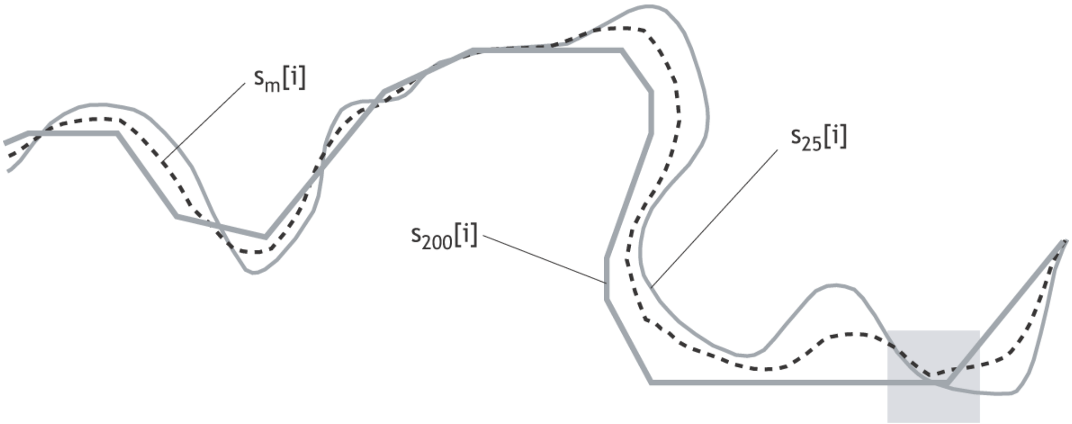


Figure 6.19: Example 2 for a long section of a river object with a large amount of vertices (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

Regardless of the computing time required Figure 6.19 depicts a good result also for longer sections. The example shows a morphed line object for a scale factor of $s_f = 50.0$. Only in the highlighted part of the result small inconsistencies appear, where the generated line object does not lie between the two 'keyframes' $s_{25}[i]$ and $s_{200}[i]$. This, however, does not really pose a problem.

As a consequence of vertex insertion the granularity (i.e. the spacing between consecutive vertices) of the morphed lines is increased and hence may be too fine for the target scale. This effect can be countered by applying a vertex weeding procedure (e.g. using the Douglas-Peucker algorithm) in order to eliminate collinear or near-collinear vertices. The decision on whether to apply a weeding procedure depends first on cartographic necessity and second on performance issues. Particularly, a server-side implementation it may become too costly to transmit excessive vertices that do not really contribute to defining the shape of cartographic lines.

Although the morphing transformation is executed sequentially on one section at a time it can also be applied to more sections or to a whole river object. Figure 6.20 shows such an example where several sections of two river objects R_i and R_j are defined. This third example consists of $m = 178$ and $n = 26$ vertices, based on all sections concerned. Computing such a 'complex' example with different rivers and different sections does not impose additional constraints or needs in the case of the river network.

River junctions can be maintained if the junction points are used as endpoints of sections. Such a case is shown in the highlighted area of Figure 6.20. The morphed representation (dashed line) was computed with a scale factor of $s_f = 50.0$ (∼1:100'000).

6.4.4 Limitations of the Method

While the morphing transformation has its merits it also exhibits certain limitations, which will be discussed in this section. Some restrictions have already been noted above, concerning

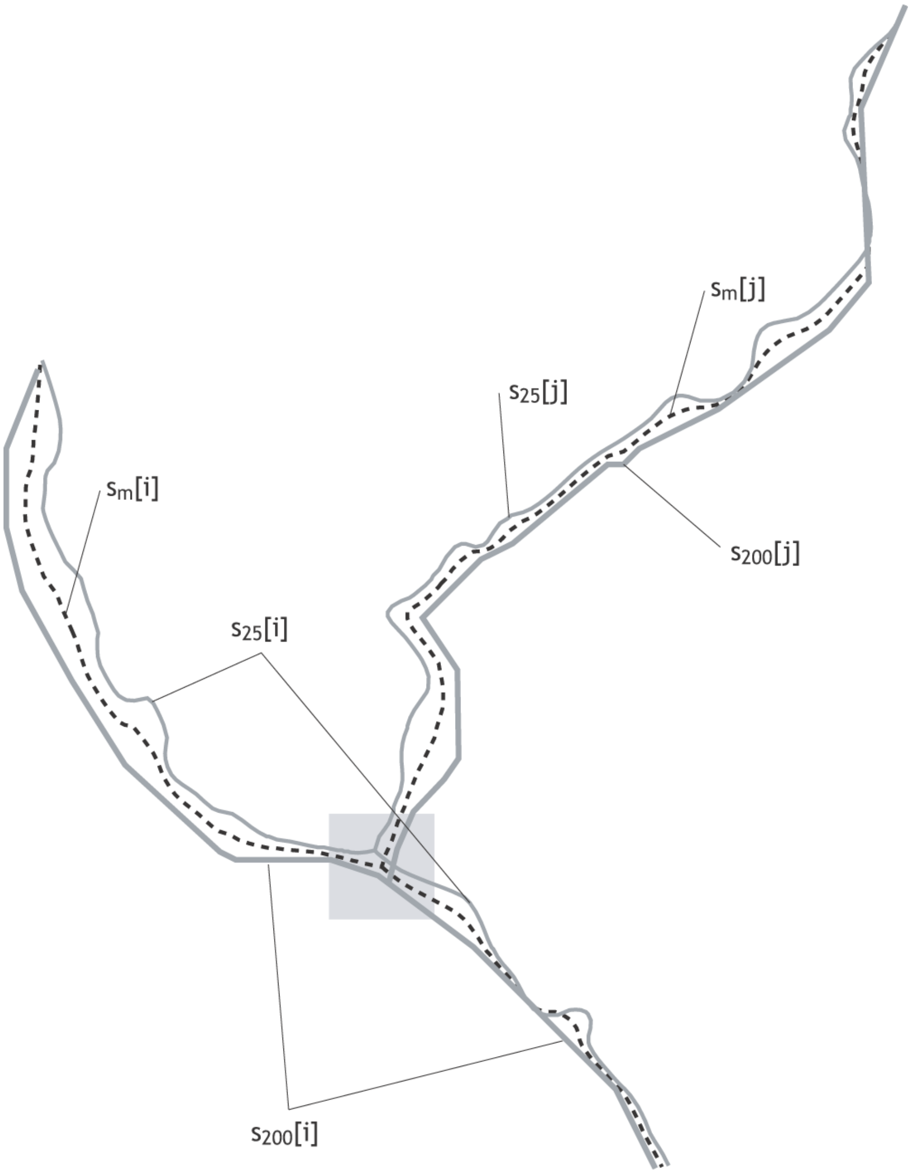


Figure 6.20: A more complex example (Example 3) implying two rivers and several sections (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

primarily the great effort needed during the preprocessing phase for line partitioning and vertex insertion. Since preprocessing only has to take place once and the result can be stored in the MSDB, its effect on the real-time part of the actual morphing procedure can be neglected. In this section, however, we will concentrate on the cartographic limitations of the method. Figure 6.21 shows two examples where in some parts an inappropriate result is obtained.

Figure 6.21 a) highlights two problematic constellations:

- α: For strongly undulating of parts of $s_{25}[i]$ the bends of the intermediate representation (dashed line) occur in some cases in advance or delayed with respect to the representation of LOD_{25}. The highlighted area α shows such an example where the morphed line responds too early. The problem lies in the distribution of the temporarily added vertices h and the following vertex matching process. Since the section is very long but the number of vertices $m = 99$ is relatively small poor matching and thus poor morphing is the consequence.
- β: Sometimes a river can be described by two or more autonomous branches (braided streams) as in the highlighted area β. In this case *one* counterpart must be defined in advance manually in the preprocessing phase which is used for the morphing process.

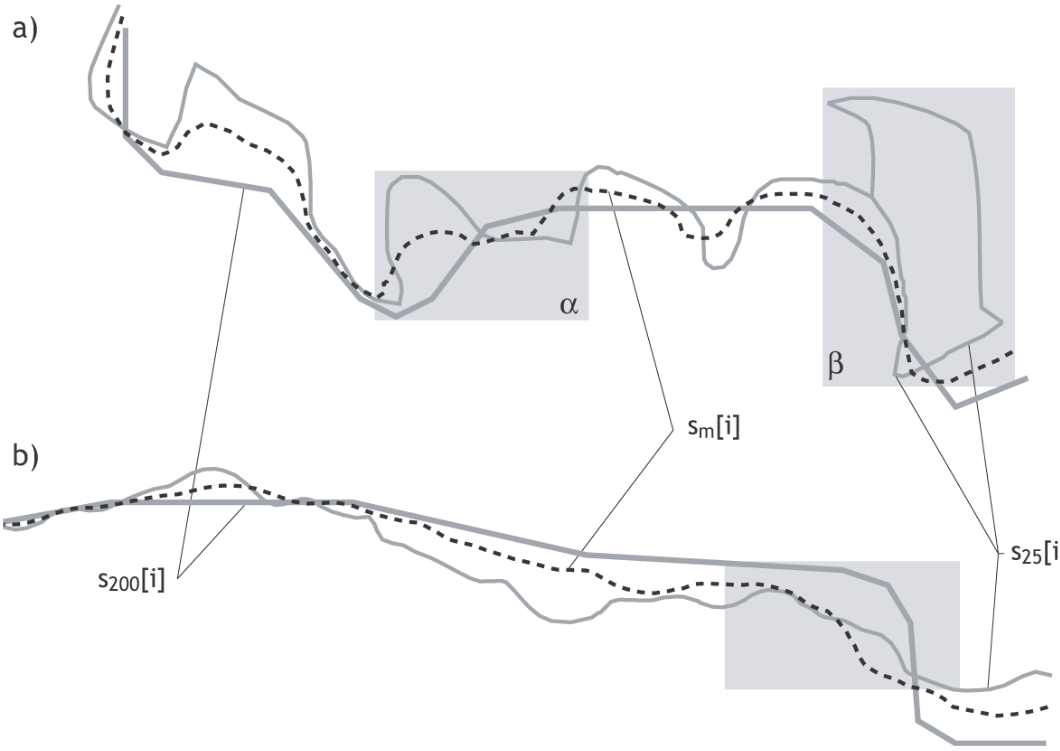


Figure 6.21: Examples 4 and 5 showing the cartographic limitations of the morphing method (not to scale). Data: VECTOR25/200 © swisstopo (BA034957).

The example in Figure 6.21 b) depicts a similar effect as in the highlighted area α of example a). If a significant shape is defined in only one of the two representations (as illustrated in the highlighted area for $s_{200}[i]$) a kind of 'uncontrolled' shape of the morphed section results. These problems could be alleviated by using additional LODs. The closer (and thus more similar) the LODs in the MSDB are, the smoother the morphing operation.

As mentioned before the granularity of the results is not always adapted to the requested map scale. In some cases small perturbations influence the characteristic shape of a line object giving it an unnatural behavior form. Whether such unaesthetic effects occur strongly depends on the shape of the more detailed 'keyframe' of LOD_{25}. The more sinuous s_{25} is compared to s_{200} the more probable the occurrence of such artifacts becomes. As mentioned above, however, such effects can be countered by applying a vertex weeding procedure as a post-processing step, following the morphing transformation.

Another weak point of the approach is the inconvenience of not having the possibility of choosing a scale value directly. Instead, the user must define a scale factor s_f between 0.0 and 100.0 which is not intuitive. This problem, however, can be eliminated by developing a conversion formula between map scales and s_f based on systematic empirical studies. Once such a conversion formula is available the system could automatically choose the appropriate s_f for a given target scale.

Obviously, the cartographic quality of the results obtained by the morphing transformation depends strongly on the original data sets and the partitioning in the preprocessing phase. If additional LODs are available spurious effects created by the process can be avoided.

An extension of this technique to large polygonal data (such as the feature class *lake*) should be addresses by future research.

Chapter 7

Conclusion

7.1 Discussion

7.1.1 Objectives and Approach

The main objectives of this work were presented in sections 1.2 and 2.5.2. The point of departure is the assumption that despite the fact that map generalization is a computationally intensive task, capabilities for scale changing and generalization should be added to on-demand mapping services on the web. While automated real-time map generalization is presently not feasible – some generalization operations are still unsolved, others are too time-consuming – a combination of both a multi-scale databases (MSDB) and map generalization is proposed to exploit the strengths of both approaches in mutual support of each other. On the one hand, MSDB alleviate and speed up the generalization task and even substitute missing generalization operators. On the other hand, generalization capabilities render MSDB more flexible and lead to better cartographic results.

7.1.2 Achievements

As noted in chapter 1, the main goal of this work was the integration of multi-scale databases and cartographic generalization for on-demand web mapping. In the cartographic research domain both methods have been studied and several approaches have been proposed. In almost all cases the research was focused on one of them. The proposed conceptual framework presented in chapter 3 combines these two methods to improve on-demand web mapping. It describes how multi-scale databases can support the cartographic generalization process. The advantage of this integration is that the complex generalization process can be subdivided into two phases, an *off-line* preprocessing and an *on-line* real-time phase, which allows two simpler generalization processes to be kept. The results of the preprocessing phase are stored as LODs in the multi-scale database. The second phase will be computed in real-time based on the LODs stored in the MSDB. The strength of this framework is that by means of a few number of LODs (in our case two) each arbitrary scale between the keyframes can be generated.

As shown in chapter 4, the MSDB contains specific additional information which supports the generalization process in the on-line phase. This information is computed in advance and enriches the original data sets. One of the main points of the data enrichment process is the linking of corresponding objects (for linear feature classes) or groups of objects (buildings) in different levels of detail. The proposed matching and linking approach for the feature class *river network* can be used, for example, for the morphing transformation and thus for the generalization process. Linking the same object in the available LODs is one of the most important condition for the interaction between MSDB and generalization. This can be seen by the discussed matching method for the feature class *building* presented in section 6.2.2. The advantage of linking *groups* of building objects instead of individual objects is, that the typification operation in the generalization process can be supported.

The strength of the approach of integration of MSDB and generalization is described by the two techniques presented in chapters 5 and 6: mesh simplification and morphing transformation. The achieved results of these two methods has shown how a MSDB can support the creation of a representation for an arbitrary scale on request. For the mesh simplification, the information of both LODs is used to define the number of *building objects* for the requested scale. The morphing uses these two LODs as keyframes to compute an intermediate representation.

The data set used for this work is based on LOD_{25} (represented by VECTOR25) and LOD_{200} (VECTOR200) provided by the Swiss Federal Office of Topography. Using the presented techniques intermediate representations for several feature classes can be generated. Even though the step from LOD_{25} to LOD_{200} is significant concerning the changes of content, the results shown in the preceding chapters are generally encouraging. Weaknesses do exist but can be alleviated by a variety of improvements discussed in the previous chapters. In particular, the cartographic quality of the proposed methods could be improved by adding further LODs to the MSDB. In this work, however, we deliberately chose to use two distant LODs to really put our proposed framework to the test.

7.1.3 Insights

Necessity of a Multi-Scale Database

Using a multi-scale database provides many advantages in regard to on-demand web mapping. Defining a MSDB with several LODs allows the map generation process to be enormously simplified. To get a map at a requested scale not at the most detailed but at the most adapted LOD the generalization process can be applied. Given different LODs allows an in-between representation to be more easily created. By enhancing the original data set, the database can be made more valuable for the generalization process. The added information should contribute by substituting missing or very time-consuming generalization algorithms or supporting the processing of algorithms (for example the determination of the orientation of newly created placeholders). The existence of several LODs can assume very important functions (for example as termination criterion for the edge collapsing process).

Beside the design of the multi-scale database, the linking of the LODs is crucial to benefit the existence of different representations. Since in cartography in most cases a [1:1]-relation is not always possible (e.g. buildings) other solutions must be found to get such an easy to handle relation. The presented approach for the feature class *building* has described such a way. By defining smaller groups of objects belonging together a matching of them can be fulfilled without problems (group matching). Instead of having a [n:m]-relation between the building objects represented in the different LODs a [1:1]-relation can be defined. This allows a simpler handling of the objects concerning the later generalization process.

The better the original data set is enriched in respect of the generalization process the better the improvement that can be achieved. Out of the existing representations (LODs) the computation of an in-between one becomes easier. A good example is the determination of the orientation of each building object in advance, meaning its computation at run-time is no longer necessary. As the value is known it can be used to define the orientation of a placeholder. Of importance is the fact that the enhanced information must be well-tuned and adapted to the defined generalization process. Only then does an enrichment of the database make sense.

Whilst the presented results for the selected test area are good, the crucial importance of the number and scale of the LODs should not be underestimated. As illustrated in section 3.4.1, the definition of such limits of applicability (range until which scale a LOD can be used) is not trivial. First experiences have been made by studying the 'generalization complexity' described in 3.4.2. These analyses can be considered only as one component of the definition process. Although the exploration was merely qualitative, it characterizes the application of the different generalization operators within a large scale band and the contribution ('costs') of each of them to the whole process. A good example can be described by Figure 3.6 for the feature class *building*. In the scale range [1:50k,1:200k] the main 'costs' of the generalization process are caused by the operator 'typification'. A reduction of the 'costs' of the generalization process in this scale range can be achieved by implementing a technique for typification which involves 'costs' that are as small as possible.

Adapted Generalization Process

On-demand web mapping requires a system that is as flexible as possible to create maps for different purposes. Just in the context of Internet mapping, various needs must be covered, as not all users want to have the same map. Depending on the requested aims different conditions and terms must regarded. Thus it is important to define scenarios which restrict all the possibilities of web mapping to an acceptable number of cases.

By the definition of the three scenarios 'Quick and Dirty', 'Map for the Web' and 'Map for Print' the most relevant cases are covered. According to the requirements of these scenarios, the map generation process must be adjusted. For that matter also the kind or severity of generalization for the mapping process must be adapted. The range is between no generalization (scenario 1) and a full automated process regarding all possibilities (scenario 3).

The focus of this work was set on the second one, where a process has been proposed for the feature classes *road network*, *building* and *river network*. Defining these classes was made to try to consider the most commonly used types of geometric primitives (points and lines) in topographic maps. For both, techniques have been proposed which take account of their use in multi-scale databases: **mesh simplification** for point objects and **morphing transformation** for linear objects.

The starting point of the mesh simplification technique was the work carried out by Hoppe *et al.* (1993). He proposed a mesh optimization technique for the reduction of number of vertices in an initially dense mesh of triangles. Their approach was to minimize an energy function to resolve this problem. Yet, the algorithm proposed was, from our point of view, not sufficiently adaptable to the requirements of on-demand web mapping. Thus an accommodation of the algorithm to a better suited one has been carried out: the mesh simplification technique.

As presented in chapter 6, the results achieved by applying this technique are pleasing. Both for computing the position of the new placeholders and for the computation of the dimension of the building object the technique is well suited. Even though only the parameter distance is applied in the typification process the results exceed the expectations made in advance. Next to the method itself the definition of size-optimized groups (in advance) of buildings plays a crucial role in the success. Through the implementation of additional parameters the quality of the algorithm can be influenced. Considering more attributes (like size or importance of the building) would improve the approach. In spite of the achieved outcomes some drawbacks (especially the dimension of the new objects) as discussed in the previous section remain. Nevertheless the presented mesh simplification technique permits to typify buildings for smaller map scales.

Out of the shape blending approach of Sederberg and Greenwood (1992) an adjusted technique for on-demand web mapping – the morphing transformation technique – has been developed to generalize linear feature objects (e.g. river network). The examples presented in chapter 6 characterize in most cases good results. The success of the application is determined by two elements: on the one hand, the quality of the original data set and on the other, the partitioning of the network into sections in the preprocessing phase. The strength of the approach is given by the possibility of defining in a simple way an in-between state out of two keyframes for a desired scale. Weak points remain the granularity of parts of the results which are not scale-adapted and the difficulty of assigning the correct scale to a computed representation.

Both approaches have shown a strength to being applied in automated map generalization in connection with multi-scale databases. The presented framework for on-demand web mapping and the discussed results pointed out how web mapping can be improved by integrating multi-scale databases in the cartographic generalization process.

7.2 Outlook

7.2.1 Possible Improvements

The presented conceptual framework and the proposed techniques have shown that in the context of on-demand web mapping several topics for enhanced research work can be found. Since web mapping does not only consist of cartographic generalization various foci can be set for further work. They range from developing progressive transmission techniques for vector data (Buttenfield (1999), Bertolotto and Egenhofer (2001)), designing models for a multi-scale database (Jones *et al.* 1996) or matching corresponding data in different representations (LODs) (Sester *et al.* 1998) to defining a generalization process for real-time mapping for small-display cartography (Harrie *et al.* (2002), Reichenbacher (2001)). As it can be seen future work can be focused on a plurality of topics. The following list gives a variety of possibilities concerning this work.

Additional Feature Classes

In general, maps do not only contain the three selected feature classes *road network*, *building* and *river network* but also various other classes. To mention are point objects (representing special features such as reservoirs or hydropower stations), linear objects (e.g. railroads or power lines) and areal objects (e.g. lakes or vegetation). Specific or derived generalization processes for the implementation of these additional feature classes are needed.

The implementation of the two techniques *mesh simplification* and *morphing transformation* have been accomplished for 'point' objects (e.g buildings represented by their center of gravity) and linear objects (e.g. river network). Other feature classes which can be represented by one of these two geometric primitives (e.g. 'railroad') similar approaches are thinkable. Not discussed is the application of one of these two techniques to areal feature classes (like 'forest' or 'lakes').

Multi-Scale Database

In context with the design and enrichment of a multi-scale database the points can be focused on:

- **Data enrichment**: The original data set must be enriched and adjusted to the subsequent generalization process. Here, more specific information must be added or precomputed to reduce the amount of processes and calculations while the generalization procedure. The information must be very specific disencumbering one special operator from calculations which need not be computed in run-time. Another focus can be set on the development of storing not only simple values (like orientation, Horton order) but also complex information about the structure of a pattern (building alignment).

- **Data matching**: To develop of robust processes especially for feature classes represented by linear (road network) or areal objects (vegetation). In addition a refinement of the matching process for the feature class *building* by defining other kinds of groups or clusters based on statistical methods can be carried out. For the matching process in general strategies for solving [n:m]-relations are looked for.

- **Data model**: The presented schema of the MSDB is kept relatively simple. Here a more specific model including the insights gained in this work can be designed and adapted for the required needs of web mapping.

Generalization Process

- **Displacement**: The displacement operation takes up an exceptional position in the generalization process as it is related to all feature classes. Most of the proposed approaches for the implementation of a displacement operation (e.g. Bader (2001)) are not suitable for on-demand web mapping due to reasons of compute performance (mainly for the defined scenarios 1 and 2). Thus, new methods and solutions must be found to minimize or even avoid the problem of overlapping objects without using a displacement operation at the end of the mapping process. Experiments with simple but computationally efficient methods have been presented by Mackaness and Purves (2001).

- **New generalization techniques**: Additional feature classes require also new techniques for the generalization process. Possible approaches may be found in the computer science or computer vision (Gallagher 1994). Within web mapping solutions must be found to decrease the 'costs' created by the application of each generalization operator.

- **Mesh simplification**: Concerning the mesh simplification other semantic attributes may be taken into account: importance or size of a buildings object. Beside this the focus must be set on preserving structures of building pattern.

- **Morphing transformation**: For the morphing transformation the computation of a representation for a determined scale must be solved. The second presented approach based on a triangulation 6.4.2 is promising good results if the specified problems can be solved.

7.2.2 Final Remarks

With on-demand web mapping a new topic in generalization research is originated. It is no longer the absolute best possible representation of a map at a requested scale which is needed but a representation that fulfills the purposes and requirements presented by the user. Within these needs the question of the position of the generalization approach must be reviewed. On the one hand the possibilities offered by the new technologies must be utilized to define new kinds of representations of spatial data and on the other hand the demands of the user must be considered.

The emergence of mobile computing and wireless devices has brought a whole palette of new possibilities for cartography. New mobile information devices such as PDAs and Smartphones provide possibilities to produce user maps on-demand, anywhere, anytime (Reichenbacher 2001). With the introduction of *location-based services* (Burghardt *et al.* 2003), where spatial information is enhanced by other information, the importance to keep map as flexible as possible increases. Thus, the generalization process must be adapted to the requirements and constrictions provoked by these new possibilities (for example Harrie *et al.*, 2002) proposes a solution for displaying maps on small display devices). Next to all these innovations the process of creating maps still remains the same. In all cases a generalization process must be developed to adjust the contents of a map to the requested scale.

Appendix A

Glossary

A.1 Terminology

To avoid misunderstandings a list of often used terms in this thesis have been defined.

Feature class	A feature class is a logical classification of feature elements with common characteristics and structures where similar methods can be used (e.g. river network; buildings; road network).
Feature type	Feature type groups elements of a similar processing (e.g. main roads).
Feature element	Feature elements of the same type are grouped in a feature class (e.g. 1^{st} class road; main river).
Object	An Object is a real existing geographical instance belonging to one feature element (e.g. river rhine; highway A1).
Database	A database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. The most prevalent type of database is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses. Databases contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles.
Database Management System	A database management system (DBMS), sometimes just called a database manager, is a program that lets one or more computer users create and access data in a database. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, who else may also be accessing the data. In handling user requests, the DBMS ensures the integrity of the data (that is, making sure it continues to be accessible and is consistently organized as intended) and security (making sure only those with access privileges can access the data).

Digital Landscape Model	A Digital Landscape Model (DLM) is an object orientated topographic database. The data structure facilitates spatial analysis and linkage of geographic objects to external data. The geometric form of the DLM is vector. The DLM often contains explicit or implicit topological information. The objects, their attributes and the relations between the objects are referred to in terms of real world entities (e.g. VECTOR25, BD TOPO®).

The denotation of *Feature class*, *Feature type* and *Feature element* are utilized especially in combination with Geographic Information Systems (GIS) and modelling of spatial data. In the notation of the *Unified Modelling Language* UML^{TM} the term 'object' is defined as *entity* and the term 'feature element' as *entity type*.

A.2 Symbols Used

α	Orientation of a building object.
A_{Bi}	Summarized area of building objects in the same group.
a_{min}	Shortest distance between two vertices v.
b	$\{b_1, ..., b_m\}$, building objects.
B_{Poly}	Number of groups of building objects after buffering and partitioning.
BUF	Number of groups of building objects after buffering with distance d.
m_{25}, m_{200}	Number of vertices representing the center of gravity of buildings.
n_l	Number of LODs in the multi-scale database.
nb	Number of building objects in the requested map scale.
LA_i	Limits of applicability.
LOD_i	Level of detail.
LOD_{25}, LOD_{200}	LODs representing objects at 1:25'000 resp. 1:200'000.
M_i	Mesh.
RS_U	Requested scale by the user.
$S_{i,l}$	Lower limit of the ScaleChanging module.
$S_{i,h}$	Higher limit of the ScaleChanging module.
s_f	Scaling factor between 0.0 and 100.0 to determine the requested scale.
s_u	Factor to define the strength of the thinning process of densely populated areas.
s_{25}, s_{200}	Sections of a river in LOD_{25} respectively in LOD_{200}.
$Sc(LOD)$	Scale value of the LOD. For example for scale 1:50'000, $Sc(LOD) = 50$.
v	$\{v_1, ..., v_m\}$, set of vertex positions of a mesh.

Bibliography

Abraham, I. M. (1989). *Automated Cartographic Line Generalization and Scale-Independent Databases*. Ph.D. thesis, University of Glamorgan.

Adam, N. R. and Gangopadhyay, A. (1997). *Database Issues in Geographic Information Systems*, volume 6 of *Advances in Database Systems*. Kluwer Academic Publishers, New York.

AGENT Consortium (1999). AGENT Workpackage C1 - Selection of Basic Measures. Technical report, Department of Geography, University of Zurich, Zurich.

Arleth, M. (1999). Problems in Screen Map Design. In: *Proceedings of the 19th ICA/ACI Conference*, pp. 849–857. Ottawa.

Badaud, J., Witkin, A. P., and Duda, R. O. (1986). Uniqueness of the Gaussian Kernel for Scale-Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8** (1), pp. 26–33.

Bader, M. (2001). *Energy Minimizing Methods for Feature Displacement in Map Generalization*. Ph.D. thesis, Department of Geography, University of Zurich, Zurich.

Bader, M., Barrault, M., Regnauld, N., Mustière, S., Duchêne, C., Ruas, A., Fritsch, E., Lecordix, F., and Barillot, X. (1999). AGENT Workpackage D2 - Selection of Basic Algorithms. Technical report, Department of Geography, Univerity of Zurich.

Bader, M. and Weibel, R. (1997). Detecting and Resolving Size and Proximity Conflicts in the Generalization of Polygonal Maps. In: *Proceedings of the 16th ICA/ACI Conference*, pp. 1525–1532. Stockholm.

Ballard, D. H. (1981). Strip-Trees: A Hierarchical Representation for Curves. *Communications of the ACM*, **1** (4), pp. 310–322.

Barrault, M. and Bader, M. (2002). Les algorithmes de dèplacement. In: A. Ruas (editor), *Généralisation et representation multiple*, pp. 287–302. Lavoisier, Paris.

Barrault, M., Regnauld, N., Duchêne, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., and Weibel, R. (2001). Integrating multi-agent, object-oriented and algorithmic techniques for improved automated map generalization. In: *Proceedings of the 19th ICA/ACI Conference*, pp. 2110–2116. Beijing.

Baudoin, C. and Hollowell, G. (1996). *Realizing the Object-Oriented Lifecycle*. Prentice Hall, Upper Saddle River.

Beard, K. (1987). How to Survive on a Single Detailed Database. In: *Proceedings of AUTO-CARTO 8*, pp. 211–220. Baltimore, Maryland.

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

Bédard, Y., Proulx, M.-J., Larrivée, S., and Bernier, E. (2002). Modeling Multiple Representations into Spatial Data Warehoses: A UML-Based Approach. In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

Berg de, M., Kreveld van, M., and Schirra, S. (1995). A New Approach to Subdivision Simplification. In: *Proceedings of AUTO-CARTO 12*, pp. 79–88. Charlotte.

Bertin, J. (1967). *Semiologie Graphique. Les Diagrammes - Les Reseaux - Les Cartes*. Mouton-Gauthier-Villars, Paris-La Haye.

Bertin, J. (1983). *Semiology of Graphics*. University of Wisconsin Press, Madison, Wisconsin.

Bertolotto, M. and Egenhofer, M. J. (2001). Progressive Transmission of Vector Map Data over the World Wide Web. *GeoInformatica*, **5** (4), pp. 345–373.

Bézier, P. E. (1968). How Renault Uses Numerical Control for Car Body Design and Fooling. *Society of Automotive Engineer's Congress*.

Bobzien, M. and Morgenstern, D. (2002). Geometry-Type Chance in Model Generalization - a Geometrical or a Topological Problem? In: *Proceedings of the Joint Workshop on Multi-Scale Representations of Spatial Data*. Ottawa.

Boffet, A. and Rocca Serra, S. (2001). Identification of Spatial Structures within Urban Blocks for Town Characterisation. In: *Proceeding of the 20th ICA/ACI Conference*, pp. 1974–1983. Beijing.

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

Brazile, F. (2001). *Semantic Infrastructure and Methods to Support Quality Evaluation in Cartographic Generalization*. Ph.D. thesis, Department of Geography, University of Zurich, Zurich.

Bruegger, B. and Frank, A. (1989). Hierarchies over Topological Data Structures. In: *Proceedings ASPRS-ACSM Annual Convention*, volume 4, pp. 137–145. Baltimore, MD.

Burghardt, D., Edwardes, A., and Weibel, R. (2003). WebPark - Entwicklung eines mobilen Informationssystems für Erholungs- und Naturschutzgebiete. *Kartographische Nachrichten*, **53** (2).

Buttenfield, B. (1993). Research Initiative 3: Multiple Representations. Closing Report. Technical report, National Center for Geographic Information and Analysis.

Buttenfield, B. (1999). Progressive Transmission of Vector Data on the Internet: A Cartographic Solution. In: *Proceedings of the 19th ICA/ACI Conference*, pp. 581–590. Ottawa.

Buttenfield, B. (2002). Transmitting Vector Geospatial Data across the Internet. In: M. J. Egenhofer and D. M. Mark (editors), *Second International Conference, GIScience 2002*, Lectures Notes in Computer Science, pp. 51–64. Springer, Berlin.

BUTTENFIELD, B. AND DELOTTO, J. S. (1989). Multiple Representation: Scientific Report for the Specialist Meeting. Technical Report 89-3, National Center for Geographic Information and Analysis.

BUTTENFIELD, B. AND MACKANESS, W. A. (1991). Visualization. In: D. J. Maguire, M. F. Goodchild, and D. W. Rhind (editors), *Geographic Information Systems: Principles and Practice*, pp. 427–443. Longman, London.

CECCONI, A. AND GALANDA, M. (2002). Adaptive Zooming in Web Cartography. *Computer Graphics Forum*, **21** (4), pp. 787–799.

CECCONI, A. AND WEIBEL, R. (2001). Map Generalization for On-Demand Mapping. *GIM International, Magazine for Geomatics*, **15** (5), pp. 12–15.

CHEN, P. P. (1976). The Entity-Relationship Model - Towards a Unified View of Data. *ACM Transactions on Database Systems*, **1** (1), pp. 9–36.

CHITHAMBARAM, R., BEARD, K., AND BARRERA, R. (1991). Skeletonizing Polygons for Map Generalization. In: *Cartography and GIS/LIS, ACSM-ASPRS*, pp. 61–72. Baltimore.

CHRISTOPHE, S. AND RUAS, A. (2002). Detecting Building Alignments for Generalisation Purposes. In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

CODD, E. F. (1970). A Relational Model for Large Shared Data Banks. *Communications of the ACM*, **13** (6), pp. 377–387.

CROMLEY, R. G. AND CAMPBELL, G. M. (1992). Integrating Quantitative and Qualitative Aspects of Digital Line Simplification. *The Cartographic Journal*, **29** (1), pp. 25–30.

DATE, C. J. (2000). *An Introduction to Database Systems*. Addison-Wesley, Reading, Mass., 7th edition.

DENT, B. D. (1990). *Principles of Thematic Map Design*. Addison-Wesley, Reading, Mass.

DESANTI, M. AND GOMSI, J. (1994). A Comparison of Object and Relational Database Technologies. *Object Magazine*, **3** (5), pp. 51–57.

DEVOGELE, T. (1997). *Processus d'intégration et d'appariement de Bases de Données Géographiques*. Ph.D. thesis, Université de Versailles.

DEVOGELE, T. (2002). A new Merging Process for Data Integration Based on the Discrete Frèchet Distance. In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

DEVOGELE, T., PARENT, C., AND SPACCAPIETRA, S. (1998). On Spatial Database Integration. *International Journal of Geographic Information Systems*, **12** (4), pp. 335–352.

DEVOGELE, T., TREVISAN, J., AND RAYNAL, L. (1996). Building a Multi-Scale Database with Scale-Transition Relationships. In: *Proceedings of the 7th International Symposium on Spatial Data Handling*, Advances in GIS Research II, pp. 6.19–6.33. Delft.

DOUGLAS, D. H. AND PEUCKER, T. K. (1973). Algorithms for the ReGeneralization Constraintsduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *The Canadian Cartographer*, **10** (2), pp. 112–123.

DUDZIAK, T. (2000). Level of Detail - Eine Übersicht. CG/CV-Seminars Sommersemester 2000.

EDELSTEIN, H. (1994). Unraveling Client/Server Architecture. *DBMS 7*, **34** (7), pp. 34–40.

EGENHOFER, M. J. AND FRANK, A. U. (1992). Object-Oriented Modeling in GIS. *URISA Journal*, **4** (2), pp. 3–19.

ELZAKKER VAN, C. (2001). Users of Maps on the Web. In: M. J. Kraak and A. Brown (editors), *Web Cartography: Developments and Prospects*, pp. 37–52. Taylor and Francis, London.

FERINGA, W. (2001). File Formats and Plugins. In: M. J. Kraak and A. Brown (editors), *Web Cartography*, pp. 177–193. Taylor and Francis, London.

FILIN, S. AND DOYTSHER, Y. (2000). The Detection of Corresponding Objects in a Linear-Based Map Conflation. *Surveying and Land Information Systems*, **60** (2), pp. 117–128.

FOLEY, J. D., VAN DAM, A., FEINER, S. K., AND HUGHES, J. (1996). *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Massachusetts.

GABAY, Y. AND DOYTSHER, Y. (1994). Adjustment of Line Maps. In: *Proceedings og GIS/LIS'94*, pp. 191–199. Phoenix.

GALANDA, M. AND WEIBEL, R. (2002). An Agent-Based Framework for Polygonal Subdivision Generalisation. In: D. Richardson and P. van Oosterom (editors), *Advances in Spatial Data Handling*, pp. 121–135. Springer, Berlin.

GALLAGHER, R. S. (1994). *Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis*. CRC Press.

GALLAUGHER, J. AND RAMANATHAN, S. (1996). Choosing a Client/Server Architecture. A Comparison of Two-Tier and Three-Tier Systems. *Information Systems Management Magazine*, **13** (2), pp. 7–13.

GARDELS, K. (1998). *A Web Mapping Scenario*. Open GIS Consortium, Inc. (OGC), Internet. URL http//:www.opengis.org/wwwmap/98-068.pdf.

GLOVER, E. AND MACKANESS, W. A. (1999). Dynamic Generalisation from Single Detailed Database to Support Web Based Interaction. In: *Proceedings of the 19th ICA/ACI Conference*, pp. 1175–1183. Ottawa.

GOODCHILD, M. F. (1999). Cartographic Futures on a Digital Earth. In: *Proceeding of the 19th ICA/ACI Conference*, pp. 4–12. Ottawa.

GOWER, R., PEPPER, J., AND EDWARDS, T. (1997). Landplan - Automated Generalisation Comes of Age. In: *Proceedings of the 16th ICA/ACI Conference*, pp. 126–133. Stockholm.

GRÜNREICH, D. (1985). Computer-Assisted Generalisation. In: *Papers CERCO Cartography Course*. Institut für Angewandte Geodäsie, Frankfurt a. M.

HARRIE, L. (1998). Incremental Generalisation: a Feasibility Study. In: *GIS RESEARCH UK: 6th National Conference*. Edinburgh.

HARRIE, L. AND HELLSTRÖM, A. K. (1999). A Protortype System for Propagating Updates between Cartographic Data Sets. *The Cartographic Journal*, **36** (2), pp. 133–140.

HARRIE, L., SARJAKOSKI, T., AND LEHTO, L. (2002). A Variable-Scale Map for Small-Display Cartography. In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

Højholt, P. (2000). Solving Space Conflicts in Map Generalization: Using a Finite Element Method. *Cartography and Geographic Information Science*, **27** (1), pp. 65–74.

Hoppe, H. (1996). Progressive Meshes. *Computer Graphics*, **30** (Annual Conference Series), pp. 99–108.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1993). Mesh Optimization. *Computer Graphics*, **27** (Annual Conference Series), pp. 19–26.

Horton, H. (1945). Erosional Development of Streams and their Drainage Basins. *Bulletin of the Geological Society of America*, **56**, pp. 275–370.

International Cartographic Association (1973). *Multilingual Dictionary of Technical Terms in Cartography*. Steiner, Wiesbaden.

Jaakkola, O. (1997). *Quality and Automatic Generalization of Land Cover Data*. Ph.D. thesis, Department of Geography, University of Helsinki, Helsinki.

Jones, C. B. (1991). Database Architecture for Multi-Scale GIS. In: *Technical Papers 1991 ACSM-ASPRS Annual Convention*. Baltimore.

Jones, C. B., Abdelmoty, A. I., Lonergan, M. E., van der Poorten, P., and Zhou, S. (2000). Multi-Scale Spatial Database Design for Online Generalisation. In: *Proceedings of the 9th International Symposium on Spatial Data Handling*, pp. 7b.34–7b.44. Beijing.

Jones, C. B., Kidner, D. B., Luo, L. Q., Bundy, G. L., and Ware, J. M. (1996). Database Design for a Multi-Scale Spatial Information System. *International Journal of Geographic Information Systems*, **10** (8), pp. 901–920.

Keates, J. S. (1996). *Understanding Maps*. Longman, London.

Kilpeläinen, T. (1997). *Multiple Representation and Generalization of Geo-Databases for Topographic Maps*. Ph.D. thesis, Finnish Geodetic Institute, Helsinki University of Technology.

Kilpeläinen, T. and Sarjakoski, T. (1995). Incremental Generalization for Multiple Representations of Geographical Objects. In: J. C. Müller, J. P. Lagrange, and R. Weibel (editors), *GIS and Generalization: Methodology and Practice*, pp. 209–218. Taylor and Francis, London.

Kraak, M. J. and Brown, A. (2001). *Web Cartography: Developments and Prospects*. Taylor and Francis, London.

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., and Weibel, R. (1999). The Application of Agents in Automated Map Generalisation. In: *Proceedings of the 19th ICA/ACI Conference*, pp. 1225–1234. Ottawa.

Lang, T. (1969). Rules for the Robot Draughtsmen. *The Geographical Magazine*, **42** (1), pp. 50–51.

Ledgard, H. (1983). *ADA, An Introduction*. Springer-Verlag, New York.

Lehto, L. and Kilpeläinen, T. (2001). Generalizing XML-Encoded Spatial Data on the Web. In: *Proceedings of the 20th ICA/ACI Conference*, pp. 2390–2396. Beijing.

Li, Z. and Openshaw, S. (1992). Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization. *International Journal of Geographic Information Systems*, **6** (5), pp. 373–389.

Lichtner, W. (1979). Computer-assisted Process of Cartographic Generalization. *Geo-Processing*, **1** (1), pp. 183–198.

Longley, P., Goodchild, M., Maguire, D. M., and Rhind, D. (1999). *Geographical information Systems: Principles, Techniques, Management, and Applications*. Wiley and Sons, New York.

Mackaness, W. and Edwards, G. (2002). The Importance of Modelling Pattern and Structure in Automated Map Generalization. In: *Proceedings of the Joint Workshop on Multi-Scale Representations of Spatial Data*. Ottawa. Http://www.ikg.uni-hannover.de/isprs/list_of_presentations.html, Accessed: February 2003.

Mackaness, W. A. (1995). A Constraint Based Approach to Human Computer Interaction in Automated Cartography. In: *Proceedings of the 17th ICA/ACI Conference*, pp. 1423–1432. Barcelona.

Mackaness, W. A. and Purves, R. S. (2001). Automated Displacement for Large Numbers of Discrete Map Objects. *Algorithmica*, **30** (2), pp. 302–311.

Martin, J. (1993). *Principles of Object-Oriented Analysis and Design*. Prentice Hall, Englewood Cliffs.

McMaster, R. (1989). The Integration of Simplification and Smoothing Algorithms in Line Generalization. *Cartographica*, **26** (1), pp. 101–121.

McMaster, R. and Shea, S. (1988). Cartographic Generalization in a Digital Environment: A Framework for Implementation in a Geographic Information System. In: *Proceedings of GIS/LIS 88*, pp. 240–249. San Antonio, Texas.

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

Monmonier, M. (1989). Interpolated Generalization: Cartographic Theory for Expert-Guided Feature Displacement. *Cartographica*, **26** (1), pp. 43–64.

Monmonier, M. and McMaster, R. (1991). *The Sequential Effects of Geometric Operators in Cartographic Line Generalization*. International Yearbook of Cartography.

Müller, J. C. (1991). Generalization of Spatial Databases. In: D. J. Maguire, M. F. Goodchild, and D. W. Rhind (editors), *Geographic Information Systems: Principles and Practice*, volume 1, pp. 457–475. Longman, London.

Mustière, S. (1998). GALBE: Adaptive Generalisation. In: *GIS PlaNet*, volume 1. Lisboa.

Mustière, S. and Moulin, B. (2002). What is Spatial Context in Cartographic Generalisation? In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

National Center for Geographic Information and Analysis (1989). The research plan of the National Center for Geographic Information and Analysis. *International Journal of Geographic Information Systems*, **3** (2), pp. 117–136.

NOAA National Geophysical Data Center (1997). *On-Demand Mapping*. Internet. URL http://www.ngdc.noaa.gov/seg/tools/gis/ondemand.shtml. Accessed: February 2003.

Oosterom van, P. (1995). The GAP-tree, an Approach to 'On-The-Fly' Map Generalization of an Area Partitioning. In: J. C. Müller, J. P. Lagrange, and R. Weibel (editors), *GIS and Generalization: Methodology and Practice*, pp. 120–125. Taylor and Francis, London.

Oosterom van, P. and Schenkelaars, V. (1995). The Development of an Interactive Multi-Scale GIS. *International Journal of Geographic Information Systems*, **9** (5), pp. 489–507.

OpenGIS (2000). *Open GIS Consortium, Inc. OpenGIS Web Map Server Interface Implementation Specification Revision 1.0.0*. Open GIS Consortium, Inc. (OGC), Internet. URL http://www.opengis.org/techno/specs/00-028.pdf.

Parent, C., Spaccapietra, S., and Zimanyi, E. (2000). MurMur: Database Management of Multiple Representations. In: *AAAI-2000 Workshop on Spatial and Temporal Granularity*. Texas.

Peter, B. (2001). Measures for the Generalization of Polygonal Maps with Categorial Data. In: *Forth ICA Workshop on Progress in Automated Map Generalization*. Beijing.

Peterson, M. P. (1994). Cognitive Issues in Cartographic Visualization. In: A. MacEachern and D. Taylor (editors), *Visualization in Modern Cartography*, pp. 27–43. Pergamon, New York.

Peterson, M. P. (1999). Trends in Internet Map Use - A Second Look. In: *Proceeding of the 19th ICA/ACI Conference*, pp. 571–580. Ottawa.

Peterson, M. P. (2001). Trends in Internet Map Use. In: *Proceeding of the 20th ICA/ACI Conference*, pp. 2306–2312. Beijing.

Plazanet, C., Affholder, J.-G., and Fritsch, E. (1995). The Importance of Geometric Modeling in Linear Feature Generalization. *Cartography and Geographic Information Systems*, **22** (4), pp. 291–305.

Puppo, E. and Scopigno, R. (1997). Simplification, LOD and Multiresolution - Principles and Applications. Principles and Applications, Eurographics '97 Tutorial Notes.

Rainsford, D. and Mackaness, W. (2002). Template Matching in Support of Generalisation of Rural Buildings. In: D. Richardson and P. van Oosterom (editors), *Advances in Spatial Data Handling*, pp. 137–151. Springer, Berlin.

Raper, J. F. and Maguire, D. J. (1992). Design Models and Functionality in GIS. *Computer and Geosciences*, **18** (4), pp. 387–400.

Ratajski, L. (1967). Phénomène des points de généralisation. In: K. Kirschbaum and K. H. Meine (editors), *International Yearbook of Cartography*, volume 7, pp. 143–152. Kirschbaum, Bonn-Bad Godesberg.

Regnauld, N. (1996). Recognition of Building Clusters for Generalization. In: *Proceedings of the 7th International Symposium on Spatial Data Handling*, pp. 4B.1–4B.14. Delft.

Regnauld, N. (1998). *Généralisation du bâti: Structure spatiale de type graphe et répresentation cartographique.*. Ph.D. thesis, Université de Provence, Aix-Marseille 1, France.

Reichenbacher, T. (2001). The World in your Pocket - Towards a Mobile Cartography. In: *Proceeding of the 20th ICA/ACI Conference*, pp. 2514–2521. Beijing.

Reumann, K. and Witkam, A. (1974). Optimizing Curve Segmentation in Computer Graphics. In: *International Computing Symposium*, pp. 467–472. Amsterdam.

Richardson, D. E. (1993). *Automatic Spatial and Thematic Generalization using a Context Transformation Model*. Ph.D. thesis, Wageningen Agricultural University, The Netherlands.

Rieger, M. and Coulson, M. (1993). Consensus or Confusion: Cartographers' Knowledge of Generalization. *Cartographica*, **30** (2 & 3), pp. 69–80.

Rigaux, P., Scholl, M., and Voisard, A. (2002). *Spatial Databases: With Application to GIS*. Morgan Kaufmann Publishers.

Robinson, H. A., Morrison, J. L., Muehrke, P. C., Kimerling, A. J., and Guptill, S. C. (1995). *Elements of Cartography*. Wiley, New York, 6th edition.

Ruas, A. (1999). *Modèle de généralisation de données géographiques á base de contraintes et d'autonomie*. Ph.D. thesis, Univeristé de Marne la Vallée, Paris.

Ruas, A. (2000). The Role of Meso Objects for Generalisation. In: *Proceedings of the 9th International Symposium on Spatial Data Handling*, pp. 3b.50–3b.63. Beijing.

Ruas, A. and Lagrange, J. P. (1995). Data and Knowledge Modelling for Generalization. In: J. C. Müller, J. P. Lagrange, and R. Weibel (editors), *GIS and Generalization: Methodology and Practice*, pp. 73–90. Taylor and Francis, London.

Ruas, A. and Plazanet, C. (1996). Strategies for Automated Generalization. In: *Proceedings of the 7th International Symposium on Spatial Data Handling*, Advances in GIS Research II, pp. 6.1–6.18. Delft.

Rusak Mazur, E. and Castner, H. W. (1990). Horton's Ordering Scheme and the Generalisation of River Networks. *The Cartographic Journal*.

Saalfeld, A. (1985). A Fast Rubber-Sheeting Transformation Using Simplicial Coordinates. *The American Cartographer*, **12** (2), pp. 169–173.

Saalfeld, A. (1987). It Doesn't Make me Nearly as CROSS. Some Advantages of the Point-Vector Representation of Line Segments in Automated Cartography. *International Journal of Geographical Infromation Systems*, **1** (4), pp. 379–386.

Samet, H. (1989). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Mass.

Sarjakoski, T., Sarjakoski, T., Lehto, L., Sester, M., Illert, A., Nissen, F., Rystedt, B., and Ruotsalainen, R. (2002). Geospatial Info-Mobility Services - A Challenge for National Mapping Agencies. In: *Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications*. Ottawa.

Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992). Decimation of triangle meshes. *Computer Graphics*, **26** (2), pp. 65–70.

Schürer, D. (2001). *Ableitung von digitalen Landschaftsmodellen mit geringem Strukturierungsgrad durch Modellgeneralisierung*. Ph.D. thesis, Institut für Kartographie und Topographie der Universität Bonn, Bonn.

Schylberg, L. (1993). *Computational Methods for Generalization of Cartographic Data in a Raster Environment*. Ph.D. thesis, Royal Institute of Technology, Department of Geodesy and Photogrammetry, Stockholm. Report No 60.

Sederberg, T. W. and Greenwood, E. (1992). A physically based approach to 2-D shape blending. *Computer Graphics (SIGGRAPH '92 Proceedings)*, **26** (2), pp. 25–34.

Sedgewick, R. (1992). *Algorithms*. Addison-Wesley, Reading, Mass., 3rd edition.

SESTER, M., ANDERS, K. H., AND WALTER, V. (1998). Linking Objects of Different Spatial Data Sets by Integration and Aggregation. *GeoInformatica*, **2** (4), pp. 335–358.

SESTER, M. AND BRENNER, C. (2000). Typification Based on Kohonen Feature Nets. In: *Proceedings of the 1st International Conference on GIScience*, pp. 21–22. Savannah (GA).

SHAPIRA, M. AND RAPPOPORT, A. (1995). Shape Blending Using the Star-Skeleton Representation. *Computer Graphics*, **15** (2), pp. 40–50.

SKIENA, S. S. (1997). *The Algorithm Design Manual*. Springer.

SPIESS, E. (1990). Siedlungsgeneralisierung. In: *Kartographische Publikationsreihe*, volume 10. Schweizerische Gesellschaft für Kartographie, Zurich.

STONEBRAKER, M., BROWN, P., AND MOORE, D. (1999). *Object-Relational DBMSs: Tracking the Next Great Wave 2e*. Morgan-Kauffman Publishers, San Francisco.

STRAHLER, A. N. (1952). Hypsometric (Area-Altitude) Analysis of Erosonal Topology. *Bulletin of the Geological Society of America*, **63**, pp. 1117–1142.

SZYPERSKI, C. AND MURER, D. G. S. (2002). *Component Software*. Addison-Wesley, 2nd edition.

THOMPSON, R. C. AND RICHARDSON, D. E. (1995). A Graph Theory Approach to Road Network Generalization. In: *Proceedings of the 17th ICA/ACI Conference*, pp. 1871–1880. Barcelona.

TÖPFER, F. (1974). Kartographische Generalisierung. In: *Ergänzungsheft Nr. 276 zu Permanns Geographische Mitteilungen*. VEB Hermann Haack, Geographisch Kartographische Anstalt, Gotha/Leipzig.

TÖPFER, F. AND PILLEWIZER, W. (1966). The Principles of Selection: a Means of Cartographic Generalization. *The Cartographic Journal*, **3** (1), pp. 10–16.

TURK, G. (1992). Re-Tiling Polygonal Surfaces. *Computer Graphics*, **26** (2), pp. 55–64.

UITERMARK, H. (1996). The Integration of Geographic Databases. In: *Proceedings of the Second Joint European Conference and Exhibition on Geographical Information*, pp. 92–95. Barcelona.

UITERMARK, H., VAN OOSTEROM, P., MARS, N., AND MOLENAAR, M. (1998). Propagating Updates: Finding Corresponding Objects in a Multi-Source Environment. In: *Proceedings of the 8th International Symposium on Spatial Data Handling*, pp. 580–591. Vancouver.

VISVALINGAM, M. AND WHYATT, J. D. (1993). Line Generalization by Repeated Elimination of Points. *The Cartographic Journal*, **30** (1), pp. 46–51.

WALTER, V. AND FRITSCH, D. (1999). Matching Spatial Data Sets: A Statistical Approach. *International Journal of Geographic Information Systems*, **13** (5), pp. 445–473.

WANG, Z. AND MÜLLER, J. C. (1993). Complex Coastline Generalization. *Cartography and Geographic Information Systems*, **20** (2), pp. 96–106.

WEIBEL, R. (1997a). Generalization of Spatial Data: Principles and Selected Algorithms. In: M. van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer (editors), *Algorithmic Foundations of Geographic Information Systems*, Lectures Notes in Computer Science, pp. 99–152. Springer, Berlin.

WEIBEL, R. (1997b). A Topology of Constraints to Line Simplification. In: M. J. Kraak and M. Molenaar (editors), *Advances in GIS Research II*, pp. 533–546. Taylor and Francis.

WEIBEL, R. AND DUTTON, G. (1998). Constraint-based Automated Map Generalization. In: *Proceedings of the 8th International Symposium on Spatial Data Handling*, pp. 214–224. Vancouver.

WEIBEL, R. AND DUTTON, G. (1999). Generalising Spatial Data and Dealing with Multiple Representations. In: P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind (editors), *Geographical Information Systems: Principles and Technical Issues*, volume 1, pp. 125–155. Wiley, New York, 2nd edition.

WOLBERG, G. (1990). *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos.

WORBOYS, M., HEARNSHAW, H., AND MAGUIRE, D. J. (1990). Object-oriented Data Modelling for Spatial Databases. *International Journal of Geographic Information Systems*, **4** (4), pp. 241–260.

Curriculum vitae

Alessandro Cecconi
born April 21st, 1972, in Sorengo (Lugano)
citizen of Vacallo TI, Switzerland

Education

1978 – 1980	Primary school in Vacallo/TI.
1981 – 1986	Primary school in Feuerthalen/ZH.
1987 – 1992	High school in Schaffhausen (Kantonsschule Schaffhausen), concluded with ”Matura” type ”C” (Mathematics and Natural Science).
1992 – 1993	Studies in mathematics at the Swiss Federal Institute of Technology Zurich (ETHZ).
1993 – 1999	Studies in geography at the University of Zurich. Minors in computer science, cartography, geology and mathematics.
1999	Diploma in geography with a thesis on ”Kartographische Darstellung von statistischen Daten im Internet” (”Cartographic visualization of statistical data in the Internet”) advised by Prof. R. Weibel and Dr. A. Herzog.
1999 – 2003	Research assistant at the Department of Geography, University of Zurich. Dissertation with a thesis on ”Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping” advised by Prof. R. Weibel.